

Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра системного аналізу, комп'ютерного моніторингу
та інженерної графіки

ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ МАКЕТУВАННЯ АФІШ

Пояснювальна записка до магістерської кваліфікаційної роботи

Виконав: студент 2 курсу, групи ІСТ-18м
спеціальності 126 – «Інформаційні системи
та технології»
Білецький Б.С.

Керівник: д.т.н., проф. Мокін В.Б.
Рецензент: к.т.н., доц. Маслій Р.В.

Вінниця ВНТУ – 2019 року

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-2	д.т.н., проф. Мокін В.Б.		
3	к.е.н., доц. Глущенко Л.Д.		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області		
2	Порівняльний аналіз існуючих систем		
3	Вибір характеристик і створення датасету		
4	Побудова моделі та створення програмного забезпечення технології		
5	Експериментальна перевірка роботи технології		
6	Оформлення матеріалів до захисту МКР		

Студент

_____ (підпис)

Білецький Б.С.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Мокін В.Б.

_____ (прізвище та ініціали)

РЕФЕРАТ

Магістерська кваліфікаційна робота: 74 ст., 11 табл., 35 рис., 12 джерел.

Об'єкт досліджень – процес макетування афіши. Предметом дослідження є інформаційна інтелектуальна технологія макетування афіш.

Мета роботи – розробити інтелектуальну технологію макетування афіши, оптимального за критерієм її якості за умов обмежень на ряд параметрів цієї афіши.

Розглянуто задачу макетування афіш, тобто видань, що містять основну графічно-текстову інформацію про подію або захід. Досліджено характеристики макетів афіш з наукових матеріалів, документації до графічних пакетів, а також систематизовано практичний досвід працівників у цій галузі. Визначено ті характеристики, на основі яких пропонується будувати інтелектуальну технологію макетування якісних афіш. Використовуючи професійні дизайнерські інтернет-ресурси Behance, DeviantArt та Dribbble, було створений датасет за допомогою Google Sheets. За цільову характеристику прийнято відношення кількості лайків до кількості переглядів афіши. Запропоновано нову інтелектуальну технологію макетування якісних афіш, яка на етапі налаштування будує модель на основі дерева рішень з оптимізацією гіперпараметрів на прикладі датасету якісних і неякісних афіш. Користувач задає ряд параметрів, які змінювати не можна (обмеження), далі технологія відфільтровує усі класи, які точно не мають таких параметрів, а серед тих, що залишились – вибирає найкращий клас макету афіши, який одночасно задовільнить ці обмеження та буде максимізувати якість афіши. На основі обраного класу формується список рекомендацій для дизайнерів. На основі цих рекомендацій синтезується розмітка афіши. Запропонована технологія успішно випробувана на практиці. Вона дозволить одночасно підвищити якість макетів афіш та зменшити час на їх проектування.

АФІША, МАКЕТУВАННЯ АФІШ, ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ, ПЕРЕДБАЧЕННЯ ЯКОСТІ АФІШИ, ДЕРЕВО РІШЕНЬ.

ABSTRACT

Master's qualification work: 74 pages, 11 tables, 35 pictures, 12 sources.

The object of research is the process of poster layout. The subject of the study is informational intellectual technology for the layout of posters.

The purpose of the work is to develop an intelligent technology for the layout of a poster that is optimal according to the criterion of its quality under conditions of restrictions on a number of parameters of this poster.

The task of poster layout is consider, that is, publications that contain the basic graphical and textual information about an event or event. The characteristics of the layouts of posters from scientific materials, documentation for graphic packages are studied, and the practical experience of workers in this area is systematized. The characteristics on the basis of which it is proposed to build an intelligent technology for the layout of quality posters are determined. Using professional design Internet resources Behance, DeviantArt and Dribbble, a dataset was created using the Google Sheets. According to the target characteristic, the ratio of the number of likes to the number of views of the poster is accepted. A new intelligent design technology of quality posters is proposed, which at the setup stage builds a model based on decision tree with hyperparameters optimization using the dataset of high-quality and low-quality posters as an example. The user sets a number of parameters that cannot be changed (restriction), then the technology filters out all classes that definitely do not have such parameters, and among those that remain, they select the best class for the poster layout that will simultaneously satisfy these restrictions and will maximize the poster quality. Based on the selected class, a list of recommendations for designers is formed. Based on these recommendations, the layout of the poster is synthesized. The proposed technology has been successfully tested on a practice. It will simultaneously improve the quality of poster layouts and reduce time for their design.

EVENT POSTER, EVENT POSTER DESIGN, INTELLECTUAL TECHNOLOGY, PREDICTING THE POSTER QUALITY, DECISION TREE.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ МАКЕТУВАННЯ АФШ	7
1.1 Аналіз предметної області	7
1.1.1 Визначення основних понять.....	7
1.1.2 Огляд та аналіз аналогів	8
1.2 Аналіз літературних джерел	11
1.3 Висновки	16
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МАКЕТУВАННЯ АФШ.....	17
2.1 Вибір характеристик для створення датасету	17
2.2 Визначення цільової характеристики	18
2.3 Створення датасету	20
2.4 Формулювання роботи технології.....	24
2.5 Побудова моделі.....	27
2.5.1 Попередня обробка датасету.....	27
2.5.2 Розвідувальний аналіз.....	28
2.5.3 Оптимізація гіперпараметрів дерева рішень.....	33
2.5.4 Побудова дерева рішень.....	34
2.6 Розробка програмного модуля рекомендацій	37
2.7 Розробка програмного модуля синтезу розмітки	40
2.8 Тестування роботи технології та тестування системи	41
2.9 Висновки	43
3 ЕКОНОМІЧНА ЧАСТИНА	44
3.1 Технологічний аудит розробленої системи.....	44
3.2 Розрахунок витрат на виконання даної роботи.....	47
3.3 Прогнозування комерційних ефектів від реалізації результатів розробки...	52
3.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	54
3.5 Висновки	58
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
Додаток А – Технічне завдання	61
Додаток Б – Лістинг Програми.....	62
Додаток В – Інструкція користувача.....	72
Додаток Г – Графічна частина	74

ВСТУП

Важливу роль в організації будь-якого заходу чи події відіграє якісна афіша, яка, по-перше, привертає увагу цільової аудиторії, по-друге, одразу дає розуміння найбільш привабливих аспектів, які характеризують цю подію, а по-третє, містить лаконічне викладення де це відбудеться і які умови участі. Щосекунди в усьому світі тисячі людей розв'язують задачу оптимального макетування якісних афіш. Для цього використовуються спеціалізовані графічні пакети програм, переважно компанії Adobe Illustrator, Adobe Photoshop та інші [1, 2].

В усьому світі багато дослідників тривалий час досліджують цю проблему, але їх увага зосереджена, перш за все, на формуванні загальних рекомендацій макетування, на формуванні шаблонів (паттернів), визнаних експертами, на формулюванні загальних принципів та правил макетування. Для характеристики якості афіш, як правило, використовується кількість схвальних голосів (лайків) на спеціалізованих сайтах-агрегаторах компаній Adobe (behance.net), Dribbble (dribbble.com) та DeviantArt(deviantart.com), де експерти збирають найкращі, як на їх думку, афіши (постери), та зареєстровані користувачі можуть оцінювати їх. Відповідно, афіши з найбільшою кількістю лайків можна вважати більш якісними, ніж інші, які там розташовані приблизно такий самий час[3-6].

Задача макетування афіши ускладнюється тим, що часто замовник одразу вносить якісь вимоги щодо афіші або фонові рисунки чи фото, які планується використати, вносять обмеження на варіанти розташування тексту біля них. Отже, ця задача є задачею з обмеженнями.

Існують сервіси для макетування афіш. Недоліком цих сервісів є статичний набір шаблонів для створення афіш. Це має два наслідки: з точки зору дизайнера такі сервіси – це низка інструментів, які зменшують творчий потенціал даючи реалізувати лише деякі задуми щодо деталей макетування, накладаючи різними способами обмеження на інші ідеї; з точки зору замовника такі сервіси

представляють собою низку інструментів, які вирішують свою задачу – створення макету афіши, але такі сервіси не мають оцінки якості створених афіш, та не дають представлення, як окремі взяті показники впливають на кінцевий результат. Поширення охоплення та зворотній зв'язок.

Тому актуальним є створення інформаційної технології, яка, з використанням технологій штучного інтелекту та машинного навчання, дозволить автоматично вибирати оптимальні параметри макету афіші, щоб забезпечити їх найбільшу якість за заданих обмежень щодо ряду параметрів.

Об'єктом дослідження є процес макетування афіші. Предметом дослідження є інформаційна інтелектуальна технологія макетування афіш.

Мета дослідження – розробити інтелектуальну технологію макетування афіши, оптимального за критерієм її якості за умов обмежень на ряд параметрів цієї афіши.

Наукова новизна. Запропоновано нову інтелектуальну технологію макетування якісних афіш, яка основана на моделі дерева рішень з оптимізацією гіперпараметрів на прикладі датасету якісних і неякісних афіш та визначенні параметрів по цьому дереву параметрів макету афіши, який одночасно задовільнить задані обмеження та буде максимізувати якість афіши, що дозволить одночасно підвищити якість макетів афіш та зменшити час на їх проектування.

Практичне значення одержаних результатів полягає у наступному:

- створено систему для побудови спеціалізованого датасету афіш, що дозволяє працювати декільком людям над цією задачею, що дає змогу розпаралелити та розподілити цю задачу, таким чином зменшити час для виконання цієї частини роботи. Ця система має легкий для сприйняття користувачем інтерфейс, в той же час має автоматичний модуль експорту в файли, формат якого використовується в задачах аналізу даних;
- побудовано спеціалізований набір даних, що містить систематизовану інформацію про афіши. Даний набір даних можна буде

використовувати не лише для задачі, поставленої в даній роботі, але й для інших задач, які є суміжними;

- написано програмний модуль, що створює модель на основі дерева рішень, що має динамічний характер, тобто за необхідності цю програму можна легко адаптувати під інші набори даних;
- написаний програмний модуль, що реалізує синтез афіши на основі вказаних користувачем обмежень.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується базуванням досліджень на сучасних напрацюваннях науки в цьому напрямку. При оцінці афіш було використано експертні оцінки фахівців, зроблених за допомогою інтернет ресурсів, аудиторія яких виключно представники дизайнерської професії. Застосуванням сучасних комп'ютерних технологій для аналізу даних та побудови моделі. При цьому застосовувались оптимізаційні механізми, які збільшували точність вимірювань, зроблених на базі моделі. Описана технологія має широке застосування, не тільки для задачі макетування афіш, та має залежність між кількістю систематизованих даних та вихідною точністю розрахунків та якості синтезу розмітки.

За результатами магістерської кваліфікаційної роботи опубліковано одну статтю в науковому журналі «Вісник Вінницького політехнічного інституту» [7].

1 АНАЛІЗ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ МАКЕТУВАННЯ АФІШ

1.1 Аналіз предметної області

1.1.1 Визначення основних понять

Тема магістерської накладає певні обмеження до змісту магістерської кваліфікаційної роботи.

Створення інформаційної інтелектуальної технології макетування передбачає наступне:

- Ця технологія має працювати з інформацією, а саме містити інструменти для створення, зміни, збереження, передачі та представлення інформація.
- Ця технологія повинна містити в собі також інтелектуальну складову, а саме використовувати засоби штучного інтелекту, машинного навчання, застосовувати при цьому спеціалізовані інструменти, мови програмування, тощо.
- Так як об'єктом дослідження є процес макетування, а саме макетуванням називається процес взаємної компоновки елементів макету з урахуванням балансу, форми, масштабу, пропорційності, кольору та інших факторів, що визначають сприйняття, для досягнення мети видання [8]. Цей процес має як технічну так і творчу складову. Відповідно дана технологія має працювати з вказаними вище параметрами та на виході давати рекомендації відповідно кращої взаємної компоновки елементів.
- Цільовим продуктом роботи цієї технології є афіша, тобто видання, що містить основну графічну та текстову інформацію про подію або захід [2]. Афіша – це твір, що має на меті звернути увагу глядача до події та спонукати його відвідати. Цей продукт має як і мистецьку, естетичну функцію, в той же час це інструмент ринку та механізмів реклами,

маркетингу. Відповідно кінцевою цілю створення афіши є здобуття прибутку від відвідування заходу зацікавленим глядачем афіши.

1.1.2 Огляд та аналіз аналогів

В сучасному світі розвиток майже всіх галузей людської діяльності пов'язане з використанням інформаційних технологій. Відповідно до предметної області, яку досліджено в даній магістерській кваліфікаційній роботі розглянемо декілька інтернет-сервісів, які пропонують свої рішення задачі макетування. Розглянемо їх в контексті тих вимог які ми описали в 1.1.1.

Розглянемо сервіс Pablo by Buffer Інтерфейс цього сервісу можна побачити на рисунку 1.1.

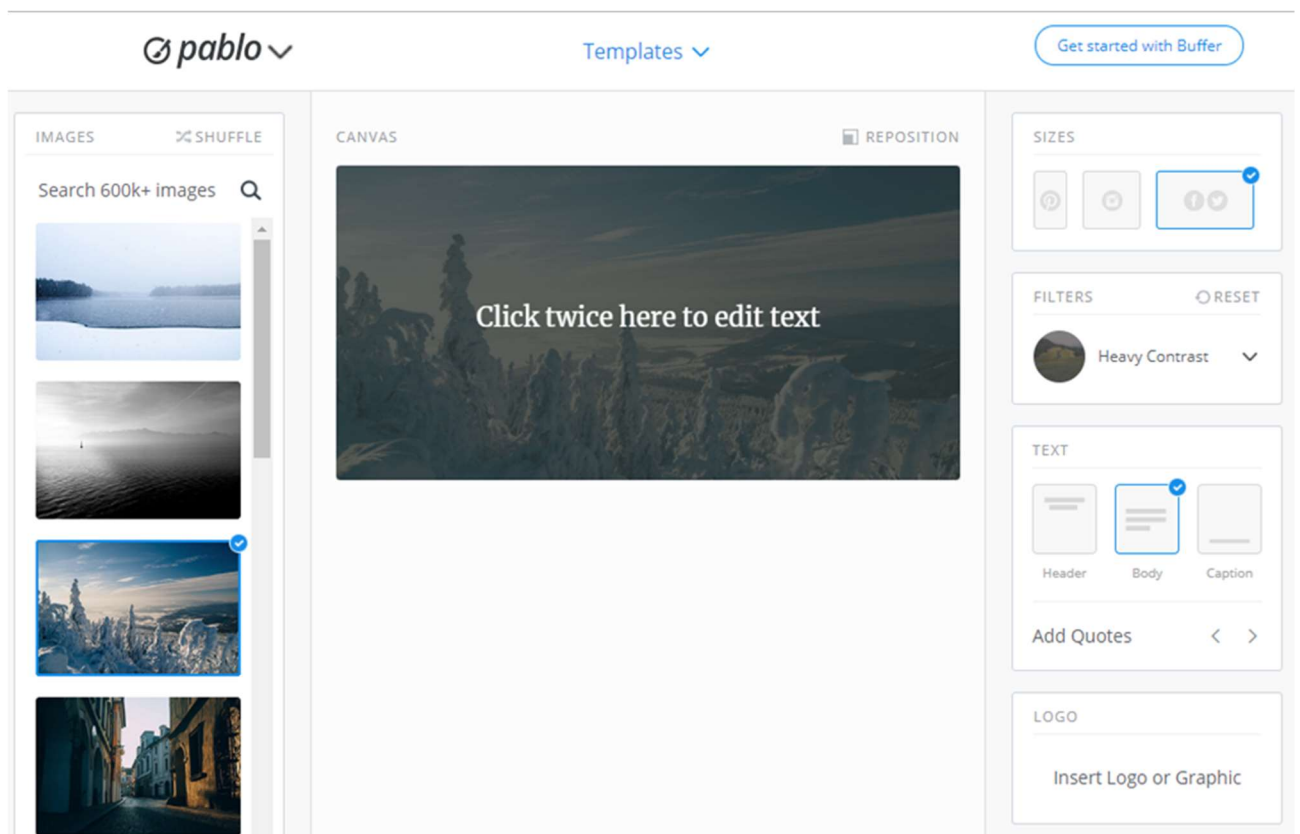


Рисунок 1.1 – Інтерфейс сервісу Pablo by Buffer

Особливостями цього ресурсу є наступне:

- цей сервіс надає можливість створення публікацій стандартних для соціальних мереж розмірів;
- також є можливість застосування трьох блоків тексту: заголовку, основного тексту та підпису;
- є можливість призначити на фон фотографію з інтернету;
- також є можливість вставити логотип організації або інші графічні елементи.

Недоліками цього ресурсу є наступне:

- відсутня можливість створювати матеріали для друку;
- так як афіша має від трьох до п'яти обов'язкових блоків, цей сервіс накладає обмеження на можливі застосування;
- сайт має низьку ступінь автоматизації користувач виконує майже всю роботу. Відсутні паттерни та заготовки для створення макету, окрім фотографій з інтернету;
- сервіс не дає оцінку створеному макету та може давати рекомендації.

Наступний сервіс, який ми розглянемо є PicMonkey Інтерфейс даного сервісу можна побачити на рисунку Рисунок 1.2.

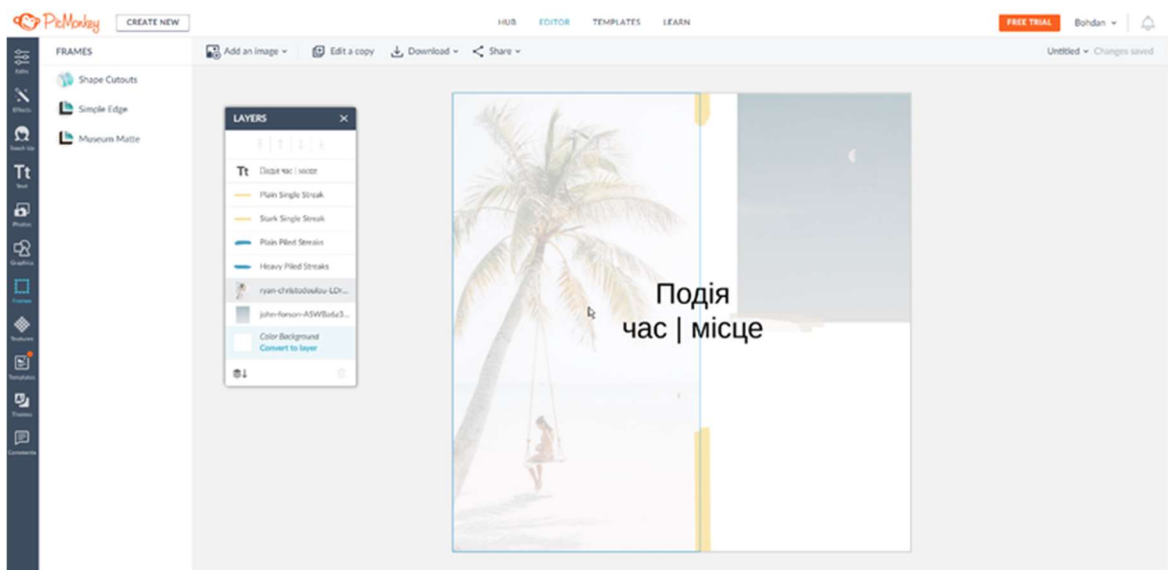


Рисунок 1.2 – Інтерфейс сервісу PicMonkey

Особливостями цього ресурсу є наступне:

- цей сервіс надає можливість обирати користувачеві розміри публікації;
- дає можливість розташувати необмежену кількість об'єктів на афіші;
- наявна велика кількість графічних елементів, які можна застосовувати;
- велика кількість шрифтів, що працюють з українськими символами.

