DOI 10.34185/1562-9945-6-155-2024-07
УДК 004.94

А.О. Бур, Т.А. Ліхоузова, Ю.О. Олійник

ПРОГНОЗУВАННЯ ПОПУЛЯРНОСТІ МУЗИЧНИХ ТРЕКІВ НА ПЛАТФОРМІ SPOTIFY НА ОСНОВІ ЧИСЛОВИХ МЕТРИК

Анотація. В сучасному світі музика відіграє важливу роль у житті мільйонів людей, а платформи для стрімінгу музики, такі як Spotify, стали невід'ємною частиною сучасної культури. Популярність музичних треків має велике значення для музичної індустрії, впливаючи на доходи артистів та тренди в музичному світі. Прогнозування популярності музичних треків є важливим завданням, яке може допомогти артистам, продюсерам та платформам краще розуміти вподобання слухачів та оптимізувати свої стратегії.

Для прогнозування популярності музичних треків застосовувались моделі Decision Tree Classifier, KNeighbors Classifier, XGBoost Classifier та Random Forest Classifier. Загальний аналіз показав, що моделі XGBoost та Random Forest є найбільш ефективними для прогнозування популярності музичних треків. Вони демонструють високу точність і стійкість до змін у наборі атрибутів, що робить їх придатними для використання у реальних умовах.

Ключові слова: інтелектуальний аналіз даних, класифікація, knn, decision tree, random forest, extreme gradient boosting.

На сьогоднішній день музична індустрія є однією з найбільш динамічно-розвиваючих секторів економіки багатьох країн. Музичні стрімінгові сервіси, а в особливості Spotify, відіграють одну з найважливіших ролей у поширенні музики в усьому світі. Саме завдяки таким платформам, користувачі мають доступ до величезної бібліотеки музичних композицій, а музиканти, в свою чергу, можуть ефективно і швидко доносити свою творчість та свої почуття до слухачів.

Популярність музичних треків значною мірою впливає на доходи артистів та музичних лейблів. Визначення популярності треків базується на кількості прослуховувань, вподобань та інших метрик, які відображають взаємодію користувачів з музикою. Успішне прогнозування популярності треків дозволяє музичним платформам оптимізувати свої рекомендаційні системи, а артистам та продюсерам — краще планувати релізи та рекламні кампанії.

Метою дослідження є створення програмного забезпечення для аналізу музичних треків на платформі Spotify, їх класифікація за рівнями популярності та подальше прогнозування рівня популярності на основі числових метрик.

© Бур, Т.А. Ліхоузова А.О., Олійник Ю.О., 2024

Матеріали та методи. Для виконання дослідження було обрано 4 джерела відкритих даних на сайті https://www.kaggle.com, а саме:

- загальна інформація про музичні треки на платформі Spotify: https://www.kaggle.com/datasets/lehaknarnauli/spotify-datasets?select=tracks.csv
- загальна інформація по виконавцям на платформі Spotify: https://www.kaggle.com/datasets/lehaknarnauli/spotify-datasets?select=artists.csv
- додаткова інформація про музичні треки з 1921 по 2020 роки: https://www.kaggle.com/datasets/ektanegi/spotifydata-19212020?select=data.csv
- інформація по прослуховуванням за датами, коли трек входив в топ 200 по прослуховуванням: https://www.kaggle.com/datasets/dhruvildave/spotify-charts?rvi=1

Для завантаження даних з датасетів до stage-зони використано скрипти, написані як на чистому SQL, так і з використанням мови програмування Python.

У моделі сховища за типом сніжинка спроектовано одна таблиця фактів та сім таблиць вимірів (рис. 1). Така модель сховища легко може бути розширена новими даними, такими як нові виконавці, нові музичні композиції чи кількість прослуховувань за певний період часу. Модель приведена до третьої нормальної форми, всі зв'язки логічні, є можливість комбінувати таблиці для отримання даних у бажаному форматі, що сильно полегшує роботу з даними: їх зберігання, зміни, обробку чи видалення. На основі даної моделі було створено SQLite [1] сховище даних. Подальша робота зі сховищем, зокрема ETL процеси, проводилася за допомогою Python [2].