Недоліками цього ресурсу є наступне:

- присутня орієнтація на цифрове видання. Відсутні інструменти підготовки файлів зображень до друку. Це зменшує можливості використання цього сервісу як засобом макетування афіши;
- велика кількість паттернів є незмінними за розмірами, що обмежує їх використання для макетування афіш;
- сервіс не дає оцінку створеному макету та може давати рекомендації;
- сайт не містить інтелектуальної складової;
- сайт має платну основу це ставить під сумнів популярність застосування його в Україні через соціально-економічне становище в країні.

Останнім розглянутим сервісом є Fotor. Інтерфейс цього сервісу можна побачити на рисунку Рисунок 1.3.

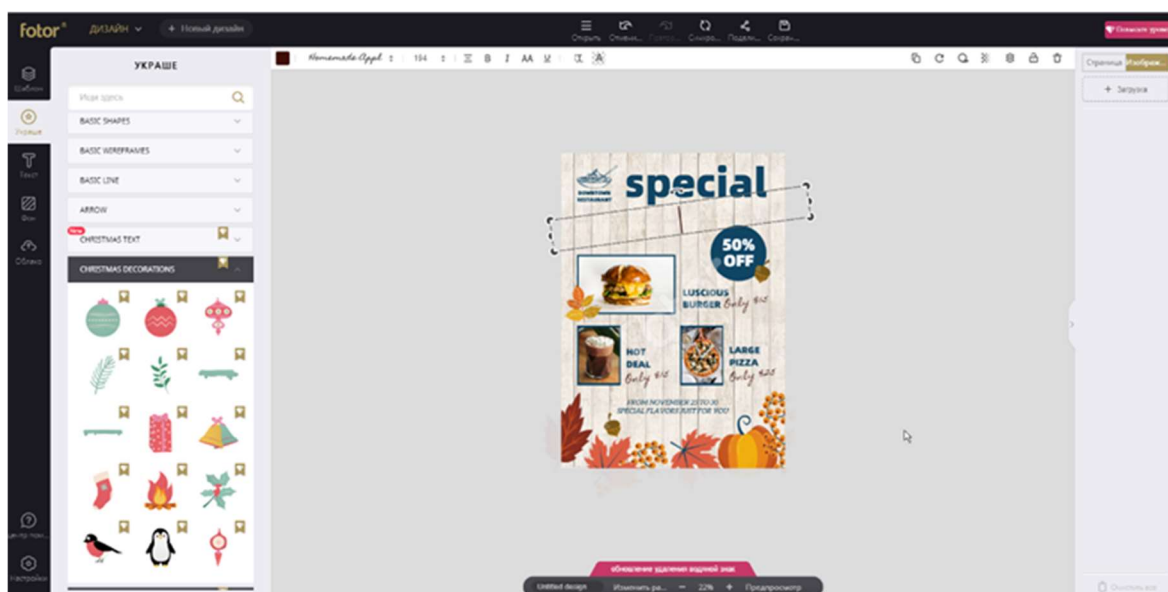


Рисунок 1.3 – Користувачький інтерфейс сервісу Fotor

Особливостями цього ресурсу є наступне:

- Цей сервіс надає можливість обирати користувачеві розміри публікації в тому числі стандартні розміри для друку.

Дає можливість розміщувати необмежену кількість об'єктів на афіші.

Наявна велика кількість графічних елементів, які можна застосовувати.

Недоліками цього ресурсу є наступне:

- Присутня орієнтація на цифрове видання. Відсутні інструменти підготовки файлів зображень до друку. Це зменшує можливості використання цього сервісу як засобом макетування афіши.
- Велика кількість паттернів є незмінними за розмірами відмінними від друкарських, що обмежує їх використання для макетування афіш.
- Сервіс не дає оцінку створеному макету та може давати рекомендації.
- Сайт не містить інтелектуальної складової.
- Сайт є платним, кількість безкоштовних функцій недостатньо для створення сучасної якісної афіши.
- Відсутність можливості працювати з українськими символами в текстових блоках.

Переглянуті вище аналоги містять декілька недоліків, що є спільними для більшості подібних сервісів:

- Відсутність оцінки створеному макету та рекомендації, щодо покращень.
- Сайт не містить інтелектуальної складової.

Отже, існує необхідність задовільнити ці вимоги в розробці інформаційної інтелектуальної технології макетування афіш.

1.2 Аналіз літературних джерел

Метою літературного дослідження в даній роботі є визначення характеристик макетування афіш і макетування в цілому. Окрім вивчення

літературних джерел було також систематизовано фаховий досвід працівників в цій галузі, зокрема автора даної роботи.

За основу було взято три електронні джерела, які знаходяться у вільному доступі в мережі інтернет.

Перше джерело – наукова стаття, яка описує 12 характеристик макетування. По деяким характеристикам була надана рекомендація. Джерело даних рекомендацій не вказано, тому вважається, що рекомендації надав автор статті, не будемо брати їх до уваги, як не фахові.

В другому джерелі перераховано 9 характеристик. З урахуванням дослідження контексту згадки лише п'ять з них мали явне визначення.

Після дослідження другого джерело стало помітно, що характеристики описуються в декількох статтях повторюючи одне-одного. Також деякі характеристики мали різні назви, але відповідно до контексту мають одну сутність. Це свідчить про недостатню формалізованість та систематизованість даної предметної області.

В третьому джерелі три оригінальні характеристики. В цій статті достатньо широко розкрита сутність цих характеристик.

Відповідно до нашої задачі очевидним став розділ характеристик на два класи:

- Перший клас – оцінки по експертним критеріям, які є гештальт-оцінками готового продукту – макету.
- Другий клас характеристики – характеристики, більшістю числові, що мають чітке визначення, та можуть бути корисними у процесі макетування афіші – Характеристики оцінки макету.

Переглянути список характеристик, які ми віднесли до першого класу можна передивитись в таблиці 1.1.

В таблицях 1.1 та 1.2 символом «+» позначено згадування відповідної характеристики у джерелі.

Таблиця 1.1 – Характеристики оцінки макету

№	Назва характеристики	Effective Poster Design [6]	Theory of poster design and presentation [9]	Elements of Poster Design [10]	Авторські
1	Вибір кольорової схеми	+			+
2	Зображення	+			+
3	Повнота інформації (Ясність)	+	+	+	
4	Відповідність	+	+	+	+
5	Стислість	+			
6	Привабливість	+			
7	Розбірливість	+	+	+	
8	Якість зображень		+		
9	Додаткові дизайн-елементи		+	в	+
10	Структурність			+	
11	Наявність головної точки звернення уваги			+	+
12	Виділення головного повідомлення			+	
13	Збалансованість елементів				+
14	Поєднання шрифтів				+
15	Наявність контрасту				+
16	Якість вихідного зображення				+

Переглянути характеристики, які було віднесено до другого класу можна в таблиці 1.2.

Також було взято до уваги показники, що не були описані вище згаданими авторами, але вони є систематизацією фахового досвіду працівників цієї сфери, зокрема автора цієї роботи.

Таблиця 1.2 – Початкові характеристики

№	Назва характеристики	Effective Poster Design [6]	Theory of poster design and presentation [9]	Elements of Poster Design [10]	Авторські
1	Компонування	+			
2	Розмір кегля	+		+	
3	Форматування	+			
4	Використання простору	+			
5	Тип шрифту	+			
6	Нарис шрифту	+			

Пропонуємо вам ознайомитись з описом накопичених параметрів:

- Вибір кольорової схеми. Вибір основних кольорів афіши. Є декілька варіантів: афіша може бути з застосуванням одного кольору, частіше за все чорний, або градації сірого. Є афіши, в яких один головний колір, також є варіанти поєднання двох або більше кольорів. Вважається, що використання більше 3-х кольорів не доцільне [11].
- Зображення. Візуальні образи, що дають інформацію про захід. Це можуть бути фотографії, рисунки, векторна графіка, абстрактні паттерни, символи для супроводження тексту.
- Повнота інформації (Ясність). Повне розуміння всіх наявних елементів афіши. Для афіш обов'язково мати інформацію про назву, її дату та час, місце та адресу події.
- Відповідність. Відповідність кожного елементу дизайну афіши до завдання щодо неї. Важливо, щоб кожен елемент відповідав характеру події та цільовій аудиторії, на яку орієнтована подія.
- Стилість. Відсутність надлишкової інформації. На афішах недоцільно розміщати детальний опис кожної складової події. Для цього

використовують посилання на інтернет-ресурс з довідковою інформацією.

- Привабливість. Візуальна естетика афіши.
- Розбірливість. Здатність тексту бути прочитаним в складних умовах: поганий зір, велика дистанція.
- Якість зображень. Якість застосованих зображень. Розмір, кількість пікселів на дюйм. Візуальна чіткість зображення. Налаштований для друку вибір кольорів.
- Додаткові дизайн-елементи. Прикрашення афіши, елементи, що підсилюють повідомлення.
- Структурність. Впорядковане розташування, при якому елементи, що пов'язані між собою, візуально згруповані.
- Наявність головної точки звернення уваги. Це – один елемент афіши, що привертає увагу глядача та дає йому інтерес до детального вивчення змісту афіши.
- Виділення головного повідомлення. Наявність елементів або способів виділення головного повідомлення. Це може бути виділення за допомогою кольору, фігур, типу та нарису шрифту тощо.
- Збалансованість елементів. Розташування об'єктів, при якому жодна сторона афіши не містить візуально більшу частину елементів.
- Поєднання шрифтів. Гармонійність поєднання різних шрифтів в одному блоці та в афіші в цілому. Доцільно використовувати на афішах не більше 4-х шрифтів [10].
- Наявність контрасту. Наявність головних та другорядних елементів.
- Якість вихідного файлу макету. Оцінка якості файлу, що подається до друку. Для якісного макету доцільно використовувати векторні формати файлів з кольоровим профілем СМУК з роздільною здатністю вище 150 точок на дюйм [12].

1.3 Висновки

Проаналізувавши предметну область та оглянувши аналоги відповідні до теми даної магістерської роботи можна виділити два недоліки, які є суттєвими:

- Відсутність оцінки створеному макету та рекомендації, щодо покращень.
- Сайт не містить інтелектуальної складової.

Дослідивши літературні джерела на предмет визначення характеристик макетування афіш і макетування в цілому. Було знайдено більше 20 характеристик. Було визначено, що деякі характеристики в різних джерелах мали різні назви, але відповідно до контексту мали одну сутність. Це свідчить про недостатню формалізованість та систематизованість даної предметної області.

Відповідно до нашої задачі очевидним став розділ характеристик на два класи:

- перший клас – оцінки по експертним критеріям, які є гештальт-оцінками готового продукту – макету;
- другий клас характеристики – характеристики, більшістю числові, що мають чітке визначення, та можуть бути корисними у процесі макетування афіші – характеристики оцінки макету.

Окрім вивчення літературних джерел було також систематизовано фаховий досвід працівників в цій галузі, зокрема автора даної роботи.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МАКЕТУВАННЯ АФІШ

2.1 Вибір характеристик для створення датасету

Відповідно до мети дослідження виділено основні числові характеристики, які є корисними під час процесу розробки макету афіши. Це сукупність числових та категоріальних характеристик, що описують афішу в цілому. Так як підготовка до магістерської обмежена в часі, було вирішено не систематизувати усі дані по афішам, а обрати чотири головні текстові блоки: заголовок, місце(адреса), дата та час.

Відповідно до них були обрані кількісні та категоріальні параметри, що достатньо точно описують текстовий блок. Параметри та їх кількість обрана мінімально достатньою умовою для коректної роботи технології.

Переглянути обрані характеристики можна в таблиці 2.1.

Таблиця 2.1 – Обрані характеристики

Характеристики						
№	Афіши	Текстового блоку				
		№	Заголовок	Дата	Час	Місце
1	Номер	11	Товщина шрифту			
2	Кількість лайків (КЛ)	12	Тип шрифту			
3	Кількість переглядів (КП)	13	Курсив			
4	Відношення КЛ до КП	14	Розташування по горизонталі			
5	Кількість блоків	15	Розташування по вертикалі			
6	Кількість шрифтів	16	Ширина блоку			
7	Найбільший розмір кегля	17	Висота блоку			
8	Найменший розмір кегля	18	Викривлення			
9	Кількість вільного місця	19	Поворот блоку			
10	Кольоровий фон	20	Вирівнювання			
		21	Розмір кегля			

Загальна кількість характеристик на кожну афішу з 4-ма різними блоками тексту – $10 + (11 * 4) = 54$.

2.2 Визначення цільової характеристики

На початку дослідження за кількісну оцінку якості було обрано кількість лайків. Так як причин, по яким людина може поставити лайк будь-якій публікації було обрано умова для даної характеристики – лайк повинен стояти на фахових дизайнерських сайтах-агрегаторів афіш. Через те, що цими сайтами здебільшого користуються фахові спеціалісти. Відповідно до цього кількість лайків – це інтегральний показник якості афіши.

Приклади сайтів-агрегаторів афіш:

- Behance.net (рис. 2.1);
- dribbble.com (рис. 2.2);
- deviantart.com (рис. 2.3).

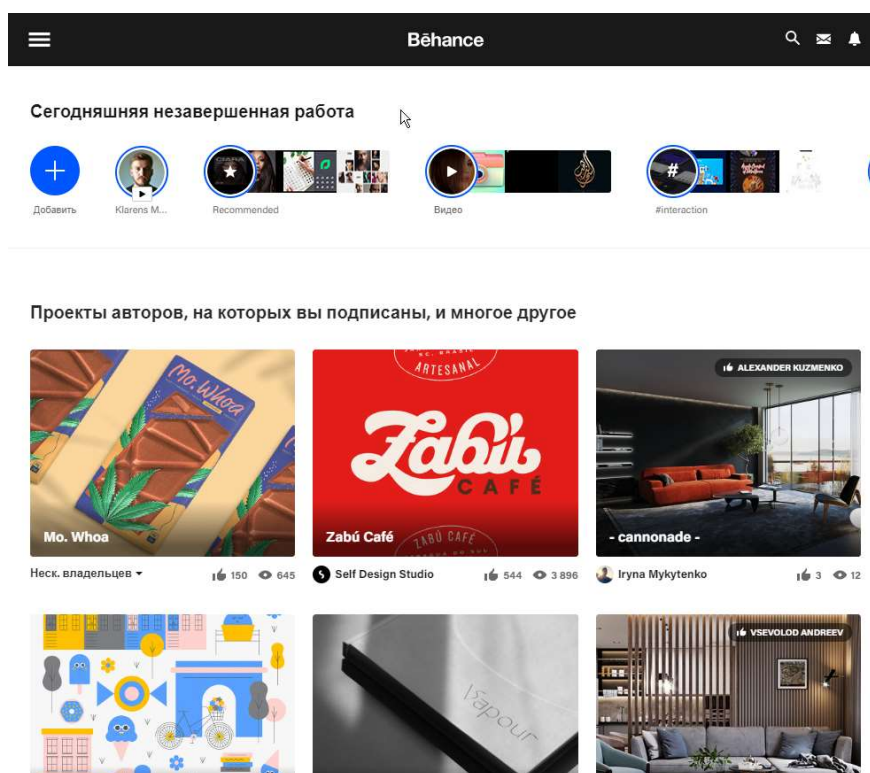


Рисунок 2.1 – Интерфейс веб-сервису Behance.net

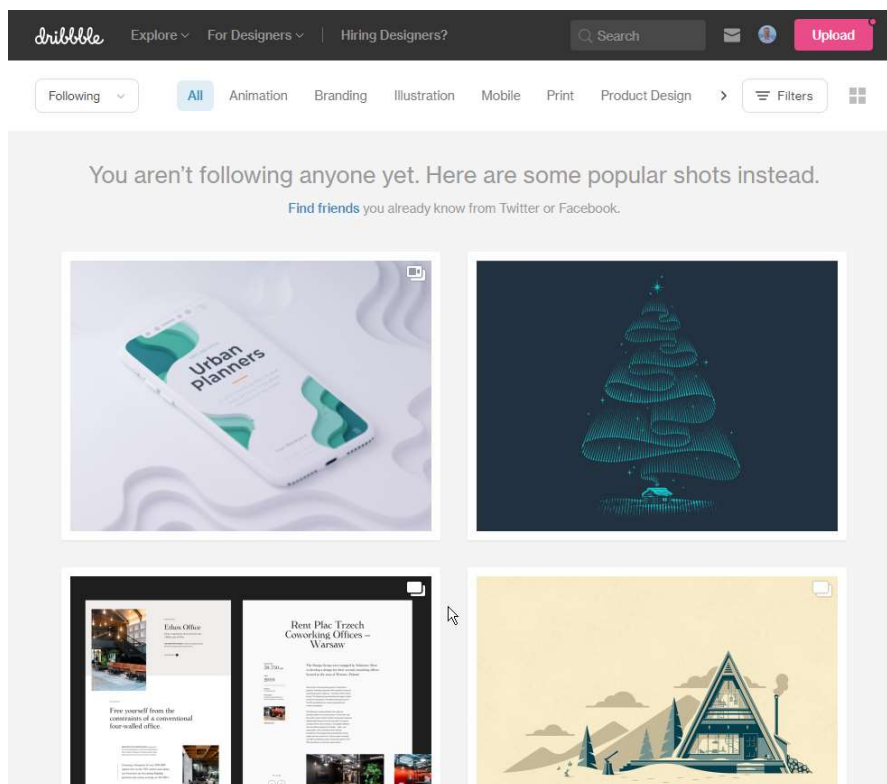


Рисунок 2.2 – Интерфейс веб-сервису Dribbble

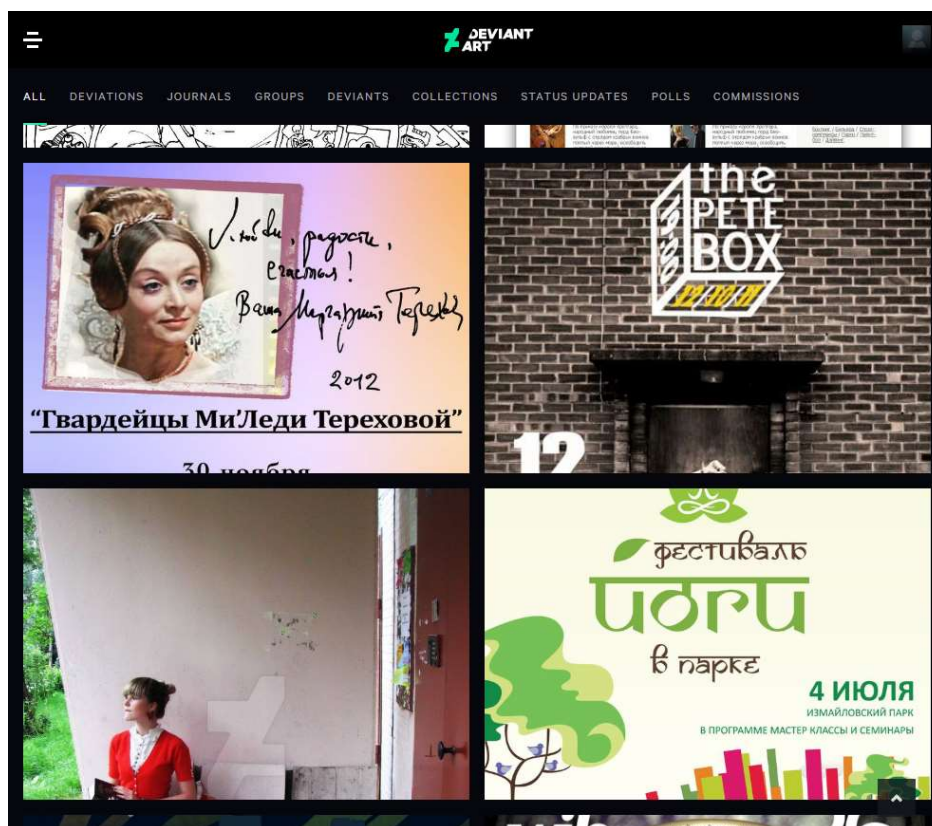


Рисунок 2.3 – Интерфейс веб-сервису Deviantart

Програмні пакети, які користуються популярністю серед дизайнерів такі як:

- Adobe Indesign;
- Adobe Photoshop;
- Adobe Illustrator.

Належать компанії Adobe, яка і є власником першого джерела інформації для створення датасету.

Протягом виконання роботи було визначено, що кількість лайків не може слугувати доцільним кількісним показником якості афіши, по причині того, що афіши, що мають однакову кількість лайків можуть мати різну якість. Для збільшення точності оцінки було обрано більш точний параметр – відношення кількості лайків до кількості переглядів. Цей показник одночасно дає більш точне відносне значення якості афіши та нормалізує можливі значення оцінки в інтервалі від 0 до 1.

але цей показник не дає абсолютного кількісного значення якості афіши через те, що афіши які тільки появились в мережі можуть ще не набрати потенційно можливу кількість лайків. Тому було вирішено брати лише ті афіші, які були завантажені на ці сайти-агрегатори більше ніж 3 місяці, так як за цей час динаміка кількості лайків ймовірно близька до 0. Таким чином ми отримуємо кількісний показник, що достатньо точно може відобразити якість афіш, що були завантажені не менше ніж 3 місяці тому. Отже, саме ця величина може служити цільовою для нашої технології. Саме цю величину ми і будемо максимізувати.

2.3 Створення датасету

Задача створення датасету була розподілена на три етапи:

- Перший етап. Пошук та завантаження афіш на комп'ютер. Відповідно до попередніх досліджень афіши для датасету мають задовольняти наступні вимоги:

- а) Необхідний розмір повинен бути пропорційним до стандартних друкарських розмірів А1, А2, А3, А4, тощо.
 - б) Афіша повинна містити в собі всі зазначені в 2.1 головні текстові блоки.
 - в) Перевага надається афішам з українським текстом з метою більшого практичного значення розробки для застосування в умовах України.
 - г) Необхідно, щоб в датасеті містились не лише якісні афіші, а й неякісні.
- Другий етап. Визначення значень категоріальних ознак афіши. Це параметри оформлення тексту, типу шрифту, тощо.
 - Третій етап. Вимірювання числових показників афіши. В даній магістерській кваліфікаційній роботі для вимірювання числових показників застосовувалась програма Adobe Indesign, яка була налаштована для розмітки афіш. Приклад робочого простору програми, та розміченої афіши можна подивитись на рисунку 2.4.

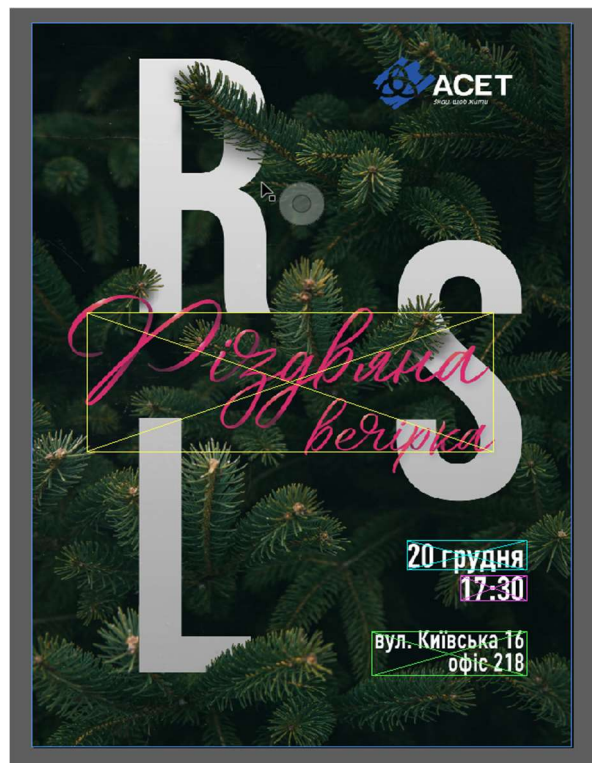


Рисунок 2.4 – Приклад робочого простору програми Adobe Indesign

Розглянемо на прикладі одного текстового блоку, як визначаються основні параметри.

Відповідно приклад значень, що були внесені до датасету можете подивитись в таблиці 2.2.

Таблиця 2.2 – Приклад запису

Загальні характеристики										
№	К-ть лайків	К-ть переглядів	Оцінка	К-ть текстових блоків	К-ть шрифтів	Максимальний розмір кегля, мм	Мін. розмір кегля, мм	К-ть пустого місця, б		
5	215	1614	13%	4	2	48	10	3		
Характеристики блоку заголовку										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Тонкий	Рукописний	1	105	148,5	185	49,5	0	0°	По-правому краю	11,667
Характеристики блоку адреси										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Товстий	Без засічок	0	180	260	20	30	0	0°	По-правому краю	11,667
Характеристики блоку часу										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Товстий	Без засічок	0	180	235	10	30	0°	0°	По-правому краю	11,667
Характеристики блоку дати										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Звичайний	Без засічок	0	180	225	10	30	0°	0°	По-правому краю	11,667

Оскільки час підготовки магістерської роботи обмежено було поставлене завдання організувати побудову датасету з використанням сучасних онлайн-

сервісів, таких як Google Sheets. Головні завдання щодо організації побудови датасету були наступні:

- Розробити систему на базі онлайн табличного процесору Google Sheets
- Зробити процес введення параметрів – автоматизованим, а деякі параметри автоматичним.
- Зробити інтерфейс системи легким для сприйняття нефахових людей для того, щоб була можливість залучити до процесу побудови датасету людей, які не знайомі з розробкою, що описується в даній магістерській кваліфікаційній роботі.
- Розробити автоматизоване виведення даних датасету в формат, який використовується в спеціалізованих інструментах побудови моделі, а саме формат CSV (документ, в якому кожне поле запису відокремлюється від сусідніх комою).

Відповідно до поставлених задач було розроблено систему, яка реалізувала всі необхідні вимоги. Демонстрацію роботи системи можна переглянути на рисунку 2.5 – 2.7.

number	source	count	sOfB	cntF	maxSize	minSize	empty	backl	Don	Title	weight	styleOfFont	Italic	X	Y	W	H	Sk		
1	<input type="checkbox"/> www.devianta http	364	52000	55	4	1	0,0 mm	0,0 mm	3	<input type="checkbox"/>	light	sans-serif	<input type="checkbox"/>							
2	<input type="checkbox"/> www.devianta http	751	104000	135	4	1	0,0 mm	0,0 mm	3	<input type="checkbox"/>	light	sans-serif	<input type="checkbox"/>							
3	<input type="checkbox"/>									<input type="checkbox"/>	light		<input type="checkbox"/>							
4	<input type="checkbox"/>									<input type="checkbox"/>	light		<input type="checkbox"/>							
5	<input type="checkbox"/> behance.net http	150	1614		7	5	48,2 mm	2,5 mm	3	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	106,4	130,9	133,7	48,2	r		
6	<input checked="" type="checkbox"/> behance.net http	200	1614		9	6	47,0 mm	2,5 mm	2	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	107,7	91,1	r	168,2	116,9		
7	<input type="checkbox"/>									<input type="checkbox"/>	light		<input type="checkbox"/>							
8	<input type="checkbox"/>									<input type="checkbox"/>	light		<input type="checkbox"/>							
9	<input type="checkbox"/> behance.net http	2	722	0	4	3	14,6 mm	3,3 mm	4	<input checked="" type="checkbox"/>	regular	sans-serif	<input checked="" type="checkbox"/>	108,0	17,8	r	132,0	17,3	r	
10	<input type="checkbox"/> behance.net http	2	97	0	10	3	16,0 mm	3,0 mm	1	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	72,7	r	31,5	r	111,3	12,3	r
11	<input type="checkbox"/>									<input type="checkbox"/>	light		<input type="checkbox"/>							
12	<input type="checkbox"/> behance.net http	5	99	0	3	2	9,5 mm	6,7 mm	2	<input type="checkbox"/>	bold	serif	<input type="checkbox"/>	102,9	119,2	160,9	8,0	m		
13	<input type="checkbox"/>									<input type="checkbox"/>	light		<input type="checkbox"/>							
14	<input type="checkbox"/> behance.net http	2	41	0	11	2	15,7 mm	2,8 mm	1	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	79,8	r	37,5	r	79,0	15,7	r
15	<input type="checkbox"/> dribbble.com http	1172	135756	0	7	2	11,5 mm	1,9 mm	1	<input checked="" type="checkbox"/>	regular	decorativ	<input type="checkbox"/>	64,0	r	225,8	77,3	38,7	r	
16	<input type="checkbox"/> dribbble.com http	594	39937	0	7	1	13,8 mm	5,3 mm	2	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	103,2	63,3	r	172,3	47,7	r	
17	<input type="checkbox"/>									<input type="checkbox"/>	light		<input type="checkbox"/>							
18	<input type="checkbox"/> dribbble.com http	15	370	0	8	3	30,9 mm	3,8 mm	2	<input checked="" type="checkbox"/>	bold	script	<input checked="" type="checkbox"/>	107,2	214,7	152,3	50,7	r		
19	<input type="checkbox"/> dribbble.com http	598	51081	0	10	2	18,5 mm	3,7 mm	2	<input checked="" type="checkbox"/>	bold	serif	<input type="checkbox"/>	105,5	161,3	93,0	86,3	r		
20	<input type="checkbox"/> behance.net http	33	1475	2	8	6	29,3 mm	5,3 mm	2	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	109,8	125,5	107,7	67,7	r		
21	<input type="checkbox"/> behance.net http	8	196	0	12	2	6,2 mm	3,4 mm	2	<input checked="" type="checkbox"/>	bold	script	<input checked="" type="checkbox"/>	61,7	r	251,1	104,7	6,2	m	

Рисунок 2.5 – Користувацька частина датасету

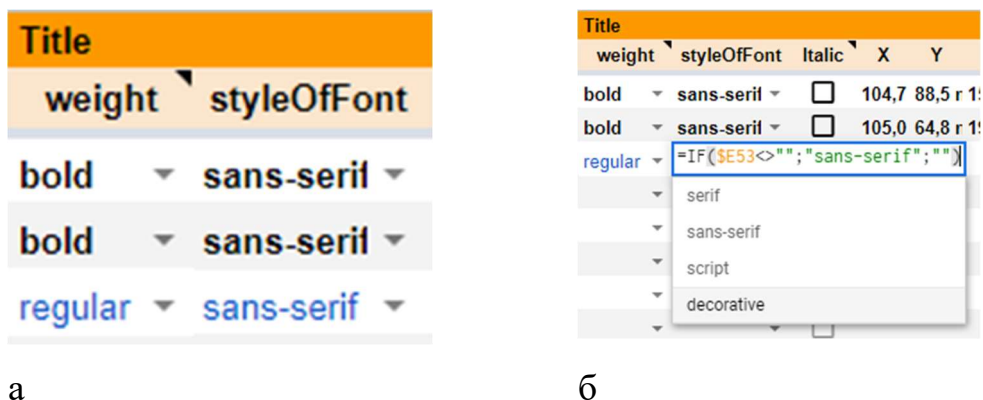


Рисунок 2.6 – Приклади автоматизації: автоматичне заповнення за замовчуванням(а) з можливістю інтерактивного змінення значення(б)

number	likes	views	Mark1	Mark2	countsOf	cntF	maxSi	minSize	empty	backFillColor	title_weight	title_st	title_
5	150	1614	9294	6506	7	5	48,2	2,5	3	0	0	1	
6	200	1614	12392	8674	9	6	47	2,5	2	0	0	1	
9	2	722	277	194	4	3	15	3,3	4	0	1	1	
10	2	97	2062	1443	10	3	16	3,0	1	0	0	1	
12	5	99	5051	3535	3	2	10	6,7	2	1	0	0	
14	2	41	4878	3415	11	2	16	2,8	1	0	0	1	
15	1172	135756	863	604	7	2	12	1,9	1	0	1	3	
16	594	39937	1487	1041	7	1	14	5,3	2	0	0	1	
18	15	370	4054	2838	8	3	31	3,8	2	0	0	2	
19	598	51081	1171	819	10	2	19	3,7	2	0	0	0	
20	33	1475	2237	1566	8	6	29	5,3	2	0	0	1	
21	8	196	4082	2857	12	2	6	3,4	2	0	0	2	
25	124	5907	2099	1469	6	3	24	3,5	1	0	0	3	
28	3	245	1224	857	5	2	18	5,0	4	0	0	0	
29	279	30742	908	635	7	2	17	2,3	3	0	0	1	
30	7	61	11475	8033	5	2	16	6,3	4	0	0	1	

Рисунок 2.7 – Модуль експорту датасету

2.4 Формулювання роботи технології

В даній задачі можна використовувати усі ці технології, але, враховуючи поставлену мету та досвід у сфері розв'язання подібних задач, автори пропонують використовувати таку технологію інтелектуального макетування афіші на основі Python:

- 1) Побудувати датасет з найбільшою кількістю ознак.
- 2) Здійснити розвідувальний аналіз даних (англ. – EDA) з використанням, наприклад, бібліотеки `pandas-profiling`, в якому, в першу чергу,

проаналізувати кількість пропущених даних, найбільші і найменші значення, гістограму значень та кореляційну матрицю – на основі цього, розробити правила для фільтрування помилкових, пропущених та аномальних даних.

- 3) Використовуючи оптимізацію гіперпараметрів побудувати класифікаційне дерево рішень, яке з достатньою точністю розділить множину значень датасету на класи.
- 4) Застосовуючи засоби, наприклад мови Python, зробити можливість користувачеві визначати деякі характеристики власноруч.
- 5) Відповідно до введених даних проходить фільтрація дерева по тих характеристиках, що вказані користувачем. З дерева видаляються ті листя, на шляху до яких порушуються задані обмеження в вузлах.
- 6) З тих листків, які залишились, обирається один, який забезпечує найбільше значення оцінки.
- 7) Вивести рекомендації для дизайнера на екран. Сукупність вузлів дерева, що ведуть до цього листка, і є рекомендаціями для створення афіши.
- 8) На основі визначених рекомендацій синтезувати макет афіши, який далі можна використати як шаблон, принаймні, у першому наближенні з подальшим удосконаленням дизайнерами.

У спрощеному вигляді проілюструємо роботу запропонованої технології. Нехай існує датасет, в якому деяка кількість афіш оцінена за трьома характеристиками. Наприклад характеристики заголовку X_1 – виділення товщиною шрифту, X_2 – курсив, а X_3 – підкреслення текст, поданого цим шрифтом. Якщо на афіші є виділення, відповідно до кожної характеристики, то значення — відповідно 1, якщо немає — то 0. Для кожної афіши визначаються показники лайків та переглядів, а також їх відношення. Застосовуючи класифікацію, за допомогою дерева рішень, видобуто 8 класів, що відрізняються своїми характеристиками та оцінкою (рис. 2.8).

Датасет				
№	Характеристики			Оцінка
	X1	X2	X3	O
1	0	1	0	53
2	0	1	1	50
3	0	0	0	91
4	1	0	0	6
5	1	1	0	83
6	0	0	1	1
7	0	1	1	50
8	0	0	1	2

Датасет				
№	Характеристики			Оцінка
	X1	X2	X3	O
9	1	1	1	56
10	0	0	0	90
11	1	0	1	39
12	1	0	0	5
13	0	1	0	53
14	1	1	1	57
15	1	1	0	83
16	1	0	1	38

Класифікація за допомогою
дерева рішень

Класи	Характеристики			Оцінка
	X1	X2	X3	O
1	0	0	0	91
2	0	0	1	2
3	0	1	0	54
4	0	1	1	51
5	1	0	0	7
6	1	0	1	40
7	1	1	0	84
8	1	1	1	58

Введені характеристики	
X1	X2
0	1

Класи	Характеристики			Оцінка
	X1	X2	X3	O
1	0	0	0	91
2	0	0	1	2
3	0	1	0	54
4	0	1	1	51
5	1	0	0	7
6	1	0	1	40
7	1	1	0	84
8	1	1	1	58

Введені характеристики		Рекомендована	Оцінка
X1	X2	X3	O
0	1	0	54

Рисунок 2.8 – Приклад застосування

2.5 Побудова моделі

Задача побудови моделі була поділена на 3 етапи:

- попередня обробка датасету;
- розвідувальний аналіз;
- оптимізація гіперпараметрів дерева рішень;
- побудова дерева рішень.

2.5.1 Попередня обробка датасету

Оскільки кількість лайків та переглядів – це складові критерію якості, то слід видалити ці характеристики з датасети, замінивши їх на їх відношення.

Оскільки для аналізу даних та створення моделей класифікації необхідно, щоб характеристики були числовими, була проведена попередня обробка даних в датасеті. Всі категоріальні характеристики (товщина шрифту, наявність курсиву, тип шрифту, тощо) були переведені в цілочисельні значення. Наприклад, товщина шрифту, що в нашому датасеті має три значення: «Тонкий», «Звичайний» та «Товстий». Вони відповідно були замінені на цілі числа – 1, 2 та 3 (рис. 2.9).

<i>number</i>	<i>likes</i>	<i>views</i>	<i>Mark1</i>	<i>countsOfBlocks</i>	<i>cntF</i>	<i>maxSize</i>	<i>minSize</i>	<i>empty</i>	<i>backFillColor</i>	<i>title_weight</i>	<i>...</i>	<i>date_styleOfFont</i>	<i>d</i>
5	150	1614	9294		7	5	482	25	3	0	0 ...	0	
6	200	1614	12392		9	6	47	25	2	0	0 ...	0	
9	2	722	277		4	3	15	33	4	0	1 ...	1	
10	2	97	2062		10	3	16	30	1	0	0 ...	3	
12	5	99	5051		3	2	10	67	2	1	0 ...	0	
14	2	41	4878		11	2	16	28	1	0	0 ...	1	
15	1172	135756	863		7	2	12	19	1	0	1 ...	1	
16	594	39937	1487		7	1	14	53	2	0	0 ...	1	
18	15	370	4054		8	3	31	38	2	0	0 ...	1	
19	598	51081	1171		10	2	19	37	2	0	0 ...	1	
20	33	1475	2237		8	6	29	53	2	0	0 ...	1	
21	8	196	4082		12	2	6	34	2	0	0 ...	1	
25	124	5907	2099		6	3	24	35	1	0	0 ...	1	
28	3	245	1224		5	2	18	50	4	0	0 ...	0	

Рисунок 2.9 – Фрагмент обробленого датасету

2.5.2 Розвідувальний аналіз

Застосування моделей LGBM, дерева рішень, логістичної та лінійної регресії дозволило побудувати діаграму важливості ознак рис, з якої видно, що основними з них є такі, як:

- Дата – Поворот блоку;
- Адреса – Розташування X;
- Час – Вирівнювання;
- Час – Курсив;
- Адреса – Поворот блоку;
- Час – Поворот блоку;
- Заголовок – Товщина шрифту;
- Дата – Вирівнювання;
- Найменший розмір кегля;
- Заголовок – Курсив;
- Кількість вільного місця;
- Кольоровий фон;
- Час – Тип шрифту;
- Дата – Курсив;
- Заголовок – Розмір;
- Адреса – Курсив;
- Адреса – Товщина шрифту;
- Адреса – Розташування Y;
- Час – Висота блоку;
- Адреса – Вирівнювання;
- Дата – Розташування Y;
- Заголовок – Поворот блоку;
- Заголовок – Вирівнювання.

Ви можете переглянути діаграму важливості ознак на рисунку 2.10

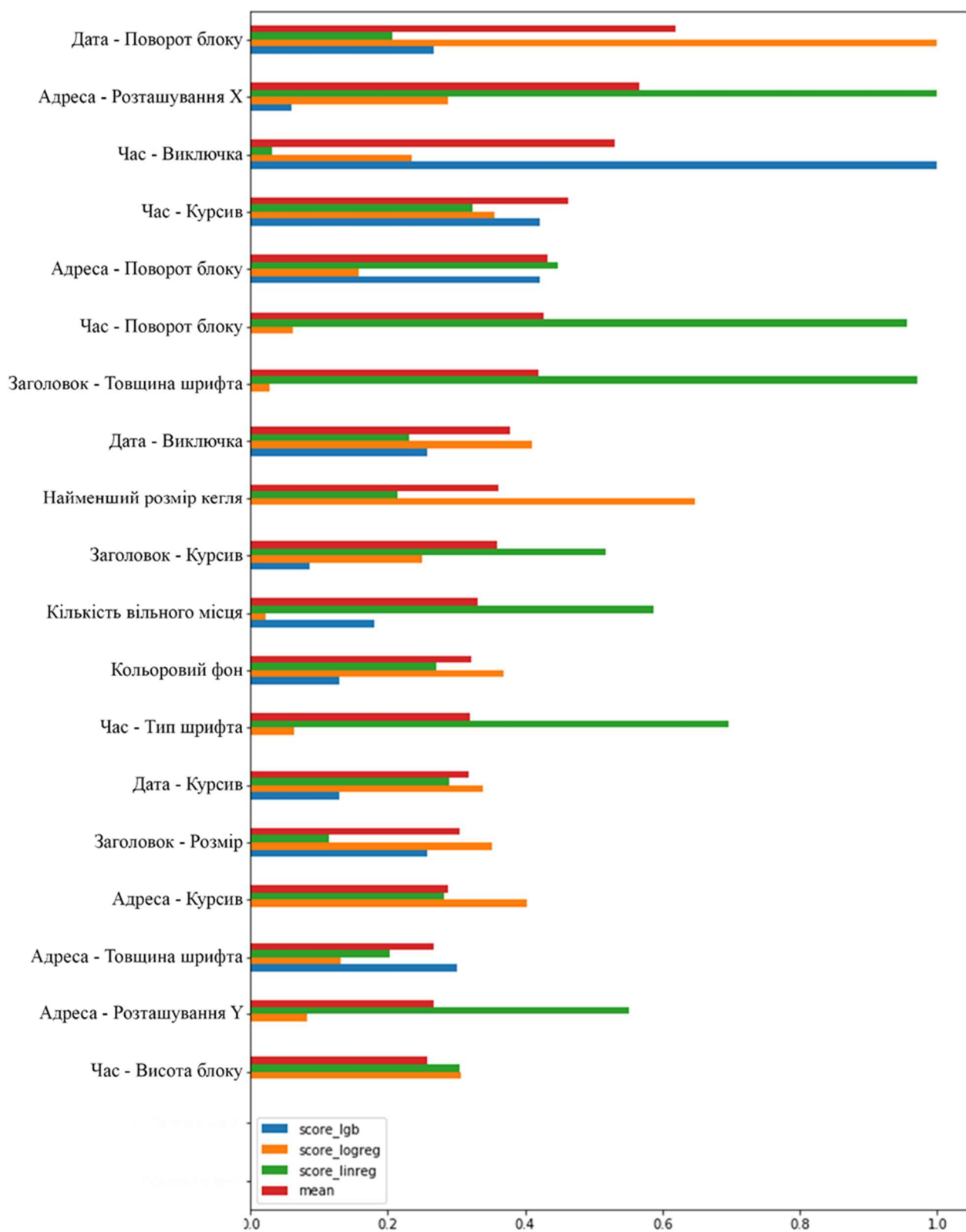


Рисунок 2.10 – Діаграма важливості ознак

Дослідимо данні датасету на основі графіків, створених за допомогою спеціалізованих бібліотек на мові python – sklearn та pyplot.

На рисунку 2.11 відображено залежність параметру положення заголовка по горизонталі та показника якості.

```
g = sns.relplot(x="title_X", y="Mark1", data=fifa19)
```

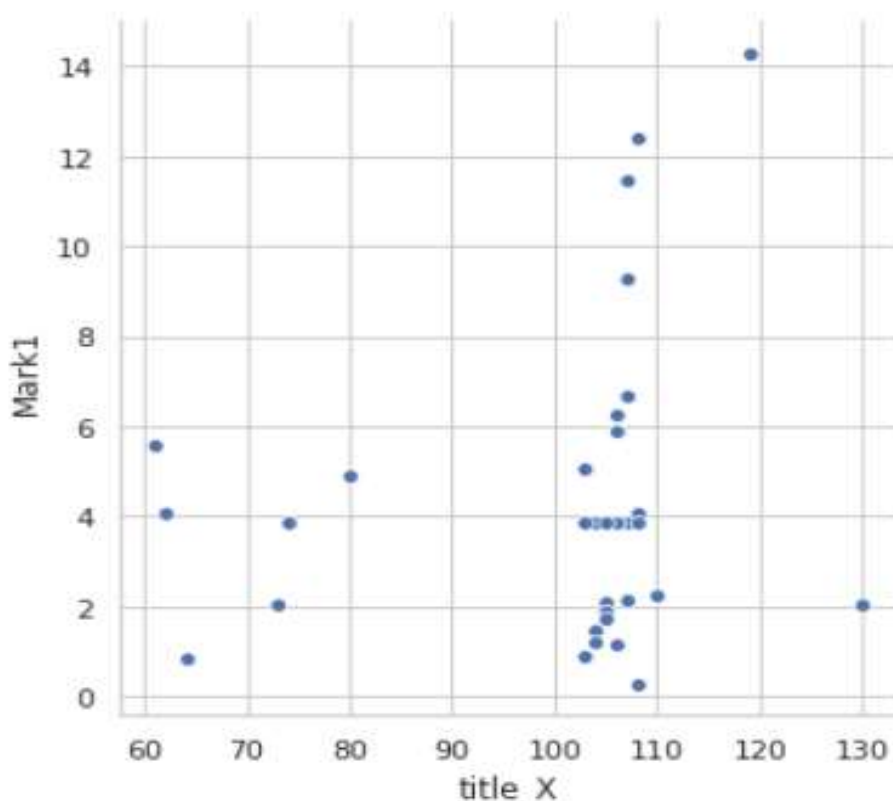


Рисунок 2.11 – Графік залежності показника якості від положення заголовку по горизонталі

Як видно на цьому графіку значення оцінки не сильно корелює з положенням по горизонталі. Однак можна зробити висновок, що здебільшого заголовки знаходяться в інтервалі від 100 до 110 мм по горизонталі. З декількома виключеннями.

На рисунку 2.12 можна побачити частоту використання товщини заголовку.

```
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['title_weight']
x = pd.Series(x, name="title_weight")
ax = sns.distplot(x, bins=10)
plt.show()
```

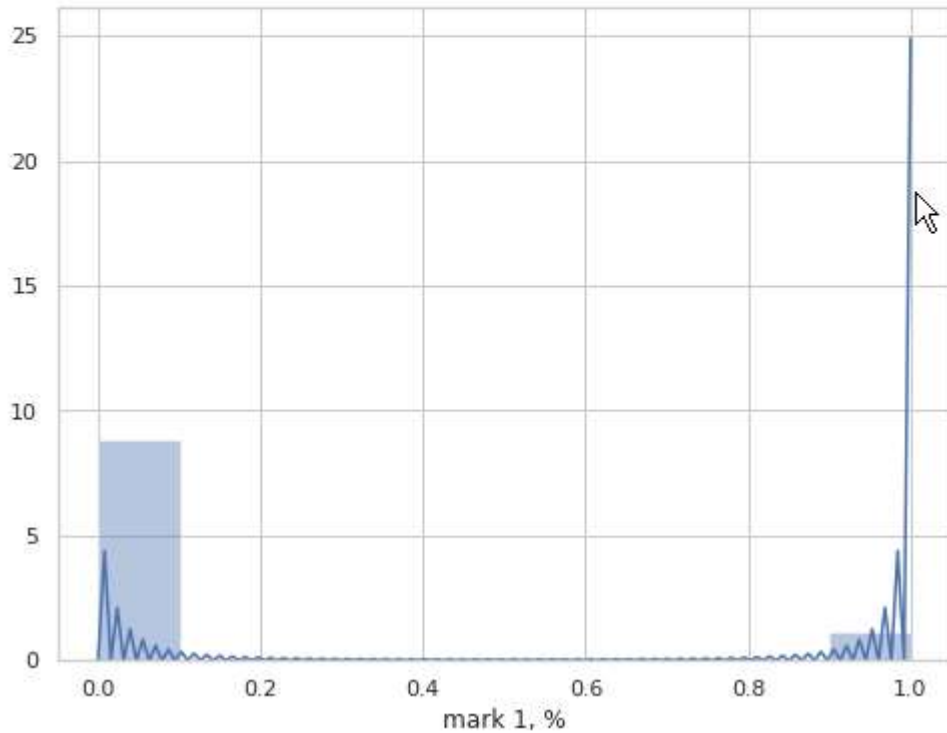


Рисунок 2.12 – Відношення товщини тексту до оцінки

Як видно з цього графіку Найчастіше заголовок не залишають зі звичайною товщиною. Його роблять або тонким або ж товстим, при чому при товстому більша ніж при тонкому.

Проглянемо та порівняємо два графіки (рис. 2.15 та рис. 2.16). На цих графіках показана щільність розподілу значень параметрів лайків та оцінки.

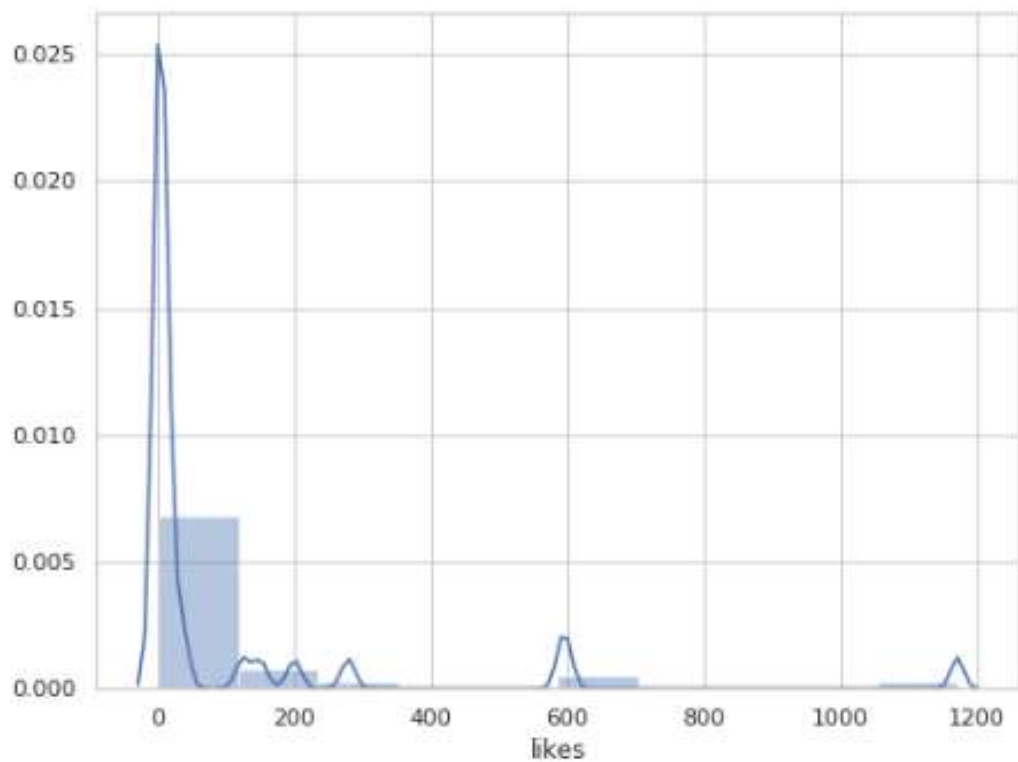


Рисунок 2.13 – Щільність розподілу значень параметру лайки

Щільність розподілу значень параметрів лайків переривчата лінія що має один абсолютний пік та багато маленьких.

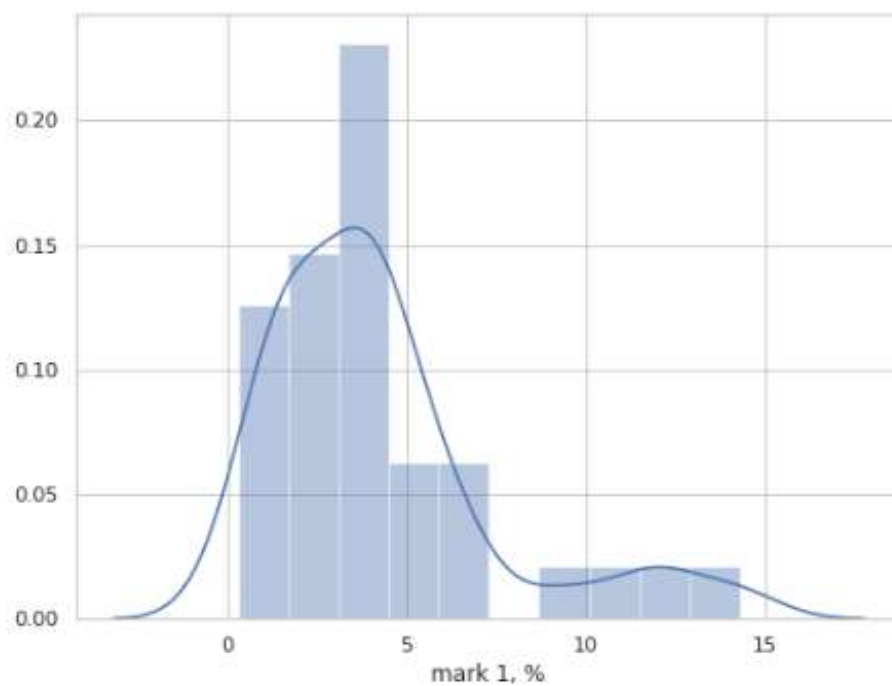


Рисунок 2.14 – Щільність розподілу значень параметру лайки

2.5.3 Оптимізація гіперпараметрів дерева рішень

Так як за основну модель ми беремо дерево рішень (Decision Tree). Перевагою якого як інструмента класифікації в тому, що на виході класифікації ми маємо класифікацію, яку легко інтерпретувати. Налаштування параметрів дерева рішень слід робити з оптимізацією гіперпараметрів, які дозволяють підібрати оптимальну структуру та параметри:

- максимальна глибина дерева;
- кількість вузлів;
- мінімальна кількість об'єктів на листі дерева та інші.

Відповідно для кожного набору значень параметрів було побудоване відповідне дерево рішень. Серед яких було обране те, що дає найбільшу точність за показником, що вимірюється за наступною формулою:

$$A(y, \hat{y}) = \frac{1}{N_3} \sum_{i=0}^{N_3-1} 1(\hat{y}_i - y_i), \quad (2.1)$$

де A – точність, N_3 – кількість зразків,

\hat{y}_i – передбачення для i -ого зразка,

y_i – істинне значення передбачуваної величини у тренувальній вибірці.

Код, що реалізує оптимізацію гіперпараметрів можна переглянути на рисунку 2.17.


```

n_nodes = btree.tree_.node_count
node_depth = np.zeros(shape=n_nodes, dtype=np.int64)
is_leaves = np.zeros(shape=n_nodes, dtype=bool)
stack = [(0, -1)] # seed is the root node id and its parent depth
while len(stack) > 0:
    node_id, parent_depth = stack.pop()
    node_depth[node_id] = parent_depth + 1

    # If we have a test node
    if (children_left[node_id] != children_right[node_id]):
        stack.append((children_left[node_id], parent_depth + 1))
        stack.append((children_right[node_id], parent_depth + 1))
    else:
        is_leaves[node_id] = True

```

Рисунок 2.18 – Код, що реалізує занесення структури дерева в датафрейм

Датафрейм структури дерева рішень можна побачити на рисунку 2.21.

4082	4878	5051	5556	5882	6250	6667	9294	11475	12392	parent_LR	parent_node
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	left	2.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	right	2.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	left	5.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	right	5.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	left	9.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	right	9.0
1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	left	12.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	right	12.0

Рисунок 2.19 – Датафрейм структури дерева рішень

Для кожного листа дерева знайдемо шлях від кореня дерева. Код, що це реалізує дану операцію зображено на рисунку 2.22.

2.6 Розробка програмного модуля рекомендацій

Для розроблення програмного модуля рекомендацій потрібно зробити наступні кроки:

```
pr = btree.predict(X);
arr = []
classes = []
path = []
for n in bLeaves.index:
    for i in range(len(X)):
        if(getDecisionPath(X,i)[-1]==n):

            classes.append((n,pr[i]))
            pat = ','.join(str(x) for x in getDecisionPath(X,i))
            print(pat)
            path.append((n,pat))
            break

bLeaves["class"] = pd.Series(dict(classes))
bLeaves["path"] = pd.Series(dict(path))
```

Рисунок 2.20 – Код знаходження шляху дерева та визначає значення цільової характеристики в ньому

Кінцеву форму датафрейма структури можна побачити на рисунку 2.23.

	9294	11475	12392	parent_LR	parent_node	class	path
3	0.0	1.0	0.0	left	2.0	11475	0,1,2,3
4	0.0	0.0	0.0	right	2.0	277	0,1,2,4
6	0.0	0.0	0.0	left	5.0	3846	0,1,5,6
7	0.0	0.0	0.0	right	5.0	1224	0,1,5,7
10	0.0	0.0	0.0	left	9.0	5882	0,8,9,10
11	0.0	0.0	0.0	right	9.0	1742	0,8,9,11
13	0.0	0.0	0.0	left	12.0	863	0,8,12,13
14	1.0	0.0	1.0	right	12.0	2099	0,8,12,14

Рисунок 2.21 – Кінцева форма датафрейму структури дерева рішень.

2.6 Розробка програмного модуля рекомендацій

Для розроблення програмного модуля рекомендацій потрібно зробити наступні кроки:

- Реалізувати зручне введення інформації в програму;
- Реалізувати пошук в структурі дерева на предмет кращого листа та шляху до нього;
- Реалізувати виведення рекомендацій на екран.

Для першого доцільно застосувати наступний код (рис. 2.24)

В цьому коді у вигляді коментаря всі можливі параметри, які враховує наша технологія та розроблена система.

Відповідно до вимог користувач знімає позначку коментаря стрічку та вносить свої значення. Перевагою такого способу є наочність того, який параметр обирається, а які є суміжні до нього.

```

1: input_param = [
    #("number", ""),
    #("countsOfBlocks", ),
    #("cntF", ),
    #("maxSize", ),
    #("minSize", ),
    #("empty", ),
    #("backFillColor", ),
    #("title_weight", ),
    #("title_styleOfFont", ),
    #("title_Italic", ),
    #("title_X", ),
    ("title_Y", 110),
    #("title_W", ),
    #("title_H", ),
    #("title_Skew", ),
    #("title_Rot.", ),
    #("title_Justification", ),
    #("title_size", ),
    #("adress_weight", ),
    #("adress_styleOfFont", ),
    #("adress_Italic", ),
    ("adress_X", 50),
    #("adress_Y", ),
    #("adress_W", ),
    #("adress_H", ),
    #("adress_Skew", ),
    #("adress_Rot.", ),
    #("adress_Justification", ),
    #("adress_size", ),
    #("time_weight", ),
    #("time_styleOfFont", ),
    #("time_Italic", ),
    #("time_X", ),
    #("time_Y", ),
    #("time_W", ),
    #("time_H", ),
    #("time_Skew", ),
    #("time_Rot.", ),
    #("time_Justification", ),
    #("time_size", ),
    #("date_weight", ),
    #("date_styleOfFont", ),
    #("date_Italic", ),
    #("date_X", ),
    #("date_Y", ),
    #("date_W", ),
    #("date_H", ),
    #("date_Skew", ),
    #("date_Rot.", ),
    #("date_Justification", ),
    #("date_size", )
]

input_param = [
    ("title_Y", 110),
    ("adress_X", 50),
]

sel_features = [feat[0] for feat in input_param]

```

Рисунок 2.22 – Інтерфейс введення даних обмежень

На рисунку 2.22 зображено змінну `input_param`. Ця змінна має тип словник. Структура словників побудована так, що вона містить ключі, та значення. Відповідно дані зберігаються на окремій ділянці пам'яті. Доступ до цих ділянок здійснюється за допомогою ключів. Так як в нашій задачі є багато параметрів афіш і кожен з них має свою назву, було обрано саме такий спосіб збереження цих даних. Застосувавши масив для збереження ми втратили б можливість мати доступ до даних за допомогою ключових слів. А це в свою чергу зменшило б читабельність коду, та ускладнило користувацький досвід

Для другого пункту доцільно використовувати написану функцію с використанням бібліотеки `pandas` (рис. 2.25).

```
def selectBestParameters(param):
    AcceptedNodes = bNodes.copy()
    AcceptedLeaves = bLeaves.copy()
    print("Обрані параметри:")
    AcceptedNodes["accepted"] = True
    for feature, value in param:
        sel_features.append(feature)
        print(f"{feature} : {value}")
        accepted_nodes_feat = AcceptedNodes[(AcceptedNodes["accepted"] == True) &
                                             (AcceptedNodes["feature"] == feature)]

        for node_index in accepted_nodes_feat.index:
            thresh = accepted_nodes_feat.loc[node_index, "threshold"]
            children_right = accepted_nodes_feat.loc[node_index, "children_right"]
            children_left = accepted_nodes_feat.loc[node_index, "children_left"]
            deleteWithChildren(AcceptedNodes, children_right if (value <= thresh) else children_left)

    path = AcceptedLeaves[(AcceptedNodes["isLeaves"] == True) &
                          (AcceptedNodes["accepted"] == True)].sort_values(by="class", ascending=False).head(1);
```

Рисунок 2.23 – Код, що описує пошук кращого листя

На рисунку 2.23 можна побачити функцію `selectBest Parameters`. Ця функція працює з попередньо створеними датафреймами, що описують структуру дерева. Проходячи по всім обраним характеристикам афіш, дана функція перевіряє чи є в вузлах дерева задані характеристики. Якщо є, то алгоритм визначає в якій гілці, що виходить з цього вузла знаходяться наступні вузли та листя, умовам яких дане значення не підлягає. Дані вузли та листя відмічаються як недоступні. В результаті цього алгоритму ми маємо датафрейм, що містить ті вузли та листя, які підходять для формування рекомендацій.

Для третього пункту розроблено наступний код (рис. 2.26).

```

recomendation = "\nРекомендації: \n"
for node_index in range(len(path)-1):
    node = path[node_index]
    feature = bNodes.loc[node, "feature"]
    if (feature in sel_features): continue
    next_node = path[node_index+1]
    Child_LR = ">" if bNodes.loc[next_node, "parent_LR"]=="right" else "<=";
    threshold = round(bNodes.loc[node, "threshold"]);
    recomendation+= f"🕒 Зробіть показник {feature} {Child_LR} {threshold}.\n"
print(recomendation)
return(AcceptedLeaves[(AcceptedNodes["isLeaves"] ==True) &
                      (AcceptedNodes["accepted"] == True)]
       .sort_values(by="class", ascending=False)
       [["parent_LR", "parent_LR", "class", "path"]])
bClass = selectBestParameters(input_param)
bClass

```

Рисунок 2.24 – Код, що виводить рекомендації на екран

На рисунку 2.24 можна побачити опис того, як формується рекомендація на основі відібраних вузлів та листів дерева, що є не суперечать заданим користувачем значенням. Відповідно до цього виконуються наступні дії:

1. Серед наявних листів обирається, той що містить найбільше значення цільового параметру, а саме відношення кількості лайків до кількості переглядів афіши.
2. Відповідно до обраної вершини ми перебираємо усі вузли, що стоять на шляху від даного листа до корня дерева.
3. На кожному вузлі ми виводимо на екран назву характеристики, що є умовою в даному вузлі, та виводимо відповідно знак нерівності, в гілці якого знаходиться обране листя а також значення, яке міститься в даному вузлі
4. Якщо ми дійшли до корня дерева це означає, що алгоритм завершено. Відповідно, якщо рекомендації дані не по всім характеристикам, це означає що дані характеристики не мають значного впливу на цільовий параметр, як дані. Результат виведення рекомендацій можна побачити на рисунку 2.27.

```

Обрані параметри:
empty : 5

Рекомендації:
○ Зробіть показник title_Y <= 61.0.
○ Зробіть показник title_H <= 18.0.
○ Зробіть показник adress_Y <= 86.0.

```

Рисунок 2.25 – Рекомендації відповідно до заданих обмежень

2.7 Розробка програмного модуля синтезу розмітки

Відповідно до даних рекомендацій потрібно розробити модуль синтезу розмітки. Це доцільно зробити за допомогою наступного модуля коду (рис 2.28).

```

output_file('filename.html') # Render to static HTML, or
output_notebook() # Render inline in a Jupyter Notebook
fig = figure(
    background_fill_color='gray',
    plot_height=800,
    plot_width=800
)

(center_X, center_Y) = (210, 297)
(width_of_poster, height_of_poster) = (210, 297)
(origin_X, origin_Y) = (105, 148.5)

fig.rect(x=center_X, y=center_Y, width=width_of_poster, height=height_of_poster, color="white")

for textBlock in ["title", "date", "time", "adress"]:
    color = ""
    if(textBlock == "title"): color = "INDIANRED"
    elif(textBlock == "date"): color = "LIGHTCORAL"
    elif(textBlock == "time"): color = "SALMON"
    elif(textBlock == "adress"): color = "LIGHTSALMON"

    textBlock_X = origin_X + get_feat(textBlock, "X")
    textBlock_Y = origin_Y + get_feat(textBlock, "Y")
    width = get_feat(textBlock, "W")
    height = get_feat(textBlock, "H")

    fig.rect(x = textBlock_X, y = textBlock_Y, width = width, height = height, color = color, name = textBlock)
    #fig.text(X, Y, text='text', angle=0, x_offset=0, y_offset=0, **kwargs)

show(fig)

```

Рисунок 2.26 – Код що реалізує синтез розмітки відповідно до умов

Відповідний результат роботи програми можна побачити на рисунку 2.29.

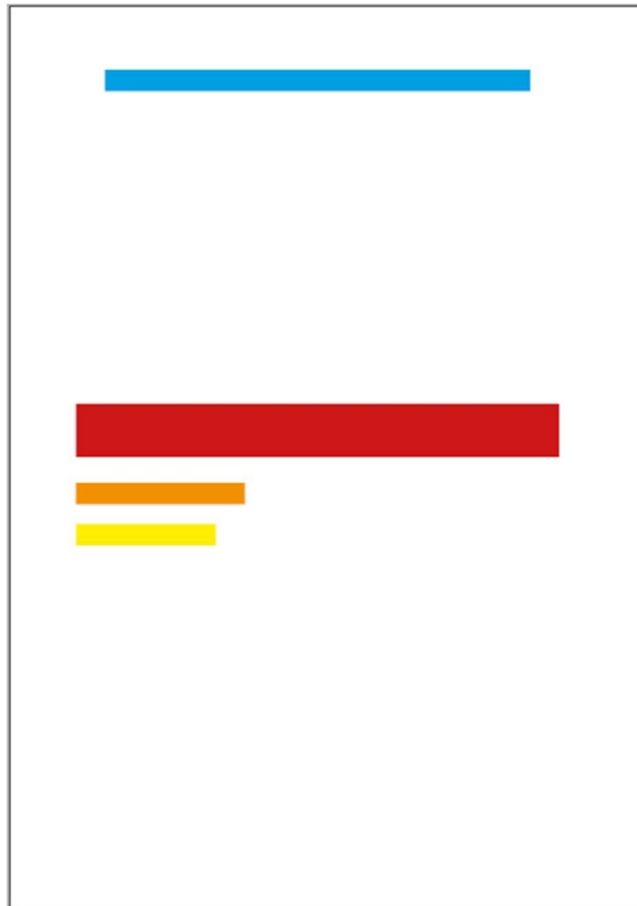


Рисунок 2.27 – Синтезована розмітка макету афіши

2.8 Тестування роботи технології та тестування системи

Користувач розробленої системи вносить обмеження (рис. 2.30).

```
input_param = [  
    ("title_Y", 110),  
    ("adress_X", 50),  
]
```

Рисунок 2.28 – Обмеження параметрів розташування адреси та заголовку афіши

На основі цих обмежень було знайдено 3 листя, які задовольняють задані обмеження (рис. 2.31).

	<i>parent_LR</i>	<i>parent_LR</i>	<i>class</i>	<i>path</i>
12	<i>right</i>	<i>right</i>	2237	0,8,10,12
11	<i>left</i>	<i>left</i>	1171	0,8,10,11
9	<i>left</i>	<i>left</i>	863	0,8,9

Рисунок 2.29 – Листя, що підходять під задані обмеження

Серед них обирається листя з індексом 12, так як значення цільової характеристики(поле *class*) – максимальне. За заданим шляхом (поле *path*) формуються рекомендації (рис. 2.32).

```

Selected features:
title_Y : 110
adress_X : 50

Our recommendations:
○ Зробіть показник time_H > 2.0.
○ Зробіть показник time_Skew > 2.0.

```

Рисунок 2.30 – Рекомендації

Відповідно до наданих рекомендацій система синтезувала розмітку макета (рис. 2.33).

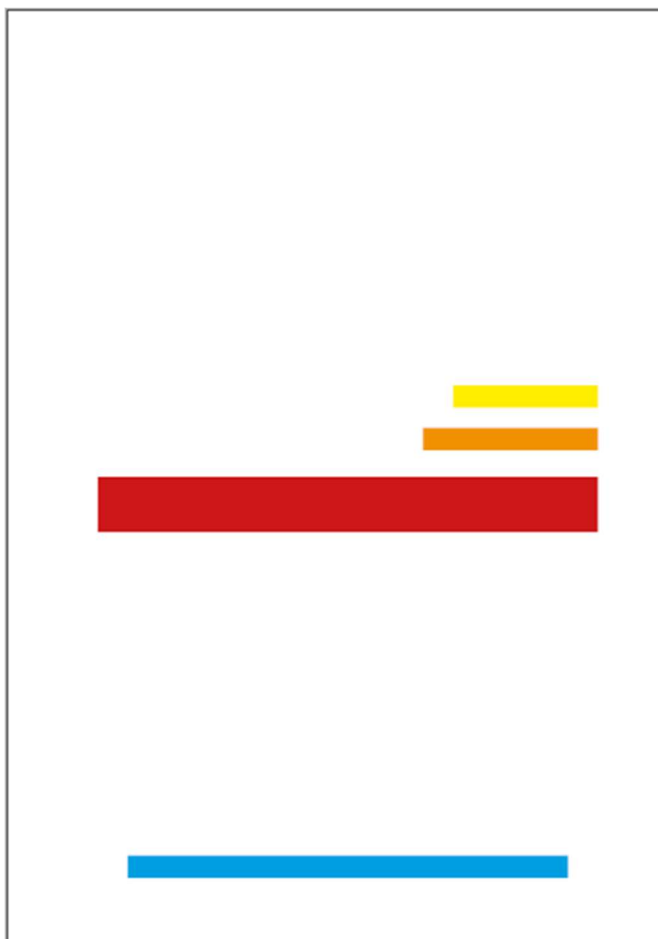


Рисунок 2.31 – Синтезована розмітка

2.9 Висновки

На базі професійних дизайнерських сайтів-агрегаторів афіш було створено датасет, з використанням сучасного сервісу Google Sheets.

Було проаналізовано даний датасет засобами аналізу даних мовою Python та спеціалізованими бібліотеками sklearn, pyplot, numpy.

Було побудовано модель на базі дерева рішень з застосуванням оптимізації гіперпараметрів для збільшення точності.

Побудована система рекомендацій необмежених користувачем характеристик, та модуль синтезу розмітки при заданих обмеженнях та рекомендаціях.

Тестування показало необхідність нарощування датасету для збільшення точності технології.

3 ЕКОНОМІЧНА ЧАСТИНА

3.1 Технологічний аудит розробленої системи

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [20].

Магістерська кваліфікаційна робота за темою “Інтелектуальна інформаційна технологія макетування афіш” передбачає розробку технології, що надає змогу збільшити якість розроблюваних афіш та зменшити час на їх розробку:

Проведемо оцінювання комерційного потенціалу даної розробки. Для проведення технологічного аудиту було залучено 3-х незалежних експертів: Мокін Віталій Борисович – керівник магістерської кваліфікаційної роботи, професор, доктор технічних наук; Крижановський Євгеній Миколайович; Ящолт Андрій Русланович.

В таблиці 3.1 наведено критерії оцінювання комерційного потенціалу розробки та їх оцінки в балах.

Таблиця 3.1 – Критерії оцінювання наукового та комерційного потенціалу розробки та їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 3.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Практична здійсненність					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промислому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років.	Термін реалізації ідеї від 3-х до 5-ти років.	Термін реалізації ідеї менше 3-х років.	Термін реалізації ідеї менше 3-х років.
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Процедура отримання дозвільних документів для виробництва	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки потрібно звести в таблицю за зразком таблиці 3.2.

Таблиця 3.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Мокін В.Б.	Крижановський Є.М.	Яцолт А.Р.
	Бали, виставлені експертами:		
1	4	3	3
2	4	3	2
3	3	4	3
4	4	1	3
5	3	3	5
6	3	5	1
7	4	2	3
8	3	5	2
9	3	2	3
10	4	5	3
11	3	4	2
12	5	2	3
Сума балів	СБ ₁ =43	СБ ₂ =39	СБ ₃ =33
Середньоарифметична сума балів СБ	$\overline{СБ} \frac{\sum_1^3 СБ_i}{3} = \frac{115}{3} = 38.$		

За даними таблиці 3.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 3.3.

Таблиця 3.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11– 20	Нижче середнього
21– 30	Середній
31– 40	Вище середнього
41– 48	Високий

Рівень комерційного потенціалу розробки, становить 38 балів, що відповідає рівню «вище середнього».

3.2 Розрахунок витрат на виконання даної роботи

Проведемо прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи для розробки програмного забезпечення, яке складається з таких етапів:

- 1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;
- 2-й етап: розрахунок загальних витрат на виконання даної роботи;
- 3-й етап: прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат приймаючи до уваги те, що для розробки інформаційної технології було залучено двох розробників програмного забезпечення. Основна заробітна плата кожного із розробників (дослідників) Z_0 , якщо вони працюють в наукових установах бюджетної сфери:

$$Z_0 = \frac{M}{T_p} \cdot t \text{ [грн]}, \quad (3.1)$$

де M – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн;

T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні;

t – число робочих днів роботи розробника (дослідника), розробка програмного забезпечення триває 80 днів.

Зроблені розрахунки внесені до таблиці 3.4.

Таблиця 3.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число Днів роботи	Витрати на оплату праці, грн.
Програміст	5 000	454.54	80	18181.6
Програміст	5 000	454.54	80	18181.6
Всього				36 363,2

2. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12%) від суми основної заробітної плати розробників та робітників, розраховується за формулою:

$$Z_{d1,2} = 0.10 \cdot 18181.6 = 1\,831,81(\text{грн}).$$

3. Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується за формулою:

$$H_{зп} = (Z_0 + Z_d) \cdot \frac{\beta}{100} [\text{грн}], \quad (3.2)$$

де Z_0 – основна заробітна плата розробника, грн.;

Z_d – додаткова заробітна плата розробника, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 37,3%.

$$H_{зп1} = (18181.6 + 1\,831,81) \cdot 0,22 = 8\,799,89(\text{грн}).$$

$$H_{3П2} = (18181.6 + 1\,831,81) \cdot 0,22 = 8\,799,89(\text{грн}).$$

$$H_{3П1} + H_{3П2} = 17599.78$$

4. Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому бути розраховані за формулою:

$$A = \frac{Ц \cdot T}{12 \cdot T_B} [\text{грн}], \quad (3.3)$$

де Ц – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн;

T – фактична тривалість використання, міс;

T_B – термін, використання обладнання, приміщень тощо, місяці, роки.

Зроблені розрахунки наведено в таблиці 3.5.

Таблиця 3.5 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Термін використання, роки	Фактична тривалість використання, міс.	Величина амортизаційних відрахувань, грн
Офісне приміщення	10 000	1	4	2500
Ноутбук 2шт	10 500*2= 21 000	1	4	2100
Всього				4600

5. Інформацію про матеріали, що використовуються при виготовленні даного інноваційного продукту можна переглянути у таблиці 3.6.

Таблиця 3.6 – Матеріали, що використовуються при виготовленні продукту

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено, шт.	Вартість витраченого матеріалу з урахуванням доставки, грн
Папір (пачка)	90,00	1	102,00
Канцтовари	40,00	1	41,00
Всього			143,00

Під час розробки програмного продукту використовували безкоштовні програмні продукти. Здійснювалась оплата домену та хостингу для розміщення сайту. Програмні послуги, що використовуються при виготовленні даного продукту можна переглянути у таблиці 3.7.

Таблиця 3.7 – Програмні послуги, що використовуються при виготовленні даного продукту

Найменування послуги	Ціна за 1 міс., грн.	Кількість місяців	Вартість використаної послуги.
Домен	55	4	220
Хостинг	95	4	380
Всього			600

6. Витрати на силову електроенергію V_e розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi} \text{ [грн]}, \quad (3.4)$$

де V – вартість 1 кВт-год. електроенергії, 1,8 грн/кВт;

P – установлена потужність обладнання, кВт;

Φ – фактична кількість годин роботи обладнання, годин;

K_{Π} – коефіцієнт використання потужності;

Потужність використовуваного комп'ютера становить $P=0.6$ кВт.

Фактична кількість годин роботи обладнання – 640 год (80 робочих днів по 8 годин на день).

$$B_e = 1,8 \cdot 0,6 \cdot 640 \cdot 0,6 = 414,72 \text{ (грн)}.$$

7. Сума всіх попередніх статей витрат дає витрати на виконання даної частини розділу роботи В.

$$\begin{aligned} B &= 36\,363,2 + 3\,636,32 + 8\,799,89 + 4600 + 143 + 414,72 + 600 \\ &= 54\,557,13 \text{ (грн)}. \end{aligned}$$

2-й етап: розрахунок загальних витрат на виконання даної роботи.

$$B_{\text{заг}} = \frac{B}{\alpha} \text{ [грн]}, \quad (3.5)$$

$$B_{\text{заг}} = \frac{54\,557,13}{1} = 54\,557,13 \text{ (грн)},$$

3-й етап. Прогнозування загальних витрат на виконання та впровадження результатів виконаної роботи. Прогнозування витрат ЗВ на виконання та впровадження виконаної роботи здійснюється за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\beta} \text{ [грн]}, \quad (3.6)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Так, як розробка знаходиться:

- на стадії розробки дослідного зразка, то $\beta \approx 0,5$;
- на стадії технічного проектування, то $\beta \approx 0,2$;
- на стадії розробки конструкторської документації то $\beta \approx 0,3$;

- на стадії розробки технології, то $\beta \approx 0,4$;
- на стадії науково-дослідних робіт, то $\beta \approx 0,1$;
- на стадії промислового зразка, $\beta \approx 0,7$;
- на стадії впровадження, то $\beta \approx 0,9$.

$$ЗВ = \frac{54\,557,13}{0,5} = 109\,114,26$$

Отже, прогноз загальних витрат на виконання та впровадження результатів становить 109 114.26 грн.

3.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі проведемо кількісне прогнозування, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. В умовах ринку узагальнюючим позитивним результатом, що його отримує підприємство від впровадження результатів тієї чи іншої розробки, є збільшення чистого прибутку підприємства. Зростання чистого прибутку можна оцінити у теперішній вартості грошей.

Зростання чистого прибутку забезпечить підприємству надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Виконання даної наукової роботи та впровадження її результатів складає приблизно 1 рік.

Позитивні результати від впровадження розробки очікуються вже в перший рік впровадження.

Проведемо детальніше прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_{\text{я}} = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_n \text{ [грн]}, \quad (3.7)$$

де $\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що внаслідок впровадження результатів наукової розробки покращується якість, що дозволяє збільшити прибуток підприємства на 120 грн, а кількість одиниць реалізованої послуги збільшиться: протягом першого року – на 200 од., протягом другого року – ще на 300 од., протягом третього року – ще на 350 од. Орієнтовно: реалізація послуг до впровадження результатів наукової розробки складала 25 шт., а її прибуток підприємства – 600 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ протягом першого року складе:

$$\Delta\Pi_1 = 25 \cdot 600 + (600 + 100) \cdot 200 = 155\,000 \text{ (грн)}.$$

Обчислимо збільшення чистого прибутку підприємства $\Delta\Pi_2$ протягом другого року:

$$\Delta\Pi_2 = 25 \cdot 600 + (600 + 100) \cdot (200 + 300) = 365\,000 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства $\Delta\Pi_3$ протягом третього року становитиме:

$$\Delta\Pi_3 = 25 \cdot 600 + (600 + 100) \cdot (200 + 300 + 350) = 610\,000 \text{ (грн)}.$$

Отже, розрахунки показують, що відповідно прогнозуванню комерційний ефект від впровадження розробки виражається у значному збільшенні чистого прибутку підприємства.

3.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає:

1-й крок. Розрахунок теперішньої вартості інвестицій PV , що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат $ЗВ$ на виконання та впровадження результатів НДДКР, тобто $ЗВ = PV = 77\,938,76$ (грн).

2-й крок. Розрахунок очікуваного збільшення прибутку $\Delta\Pi_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження проведено вище.

3-й крок. Будуємо вісь часу, на якій відображаємо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів. Платежі показуємо у ті терміни, коли вони здійснюються.

Припустимо, що загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнює 77 938,76 грн. Результати вкладених у наукову розробку інвестицій почнуть з'являтися протягом трьох років.

Ці результати виявляться у тому, що у першому році підприємство отримає збільшення чистого прибутку на 132 500 грн відносно базового року, у другому році – збільшення чистого прибутку на 312 500 грн (відносно базового року), у третьому році – збільшення чистого прибутку на 522 500 грн (відносно базового року).

Тоді рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рисунку 3.1.

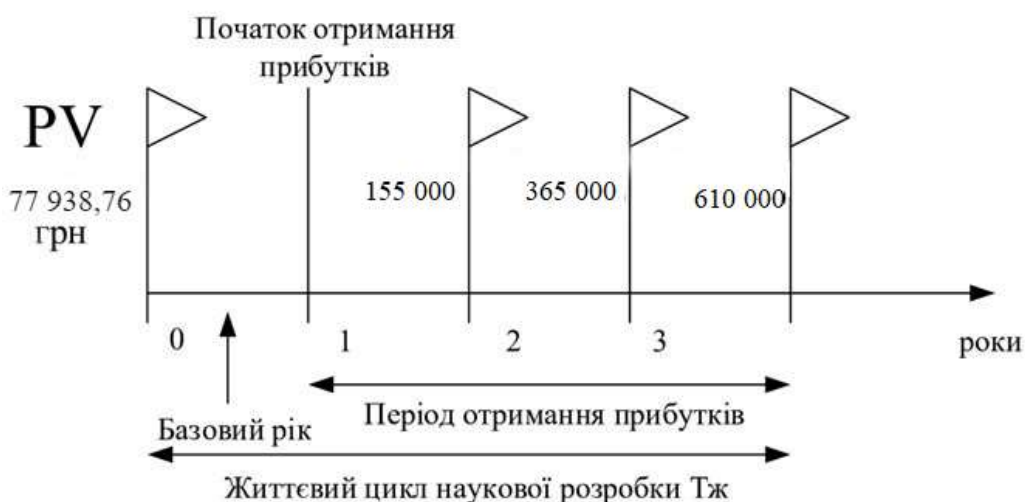


Рисунок 3.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (3.8)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні – 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

$$ПП = \frac{155\,000}{(1 + 0,1)^1} + \frac{365\,500}{(1 + 0,1)^2} + \frac{610\,000}{(1 + 0,1)^3} = 801\,277.23 \text{ (грн)}.$$

$$E_{abc} = 801\,277.23 - 77\,938.76 = 723\,338.47 \text{ (грн)}.$$

Оскільки $E_{abc} > 0$, результат від проведення наукових досліджень щодо розробки програмного продукту та їх впровадження принесе прибуток, тобто є доцільним, але це ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної програми.

5-й крок. Розраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v за формулою:

$$E_v = \sqrt[t]{1 + \frac{E_{abc}}{PV}} - 1, \quad (3.9)$$

де $E_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

$T_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

$$E_B = \sqrt[3]{1 + \frac{723338,47}{77\,938,76}} - 1 = \sqrt[3]{10,02} - 1 = 1,17 \text{ або } 117,00\%.$$

Порівняємо E_B з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

Спрогнозуємо величину $\tau_{\text{мін}}$. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f, \quad (3.10)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; $d = 0,14$;

f – показник, що характеризує ризикованість вкладень; величина $f = 0,3$.

$$\tau = 0,14 + 0,3 = 0,44.$$

Припустимо, що за даних умов прибуток буде збільшуватись, то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

6-й крок. Розраховують термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ за формулою:

$$T_{\text{ок}} = \frac{1}{E_g} \text{ [роки]}. \quad (3.11)$$

$$T_{\text{ок}} = \frac{1}{1,17} = 0,85 \text{ (роки)}.$$

Оскільки термін окупності вкладених у реалізацію наукового проекту інвестицій менше трьох років ($T_{ок} < 3$ років), то фінансування нової розробки є доцільним.

3.5 Висновки

В даному розділі було здійснено оцінювання комерційного потенціалу розробки автоматизованої інформаційної системи робочих місць для співробітників станції технічного обслуговування.

Проведено технологічний аудит з залученням трьох експертів. Аналіз експертних даних показав, що рівень комерційного потенціалу розробки вище середнього. Дослідження комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку.

На основі зроблених підрахунків в економічній частині магістерської кваліфікаційної роботи досягнуті наступні результати:

- витрати на розробку та її впровадження складають 77 938,76 грн;
- абсолютний ефект розробки складає 801 277.23 грн;
- за здійсненими підрахунками було виявлено, що внутрішня норма дохідності була досягнута і складає 117%;
- термін окупності системи, що розробляється складає 0,85 років.

Отже, дана розробка є економічно вигідною та доцільною для вкладання в неї коштів інвесторів.

ВИСНОВКИ

Розроблення інтелектуальної інформаційної технології синтезу афіш показало, що використовувати кількість лайків, як інтегральну оцінку якості постера недоцільно, краще використовувати відношення кількості лайків до кількості переглядів, так як це більше відображає якість афіши. Аналіз літератури показав, що загалом характеристики макетування афіш діляться на два класи – ті, що можуть бути вхідними даними для розробки афіш, та ті, що є цілісними оцінками готового макету. Тож для розробки технології синтезу макету афіш доцільно використовувати саме перші. Також розроблення показало, що гілки дерева рішень можна застосовувати для рекомендаційних систем, в яких є велика кількість характеристик та цільова. Створено програмний продукт, що дає змогу надавати рекомендації відповідно до заданих користувачем обмежень щодо параметрів головних текстових блоків та синтезувати розмітку макету на основі рекомендацій. Тестування показало необхідність нарощування датасету для збільшення точності технології.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Adobe Photoshop CC. Andrew Faulkner, Conrad Chavez, 2018, 384 p.
2. Learn Adobe InDesign CC for Print and Digital Media Publication. Jonathan Gordon, Rob Schwartz, Cari Jansen, 2018, 496 p.
3. Pedwell, RK, Hardy, JA i Rowland, SL (2017), Effective visual design and communication practices for research posters: Exemplars based on the theory and practice of multimedia learning and rhetoric – Pedwell, 2017, Biochemistry and Molecular Biology Education – Wiley Online Library 45: p. 249-261.
4. M. Boers. Theory of poster design and presentation Annals of the Rheumatic Diseases, BMJ Publishing Group Limited, 2018, p. 77 – 118
5. Introduction: poster presentations [Electronic resource] // Access mode: http://www.kumc.edu/SAH/OTEd/jradel/Poster_Presentations/PstrStart
6. Effective Poster Design [Electronic resource] / J.VAN DALEN, H. Gubbels, C. Engel, K. Mfenyana // Education for Health., 2002., Access mode: <https://www.brown.edu/academics/medical/sites/brown.edu/academics/medical/files/uploads/Poster-design.pdf>.
7. Мокін В.Б., Білецький Б.С. Інтелектуальна технологія макетування якісних афіш // Вісник Вінницького політехнічного інституту, 2019, №6
8. ДСТУ 3017 : 2015. Видання. Основні види. Терміни та визначення понять. Київ : Держспоживстандарт України, 2016, с. 42
9. Boers Theory of poster design and presentation Annals of the Rheumatic Diseases 2018, p. 41.
10. Elements of Poster Design [Electronic resource] – Access mode: <http://www.cis.rit.edu/htbooks/dtp/projects/poster/poster1>
11. Yang, C. M., & Hsu, T. F. Applying Narrative Theories in the Design of Public-cause Posters: Process and Educational Implications. // Services for Science and Education, United Kingdom, 2017, p. 29 – 32.
12. Ambrose, Gavin, Paul Harris, and Nigel Ball. The fundamentals of graphic design. Bloomsbury Publishing, 2019, p. 43 – 45.

Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики

ЗАТВЕРДЖУЮ

Завідувач кафедри САКМІГ

_____ д. т. н., проф. В.Б. Мокін

(підпис)

“ ” _____ 2019

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу

ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ

МАКЕТУВАННЯ АФІШ

08-53.МКР.001.02.002 ТЗ

Керівник магістерської
кваліфікаційної роботи
д.т.н., проф.

_____ В.Б. Мокін

(підпис)

“ ” _____ 2019 р.

Розробив студент гр. ІСТ-18м

_____ Б.С. Білецький

(підпис)

“ ” _____ 2019 р.

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 201_ р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САКМІГ від «__» _____ 201_ р.

2. Джерела розробки

1. ДСТУ 3017 : 2015. Видання. Основні види. Терміни та визначення понять. Київ : Держспоживстандарт України, 2016, 42 с.

2. Adobe Photoshop CC. Andrew Faulkner, Conrad Chavez, 2018, 384 p.

3. Learn Adobe InDesign CC for Print and Digital Media Publication. Jonathan Gordon, Rob Schwartz, Cari Jansen, 2018, 496 p.

4. Pedwell, RK, Hardy, JA i Rowland, SL (2017), Effective visual design and communication practices for research posters: Exemplars based on the theory and practice of multimedia learning and rhetoric - Pedwell - 2017 - Biochemistry and Molecular Biology Education - Wiley Online Library 45: 249-261. doi: 10.1002 / bmb.21034

5. M. Boers. Theory of poster design and presentation Annals of the Rheumatic Diseases, BMJ Publishing Group Limited, 2018 Стр. 77-118

3. Мета і призначення роботи

Розробити інтелектуальну технологію макетування афіши, оптимального за критерієм її якості за умов обмежень на ряд параметрів цієї афіши.

4. Вихідні дані для проведення робіт

1) матеріали з сайтів-агрегаторів

5. Методи дослідження

Моделі на основі дерева рішень

6. Етапи роботи і терміни їх виконання

1) Аналіз проблеми макетування афіш..... __. - __

2) Розробка інформаційної технології макетування афіш __. - __

3) Економічна частина __. - __

7. Очікувані результати та порядок реалізації

Створення інформаційної технології, яка, з використанням технологій штучного інтелекту та машинного навчання, дозволить автоматично вибирати оптимальні параметри макету афіші, щоб забезпечити їх найбільшу якість за заданих обмежень щодо ряду параметрів.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

9. Порядок приймання роботи

Публічний захист..... __. __. 201_ р.

Початок розробки «__» _____ 201_ р.

Граничні терміни виконання МКР «__» _____ 201_ р.

Розробив студент групи ІСТ-18м _____ Білецький Б.С.

Додаток Б

Лістинг Програми

```
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-
python

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV,
StratifiedKFold
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
import graphviz
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import pandas_profiling as pp
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
/kaggle/input/posters-dataset/Data.csv
Fnames = pd.Series({
"number": "Номер",
"likes" : "Вподобайки",
"views" : "Перегляди",
"Mark1" : "Оцінка 1",
"Mark2" : "Оцінка 2",
"countsOfBlocks" : "Кількість блоків",
"cntF" : "Кількість шрифтів",
"maxSize" : "Найбільший розмір кегля",
"minSize" : "Найменший розмір кегля",
"empty" : "Кількість вільного місця",
"backFillColor" : "Кольоровий фон",
"title_weight" : "Заголовок - Товщина шрифта",
"title_styleOfFont" : "Заголовок - Тип шрифта",
"title_Italic" : "Заголовок - Курсив",
"title_X" : "Заголовок - Положення X",
"title_Y" : "Заголовок - Положення Y",
"title_W" : "Заголовок - Ширина блоку",
"title_H" : "Заголовок - Висота блоку",
"title_Skew" : "Заголовок - Викривлення",
"title_Rot." : "Заголовок - Поворот блоку",
"title_Justification" : "Заголовок - Виключка",
"title_size" : "Заголовок - Розмір",
"adress_weight" : "Адреса - Товщина шрифта",
"adress_styleOfFont" : "Адреса - Тип шрифта",
"adress_Italic" : "Адреса - Курсив",
"adress_X" : "Адреса - Положення X",
"adress_Y" : "Адреса - Положення Y",
```

```

"adress_Y":"Адреса - Положенння Y",
"adress_W":"Адреса - Ширина блоку",
"adress_H":"Адреса - Висота блоку",
"adress_Skew":"Адреса - Викривлення",
"adress_Rot.":"Адреса - Поворот блоку",
"adress_Justification":"Адреса - Виключка",
"adress_size":"Адреса - Розмір",
"time_weight":"Час - Товщина шрифту",
"time_styleOfFont":"Час - Тип шрифту",
"time_Italic":"Час - Курсив",
"time_X":"Час - Положення X",
"time_Y":"Час - Положенння Y",
"time_W":"Час - Ширина блоку",
"time_H":"Час - Висота блоку",
"time_Skew":"Час - Викривлення",
"time_Rot.":"Час - Поворот блоку",
"time_Justification":"Час - Виключка",
"time_size":"Час - Розмір",
"date_weight":"Дата - Товщина шрифту",
"date_styleOfFont":"Дата - Тип шрифту",
"date_Italic":"Дата - Курсив",
"date_X":"Дата - Положення X",
"date_Y":"Дата - Положенння Y",
"date_W":"Дата - Ширина блоку",
"date_H":"Дата - Висота блоку",
"date_Skew":"Дата - Викривлення",
"date_Rot.":"Дата - Поворот блоку",
"date_Justification":"Дата - Виключка",
"date_size":"Дата - Розмір"});
#pp.ProfileReport(Data)
from sklearn import tree
#Data[["likes","Mark1","views"]].plot()
target = "Mark1"
test_part = 0.2
dropped = ["Mark1","likes","views","Mark2"]
Data_e = Data

for feature in dropped:
    if(feature!=target):
        Data_e = Data_e.drop(feature, axis=1)

X = Data_e.drop(target, axis=1)
Y = Data_e[target]
score = 0.0
while(score<=0.4):
    X_train,X_test,Y_train,Y_test = train_test_split(X,Y, test_size=test_part)

    parameters = {'max_depth' : np.arange(3, 7, dtype=int),
                  'min_samples_leaf' : np.arange(1, 3, dtype=int)}
    classifier = tree.DecisionTreeClassifier(random_state=1000)
    model = GridSearchCV(estimator=classifier, param_grid=parameters,
scoring='accuracy', n_jobs=-1)

    model.fit(X_train, Y_train)
    best_parameters = model.best_params_
    #print(best_parameters)

    btree = tree.DecisionTreeClassifier(max_depth = best_parameters['max_depth'],
                                       random_state = 1118)
    btree.fit(X_train,Y_train);
    score = btree.score(X_test,Y_test)

```

```

print(score)
0.42857142857142855
exp
export_graphviz(btree,out_file="btree.dot",feature_names=names,filled=True,leaves_parallel=True,class_names = btree.classes_.astype(str))
! dot -Tpng btree.dot -o tree.png
image.pngВы можете скачать PNG схемы дерева в PDF

importances = btree.feature_importances_
from matplotlib import pyplot

importances = pd.DataFrame(zip(btree.feature_importances_,
                              index=X.columns,
                              columns=['importance'])).sort_values('importance',
                                                                    ascending=True)

names = pd.Series({
"countsOfBlocks":"Кількість блоків",
"cntF":"Кількість шрифтів",
"maxSize":"Найбільший розмір кегля",
"minSize":"Найменший розмір кегля",
"empty":"Кількість вільного місця",
"backFillColor":"Кольоровий фон",
"title_weight":"Заголовок - Товщина шрифта",
"title_styleOfFont":"Заголовок - Тип шрифта",
"title_Italic":"Заголовок - Курсив",
"title_X":"Заголовок - Положення X",
"title_Y":"Заголовок - Положення Y",
"title_W":"Заголовок - Ширина блоку",
"title_H":"Заголовок - Висота блоку",
"title_Skew":"Заголовок - Викривлення",
"title_Rot.":"Заголовок - Поворот блоку",
"title_Justification":"Заголовок - Виключка",
"title_size":"Заголовок - Розмір",
"adress_weight":"Адреса - Товщина шрифта",
"adress_styleOfFont":"Адреса - Тип шрифта",
"adress_Italic":"Адреса - Курсив",
"adress_X":"Адреса - Положення X",
"adress_Y":"Адреса - Положення Y",
"adress_W":"Адреса - Ширина блоку",
"adress_H":"Адреса - Висота блоку",
"adress_Skew":"Адреса - Викривлення",
"adress_Rot.":"Адреса - Поворот блоку",
"adress_Justification":"Адреса - Виключка",
"adress_size":"Адреса - Розмір",
"time_weight":"Час - Товщина шрифта",
"time_styleOfFont":"Час - Тип шрифта",
"time_Italic":"Час - Курсив",
"time_X":"Час - Положення X",
"time_Y":"Час - Положення Y",
"time_W":"Час - Ширина блоку",
"time_H":"Час - Висота блоку",
"time_Skew":"Час - Викривлення",
"time_Rot.":"Час - Поворот блоку",
"time_Justification":"Час - Виключка",
"time_size":"Час - Розмір",
"date_weight":"Дата - Товщина шрифта",
"date_styleOfFont":"Дата - Тип шрифта",
"date_Italic":"Дата - Курсив",
"date_X":"Дата - Положення X",
"date_Y":"Дата - Положення Y",
"date_W":"Дата - Ширина блоку",

```

```

"date_H": "Дата - Висота блоку",
"date_Skew": "Дата - Викривлення",
"date_Rot.": "Дата - Поворот блоку",
"date_Justification": "Дата - Виключка",
"date_size": "Дата - Розмір"});
#pp.ProfileReport(Data)
from sklearn import tree
#Data[["likes", "Mark1", "views"]].plot()
target = "Mark1"
test_part = 0.2
dropped = ["Mark1", "likes", "views", "Mark2"]
Data_e = Data

imp = importances[importances > 0].dropna()
imp.plot.barh()
<matplotlib.axes._subplots.AxesSubplot at 0x7f3bf6531e80>

n_nodes = btree.tree_.node_count
children_left = btree.tree_.children_left
children_right = btree.tree_.children_right
feature = btree.tree_.feature
threshold = list(map(lambda val: np.round(val,0), btree.tree_.threshold))

# The tree structure can be traversed to compute various properties such
# as the depth of each node and whether or not it is a leaf.
node_depth = np.zeros(shape=n_nodes, dtype=np.int64)
is_leaves = np.zeros(shape=n_nodes, dtype=bool)
stack = [(0, -1)] # seed is the root node id and its parent depth
while len(stack) > 0:
    node_id, parent_depth = stack.pop()
    node_depth[node_id] = parent_depth + 1

    # If we have a test node
    if (children_left[node_id] != children_right[node_id]):
        stack.append((children_left[node_id], parent_depth + 1))
        stack.append((children_right[node_id], parent_depth + 1))
    else:
        is_leaves[node_id] = True

bNodes
Its Dataframe with all nodes

n_nodes = btree.tree_.node_count
node_depth = np.zeros(shape=n_nodes, dtype=np.int64)
is_leaves = np.zeros(shape=n_nodes, dtype=bool)
stack = [(0, -1)] # seed is the root node id and its parent depth
while len(stack) > 0:
    node_id, parent_depth = stack.pop()
    node_depth[node_id] = parent_depth + 1

    # If we have a test node
    if (children_left[node_id] != children_right[node_id]):
        stack.append((children_left[node_id], parent_depth + 1))
        stack.append((children_right[node_id], parent_depth + 1))
    else:
        is_leaves[node_id] = True
def Parent(node):
    there_is = (node in btree.tree_.children_left) or (node in
btree.tree_.children_right)
    if(there_is):
        #print("yes")

```

```

    for n in range(len(btrees.tree_.children_left)):
        #print(n)
        if(btrees.tree_.children_left[n]==node):
            #print(n)
            return ("left",n)
        elif(btrees.tree_.children_right[n]==node):
            #print(n)
            return ("right",n)
    return ("none",-1)
#print(f"left : {btrees.tree_.children_left}")
#print(f"right : {btrees.tree_.children_right}")
#print(Parent(14))
def getDecisionPath(Xdat,n):
    a = btrees.decision_path(Xdat).indptr[n]
    b = btrees.decision_path(Xdat).indptr[n+1] if (a!=len(X)-1) else
(len(btrees.decision_path(X).indices)-1)
    return btrees.decision_path(Xdat).indices[a:b]
def getFeatureNameByID(val):
    return X.columns[val] if val!=-2 else "NaN"
def deleteWithChildren(AcceptedNodes,node):
    AcceptedNodes.loc[node,"accepted"] = False
    if(AcceptedNodes.loc[node,"children_right"] != -1):
        children_left = AcceptedNodes.loc[node,"children_left"]
        children_right = AcceptedNodes.loc[node,"children_right"]
        ret = [children_left,children_right]
        for child in children_left,children_right:
            deleteWithChildren(AcceptedNodes,child)
            AcceptedNodes.loc[child,"accepted"] = False
    return AcceptedNodes
def indexToClassName(index):
    if(type(index) != str):
        return btrees.classes_[index]
    else: return index
val = btrees.tree_.value
bNodes = pd.DataFrame()
for n in range(len(val)):
    bNodes = bNodes.append(pd.DataFrame(val[n][0].reshape(1,-1)),ignore_index=True)
    #print(val[n][0])
bNodes['isLeaves'] = is_leaves
bNodes["children_left"] = btrees.tree_.children_left
bNodes["children_right"] = btrees.tree_.children_right
bNodes["feature"] = list(map(getFeatureNameByID,btrees.tree_.feature))
bNodes["threshold"] = btrees.tree_.threshold
bNodes.rename(indexToClassName, axis='columns',inplace = True)
for ind in bNodes.index:
    (bNodes.loc[ind,"parent_LR"],bNodes.loc[ind,"parent_node"]) = Parent(ind)
print()
bLeaves =
bLeaves=
bNodes[bNodes["isLeaves"]
True].drop(["isLeaves","feature","threshold","children_left","children_right"],axis=1)
bLeaves
pr = btrees.predict(X);
arr = []
classes = []
path = []
for n in bLeaves.index:
    for i in range(len(X)):
        if(getDecisionPath(X,i)[-1]==n):
            classes.append((n,pr[i]))
            pat = ','.join(str(x) for x in getDecisionPath(X,i))
            print(pat)

```

```

        path.append((n,pat))
        break

    names = pd.Series({
"countsOfBlocks":"Кількість блоків",
"cntF":"Кількість шрифтів",
"maxSize":"Найбільший розмір кегля",
"minSize":"Найменший розмір кегля",
"empty":"Кількість вільного місця",
"backFillColor":"Кольоровий фон",
"title_weight":"Заголовок - Товщина шрифта",
"title_styleOfFont":"Заголовок - Тип шрифта",
"title_Italic":"Заголовок - Курсив",
"title_X":"Заголовок - Положення X",
"title_Y":"Заголовок - Положення Y",
"title_W":"Заголовок - Ширина блоку",
"title_H":"Заголовок - Висота блоку",
"title_Skew":"Заголовок - Викривлення",
"title_Rot.":"Заголовок - Поворот блоку",
"title_Justification":"Заголовок - Виключка",
"title_size":"Заголовок - Розмір",
"adress_weight":"Адреса - Товщина шрифта",
"adress_styleOfFont":"Адреса - Тип шрифта",
"adress_Italic":"Адреса - Курсив",
"adress_X":"Адреса - Положення X",
"adress_Y":"Адреса - Положення Y",
"adress_W":"Адреса - Ширина блоку",
"adress_H":"Адреса - Висота блоку",
"adress_Skew":"Адреса - Викривлення",
"adress_Rot.":"Адреса - Поворот блоку",
"adress_Justification":"Адреса - Виключка",
"adress_size":"Адреса - Розмір",
"time_weight":"Час - Товщина шрифта",
"time_styleOfFont":"Час - Тип шрифта",
"time_Italic":"Час - Курсив",
"time_X":"Час - Положення X",
"time_Y":"Час - Положення Y",
"time_W":"Час - Ширина блоку",
"time_H":"Час - Висота блоку",
"time_Skew":"Час - Викривлення",
"time_Rot.":"Час - Поворот блоку",
"time_Justification":"Час - Виключка",
"time_size":"Час - Розмір",
"date_weight":"Дата - Товщина шрифта",
"date_styleOfFont":"Дата - Тип шрифта",
"date_Italic":"Дата - Курсив",
"date_X":"Дата - Положення X",
"date_Y":"Дата - Положення Y",
"date_W":"Дата - Ширина блоку",
"date_H":"Дата - Висота блоку",
"date_Skew":"Дата - Викривлення",
"date_Rot.":"Дата - Поворот блоку",
"date_Justification":"Дата - Виключка",
"date_size":"Дата - Розмір"});
#pp.ProfileReport(Data)
from sklearn import tree
#Data[["likes","Mark1","views"]].plot()
target = "Mark1"
test_part = 0.2
dropped = ["Mark1","likes","views","Mark2"]
Data_e = Data

```

```

bLeaves["class"] = pd.Series(dict(classes))
bLeaves["path"] = pd.Series(dict(path))
0,1,2,3
0,1,2,4
0,1,5,6
0,1,5,7
0,8,9,10
0,8,9,11
0,8,12,13
0,8,12,14
bLeaves
bNotLeaves = bNodes[bNodes["isLeaves"] == False].drop("isLeaves",axis=1)
bNotLeaves
    adress_styleOfFont 0.500000    right 8.0
def Path():
    pr = btree.predict(X_train)
    c = 0
    string = ""
    for i in X_train.index:
        Dat = X_train.loc[i,:]
        print(pr[c])
        c+=1
        print(f"{i}:")
        for node_index in getDecisionPath(X_train,c):
            thresh = bNodes.loc[node_index,'threshold']
            feat = bNodes.loc[node_index,'feature']
            if(feat!="NaN"):
                val = X.loc[i,feat]
                more_or_less = "<=" if val<=thresh else ">"
                print(f"[{node_index}]  {bNodes.loc[node_index,'feature']}  =  {val}
{more_or_less} {round(thresh)}")
            else: print(f"[{node_index}] leaf")
        #print(f"Data:\n{Dat}")
        print()
Input parameters
To select parameters remove "#" and place your value after comma before right bracket

input_param = [
    #("number", ""),
    #("countsOfBlocks", ),
    #("cntF", ),
    #("maxSize", ),
    #("minSize", ),
    #("empty", ),
    #("backFillColor", ),
    #("title_weight", ),
    #("title_styleOfFont", ),
    #("title_Italic", ),
    #("title_X", ),
    #("title_Y", ),
    #("title_W", ),
    #("title_H", ),
    #("title_Skew", ),
    #("title_Rot.", ),
    #("title_Justification", ),
    #("title_size", ),
    #("adress_weight", ),
    #("adress_styleOfFont", ),
    #("adress_Italic", ),
    ("adress_X", 50),

```



```

#("adress_Y", ),
#("adress_W", ),
#("adress_H", ),
#("adress_Skew", ),
#("adress_Rot.", ),
#("adress_Justification", ),
#("adress_size", ),
#("time_weight", ),
#("time_styleOfFont", ),
#("time_Italic", ),
#("time_X", ),
#("time_Y", ),
#("time_W", ),
#("time_H", ),
#("time_Skew", ),
#("time_Rot.", ),
#("time_Justification", ),
#("time_size", ),
#("date_weight", ),
#("date_styleOfFont", ),
#("date_Italic", ),
#("date_X", ),
#("date_Y", ),
#("date_W", ),
#("date_H", ),
#("date_Skew", ),
#("date_Rot.", ),
#("date_Justification", ),
#("date_size", )
]

maximum = path
class_ = path.iloc[0,-2]
path = path.iloc[0].path.split(',')

path = list(map(int,path))

recomendation = "\nOur recomendations: \n"
for node_index in range(len(path)-1):
    node = path[node_index]
    feature = bNodes.loc[node,"feature"]
    if (feature in sel_features): continue
    next_node = path[node_index+1]
    Child_LR = ">" if bNodes.loc[next_node,"parent_LR"]=="right" else "<=";
    threshold = round(bNodes.loc[node,"threshold"]);
    recomendation+= f"🕒 Зробіть показник {feature} {Child_LR} {threshold}.\n"
print(recomendation)
return(class_)

bClass = selectBestParameters(input_param)
bClass
Selected features:
adress_X : 50
from bokeh.io import output_file, output_notebook
from bokeh.plotting import figure, show
from bokeh.models import ColumnDataSource
from bokeh.layouts import row, column, gridplot
from bokeh.models.widgets import Tabs, Panel
from bokeh.models import LabelSet
from bokeh.models.glyphs import Text
Color Random generator
bokeh.colors.rgb.RGB

from bokeh.colors import *
```

```

def random_color():
    return (randint(0,255),randint(0,255), randint(0,255), random())
random_color()
(35, 203, 151, 0.5369677682653058)
Canvas
data_paint_X = X.copy()
data_paint_X['predict'] = btree.predict(X)
example_posters = data_paint_X[data_paint_X["predict"] == bClass]
select_index = randint(0,len(example_posters)-1)
example_poster = example_posters.iloc[select_index,:]
example_poster = dict(example_poster)
def get_feat(textBlock,feat):
    get_feat_feature = textBlock+"_"+feat
    #print(get_feat_feature)
    if(get_feat_feature in sel_features):
        val = input_param[sel_features.index(get_feat_feature)][1]
        #print(f"{get_feat_feature} : {val}")
        return val
    else:
        val = int(example_poster[get_feat_feature])
        #print(f"{get_feat_feature} : {val}")
        return val
output_file('filename.html') # Render to static HTML, or
output_notebook() # Render inline in a Jupyter Notebook
fig = figure(
    background_fill_color='gray',
    plot_height=800,
    plot_width=800
)

(center_X,center_Y) = (210,297)
(width_of_poster, height_of_poster) = (210,297)
(origin_X, origin_Y) = (105,148.5)

fig.rect(x=center_X,y=center_Y,width=width_of_poster,height=height_of_poster,color="white")

for textBlock in ["title","date","time","adress"]:
    color = ""
    if(textBlock == "title"): color = "INDIANRED"
    elif(textBlock == "date"): color = "LIGHTCORAL"
    elif(textBlock == "time"): color = "SALMON"
    elif(textBlock == "adress"): color = "LIGHTSALMON"

    textBlock_X = origin_X + get_feat(textBlock,"X")
    textBlock_Y = origin_Y + get_feat(textBlock,"Y")
    width = get_feat(textBlock,"W")
    height = get_feat(textBlock,"H")

    fig.rect(x = textBlock_X, y = textBlock_Y, width = width, height = height, color =
color,name = textBlock)
    #fig.text(X, Y, text='text', angle=0, x_offset=0, y_offset=0, **kwargs)

show(fig)

```

BokehJS 1.3.4 successfully loaded.

Додаток В

Інструкція користувача

Користувач розробленої системи вносить обмеження в поле `input_param` (рис. В.1). Та запускає програму

```
input_param = [
    ("title_Y", 110),
    ("adress_X", 50),
]
```

Рисунок В.1 – Обмеження параметрів розташування адреси та заголовку афіши

На основі цих обмежень програма знаходить листя, які задовольняють задані обмеження (рис. В2).

	<i>parent_LR</i>	<i>parent_LR</i>	<i>class</i>	<i>path</i>
12	<i>right</i>	<i>right</i>	2237	0,8,10,12
11	<i>left</i>	<i>left</i>	1171	0,8,10,11
9	<i>left</i>	<i>left</i>	863	0,8,9

Рисунок В.2 – Листя, що підходять під задані обмеження

Серед них обирається листя із максимальним значенням цільової характеристики(поле `class`). За заданим шляхом (поле `path`) формуються рекомендації (рис. В.3).

```
Selected features:  
title_Y : 110  
adress_X : 50  
  
Our recomendations:  
○ Зробіть показник time_H > 2.0.  
○ Зробіть показник time_Skew > 2.0.
```

Рисунок В.3 – Рекомендації

Відповідно до наданих рекомендацій система синтезувала розмітку макета (рис. В.4).

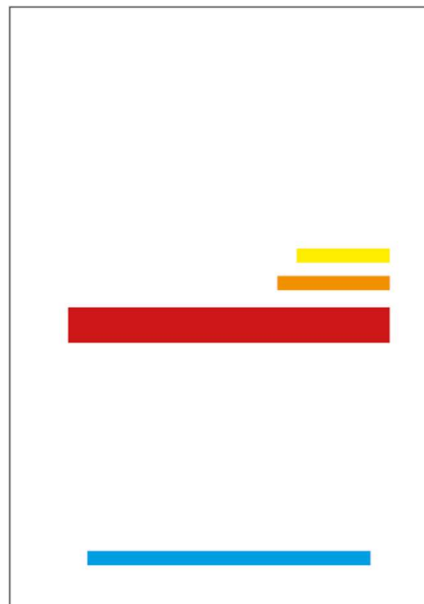


Рисунок В.4 – Синтезована розмітка

Синтезована розмітка та надані рекомендації і є результатом роботи системи.

Додаток Г
Графічна частина

Характеристики афіш

№	Назва характеристики	Effective Poster Design [7]	Theory of poster design and presentation [8]	Elements of Poster Design [9]	Авторські
1	Вибір кольорової схеми	+			+
2	Зображення	+			+
3	Повнота інформації (Ясність)	+	+	+	
4	Відповідність	+	+	+	+
5	Стислість	+			
6	Привабливість	+			
7	Розбірливість	+	+	+	
8	Якість зображень		+		
9	Додаткові дизайн-елементи		+		+
10	Структурність			+	
11	Наявність головної точки звернення уваги			+	+
12	Виділення головного повідомлення			+	
13	Збалансованість елементів				+
14	Поєднання шрифтів				+
15	Наявність контрасту				+
16	Якість вихідного зображення				+
17	Компонування	+			
18	Розмір кегля	+		+	
19	Форматування	+			
20	Використання простору	+			
21	Тип шрифту	+			
22	Нарис шрифту	+			

08-53.МКР.001.03.008.ПЛ

					08-53.МКР.001.03.008.ПЛ			
					Інтелектуальна інформаційна технологія макетування афіш	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				1 : 1
Розроб.		Білецький Б.С.						
Перевір.		Мокін В.Б.						
Т. Контр.						Арк.	1	Аркушів 8
Реценз.		Маслій Р.В.			Характеристики афіш	ІСТ-18м		
Н. Контр.		Жуков С.О.						
Затверд.		Мокін В.Б.						

Приклад афіши



08-53.МКР.001.03.008.ПЛ

					08-53.МКР.001.03.008.ПЛ			
					Інтелектуальна інформаційна технологія макетування афіш	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				1 : 1
Розроб.		Білецький Б.С.						
Перевір.		Мокін В.Б.						
Т. Контр.						Арк.	1	Аркушів 8
Реценз.		Маслій Р.В.			Характеристики афіш	ІСТ-18м		
Н. Контр.		Жуков С.О.						
Затверд.		Мокін В.Б.						

Приклад запису в датасеті

Загальні характеристики										
№	Кількість лайків	К-ть переглядів	Оцінка	К-ть текстових блоків	К-ть шрифтів	Максимальний розмір кегля, мм	Мін. розмір кегля, мм	К-ть пусто го місця, б		
5	215	1614	13%	4	2	48	10	3		
Характеристики блоку заголовку										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Тонкий	Рукописний	1	105	148,5	185	49,5	0	0°	По-правому краю	11,667
Характеристики блоку адреси										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Товстий	Без засічок	0	180	260	20	30	0	0°	По-правому краю	11,667
Характеристики блоку часу										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Товстий	Без засічок	0	180	235	10	30	0°	0°	По-правому краю	11,667
Характеристики блоку дати										
Товщина шрифту	Тип шрифту	Курсив	Розташування X, мм	Розташування Y, мм	Ширина	Висота	Викривлення	Поворот	Вирівнювання	Розмір кегля, мм
Звичайний	Без засічок	0	180	225	10	30	0°	0°	По-правому краю	11,667

Користувачька частина датасету

number	source					count sOfBlocks	cntF	maxSize	minSize	empty	backl	Done	Title	weight	styleOfFont	Italic	X	Y	W	H	Sk
1	<input type="checkbox"/>	www.deviantart.com	http	364	52000	55	4	1	0,0 mm	0,0 mm	3	<input checked="" type="checkbox"/>	light	sans-serif	<input type="checkbox"/>						
2	<input type="checkbox"/>	www.deviantart.com	http	751	104000	135	4	1	0,0 mm	0,0 mm	3	<input checked="" type="checkbox"/>	light	sans-serif	<input type="checkbox"/>						
3	<input type="checkbox"/>										<input checked="" type="checkbox"/>	light		<input type="checkbox"/>							
4	<input type="checkbox"/>										<input checked="" type="checkbox"/>	light		<input type="checkbox"/>							
5	<input type="checkbox"/>	behance.net	http	150	1614		7	5	48,2 mm	2,5 mm	3	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	106,4	130,9	133,7	48,2 r		
6	<input checked="" type="checkbox"/>	behance.net	http	200	1614		9	6	47,0 mm	2,5 mm	2	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	107,7	91,1 r	168,2	116,9		
7	<input type="checkbox"/>										<input checked="" type="checkbox"/>	light		<input type="checkbox"/>							
8	<input type="checkbox"/>										<input checked="" type="checkbox"/>	light		<input type="checkbox"/>							
9	<input type="checkbox"/>	behance.net	http	2	722	0	4	3	14,6 mm	3,3 mm	4	<input checked="" type="checkbox"/>	regular	sans-serif	<input checked="" type="checkbox"/>	108,0	17,8 r	132,0	17,3 r		
10	<input type="checkbox"/>	behance.net	http	2	97	0	10	3	16,0 mm	3,0 mm	1	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	72,7 r	31,5 r	111,3	12,3 r		
11	<input type="checkbox"/>										<input checked="" type="checkbox"/>	light		<input type="checkbox"/>							
12	<input type="checkbox"/>	behance.net	http	5	99	0	3	2	9,5 mm	6,7 mm	2	<input type="checkbox"/>	bold	serif	<input type="checkbox"/>	102,9	119,2	160,9	8,0 m		
13	<input type="checkbox"/>										<input checked="" type="checkbox"/>	light		<input type="checkbox"/>							
14	<input type="checkbox"/>	behance.net	http	2	41	0	11	2	15,7 mm	2,8 mm	1	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	79,8 r	37,5 r	79,0 r	15,7 r		
15	<input type="checkbox"/>	dribbble.com	http	1172	135756	0	7	2	11,5 mm	1,9 mm	1	<input checked="" type="checkbox"/>	regular	decorative	<input type="checkbox"/>	64,0 r	225,8	77,3 r	38,7 r		
16	<input type="checkbox"/>	dribbble.com	http	594	39937	0	7	1	13,8 mm	5,3 mm	2	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	103,2	63,3 r	172,3	47,7 r		
17	<input type="checkbox"/>										<input checked="" type="checkbox"/>	light		<input type="checkbox"/>							
18	<input type="checkbox"/>	dribbble.com	http	15	370	0	8	3	30,9 mm	3,8 mm	2	<input checked="" type="checkbox"/>	bold	script	<input checked="" type="checkbox"/>	107,2	214,7	152,3	50,7 r		
19	<input type="checkbox"/>	dribbble.com	http	598	51081	0	10	2	18,5 mm	3,7 mm	2	<input checked="" type="checkbox"/>	bold	serif	<input type="checkbox"/>	105,5	161,3	93,0 r	86,3 r		
20	<input type="checkbox"/>	behance.net	http	33	1475	2	8	6	29,3 mm	5,3 mm	2	<input checked="" type="checkbox"/>	bold	sans-serif	<input type="checkbox"/>	109,8	125,5	107,7	67,7 r		
21	<input type="checkbox"/>	behance.net	http	8	196	0	12	2	6,2 mm	3,4 mm	2	<input checked="" type="checkbox"/>	bold	script	<input checked="" type="checkbox"/>	61,7 r	251,1	104,7	6,2 m		

08-53.МКР.001.03.008.ПЛ

Змн.	Арк.	№ докум.	Підпис	Дата	Інтелектуальна інформаційна технологія макетування афіш	Літ.	Маса	Масштаб
					Користувацька частина датасету	Арк.	4	Аркушів 8
						ICT-18М		
Розроб.		Білецький Б.С.						
Перевір.		Мокін В.Б.						
Т. Контр.								
Реценз.		Маслій Р.В.						
Н. Контр.		Жуков С.О.						
Затверд.		Мокін В.Б.						

Модуль экспорту датасету

number	likes	views	Mark1	Mark2	countsOfI	cntF	maxSi	minSize	empty	backFillColor	title_weight	title_sb	title_It	title_X	title_Y	title_W	title_H
5	150	1614	9294	6506	7	5	48.2	2.5	3	0	0	1	1	107	tree.c	133,7	48,2
6	200	1614	12392	8674	9	6	47	2.5	2	0	0	1	1	108		168,2	116,9
9	2	722	277	194	4	3	15	3,3	4	0	1	1	0	108		132,0	17,3
10	2	97	2062	1443	10	3	16	3,0	1	0	0	1	1	73		111,3	12,3
12	5	99	5051	3535	3	2	10	6,7	2	1	0	0	1	103		160,9	8,0
14	2	41	4878	3415	11	2	16	2,8	1	0	0	1	1	80		79,0	15,7
15	1172	135756	863	604	7	2	12	1,9	1	0	1	3	1	64		77,3	38,7
16	594	39937	1487	1041	7	1	14	5,3	2	0	0	1	1	104		172,3	47,7
18	15	370	4054	2838	8	3	31	3,8	2	0	0	2	0	108		152,3	50,7
19	598	51081	1171	819	10	2	19	3,7	2	0	0	0	1	106		93,0	86,3
20	33	1475	2237	1566	8	6	29	5,3	2	0	0	1	1	110		107,7	67,7
21	8	196	4082	2857	12	2	6	3,4	2	0	0	2	0	62		104,7	6,2
25	124	5907	2099	1469	6	3	24	3,5	1	0	0	3	0	105		117,2	42,9
28	3	245	1224	857	5	2	18	5,0	4	0	0	0	1	104		79,0	53,0
29	279	30742	908	635	7	2	17	2,3	3	0	0	1	1	103		147,5	16,5
30	7	61	11475	8033	5	2	16	6,3	4	0	0	1	1	107		173,7	16,3

08-53.МКР.001.03.008.ПЛ

Інтелектуальна інформаційна
технологія макетування афіш

Літ.

Маса

Масштаб

1 : 1

Арк.

5

Аркушів

8

Модуль експорту датасету

ICT-18М

Змн.	Арк.	№ докум.	Підпис	Дата			
					08-53.МКР.001.03.008.ПЛ		
					Інтелектуальна інформаційна технологія макетування афіш		
					Літ.	Маса	Масштаб
							1 : 1
					Арк.	5	Аркушів 8
					Модуль експорту датасету		
					ICT-18М		

Приклад застосування технології

Датасет				
№	Характеристики			Оцінка
	X1	X2	X3	O
1	0	1	0	53
2	0	1	1	50
3	0	0	0	91
4	1	0	0	6
5	1	1	0	83
6	0	0	1	1
7	0	1	1	50
8	0	0	1	2

Датасет				
№	Характеристики			Оцінка
	X1	X2	X3	O
9	1	1	1	56
10	0	0	0	90
11	1	0	1	39
12	1	0	0	5
13	0	1	0	53
14	1	1	1	57
15	1	1	0	83
16	1	0	1	38

Класифікація за допомогою
дерева рішень

Класи	Характеристики			Оцінка
	X1	X2	X3	O
1	0	0	0	91
2	0	0	1	2
3	0	1	0	54
4	0	1	1	51
5	1	0	0	7
6	1	0	1	40
7	1	1	0	84
8	1	1	1	58

Введені характеристики	
X1	X2
0	1

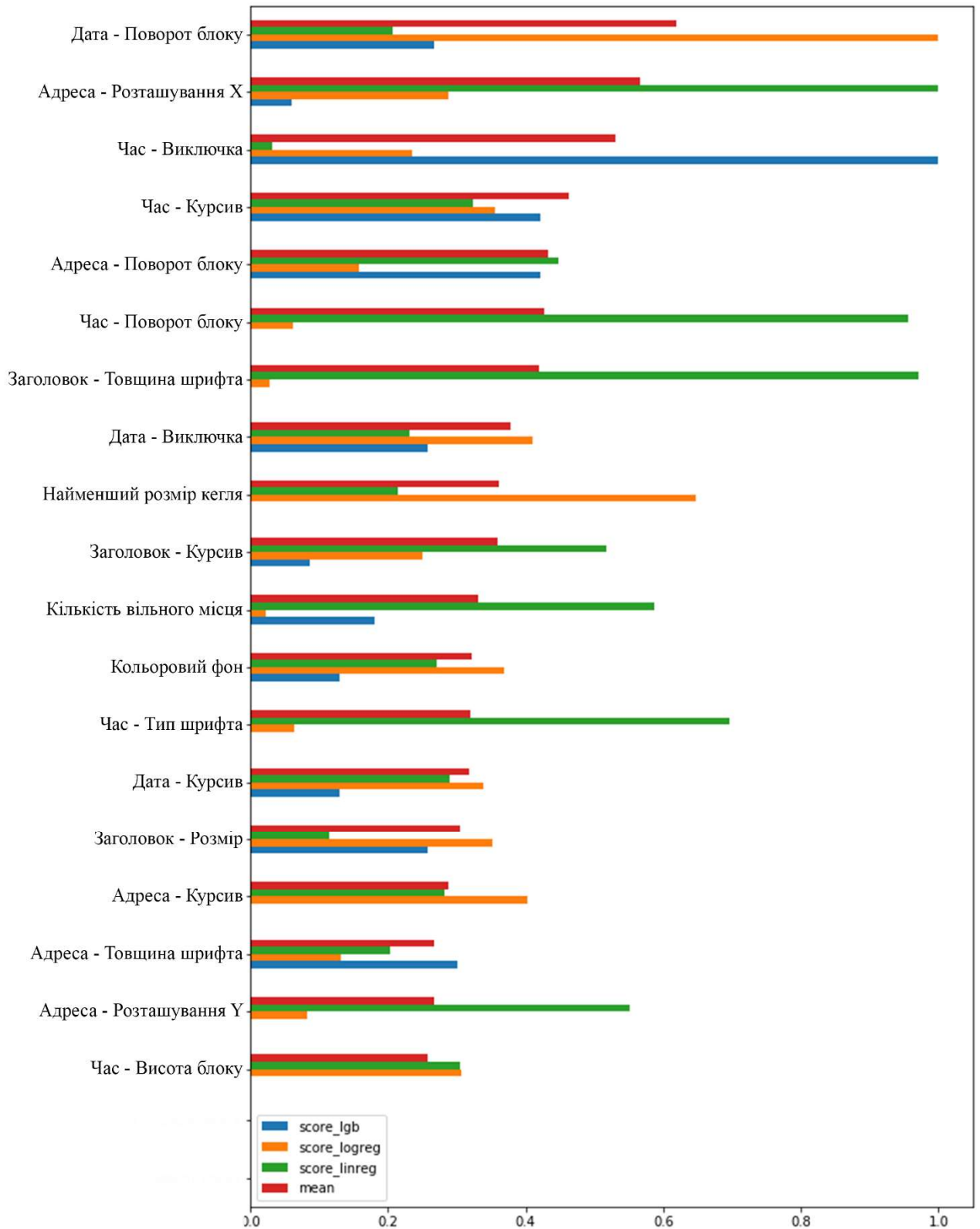
Класи	Характеристики			Оцінка
	X1	X2	X3	O
1	0	0	0	91
2	0	0	1	2
3	0	1	0	54
4	0	1	1	51
5	1	0	0	7
6	1	0	1	40
7	1	1	0	84
8	1	1	1	58

Введені характеристики		Рекомендована	Оцінка
X1	X2	X3	O
0	1	0	54

08-53.МКР.001.03.008.ПЛ

Змн.	Арк.	№ докум.	Підпис	Дата	Інтелектуальна інформаційна технологія макетування афіш	Літ.	Маса	Масштаб
Розроб.		Білецький Б.С.			Приклад застосування технології	Арк.	6	Аркушів 8
Перевір.		Мокін В.Б.						
Т. Контр.								
Реценз.		Маслій Р.В.						
Н. Контр.		Жуков С.О.						
Затверд.		Мокін В.Б.						

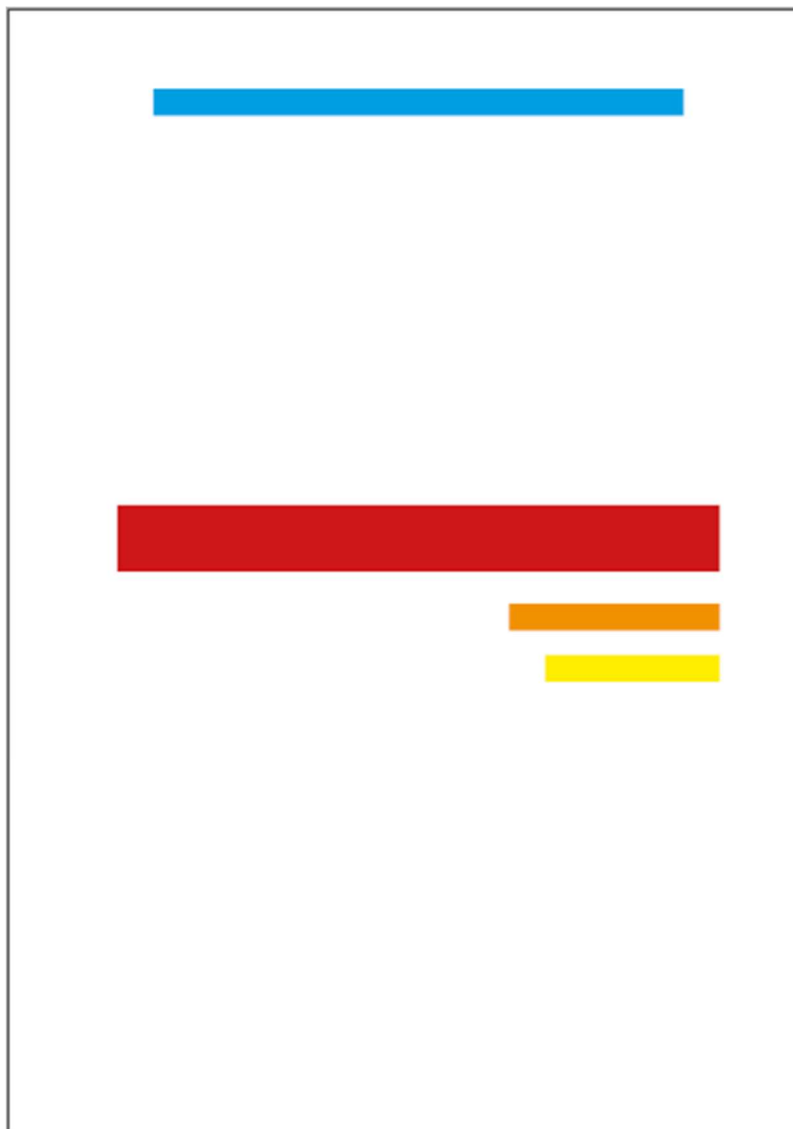
Діаграма важливості ознак



08-53.МКР.001.03.008.ПЛ

					08-53.МКР.001.03.008.ПЛ			
						Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата	Інтелектуальна інформаційна технологія макетування афіш			1 : 1
Розроб.		Білецький Б.С.						
Перевір.		Мокін В.Б.						
Т. Контр.					Арк.	7	Аркушів	8
Реценз.		Маслій Р.В.			Діаграма важливості ознак			ICT-18М
Н. Контр.		Жуков С.О.						
Затверд.		Мокін В.Б.						

Синтезована розмітка макету афіши



08-53.МКР.001.03.008.ПЛ

Змн.	Арк.	№ докум.	Підпис	Дата	Інтелектуальна інформаційна технологія макетування афіш	Літ.	Маса	Масштаб
Розроб.		Білецький Б.С.			Синтезована розмітка макету афіши	Арк.	8	Аркушів 8
Перевір.		Мокін В.Б.						
Т. Контр.								
Реценз.		Маслій Р.В.						
Н. Контр.		Жуков С.О.						
Затверд.		Мокін В.Б.						

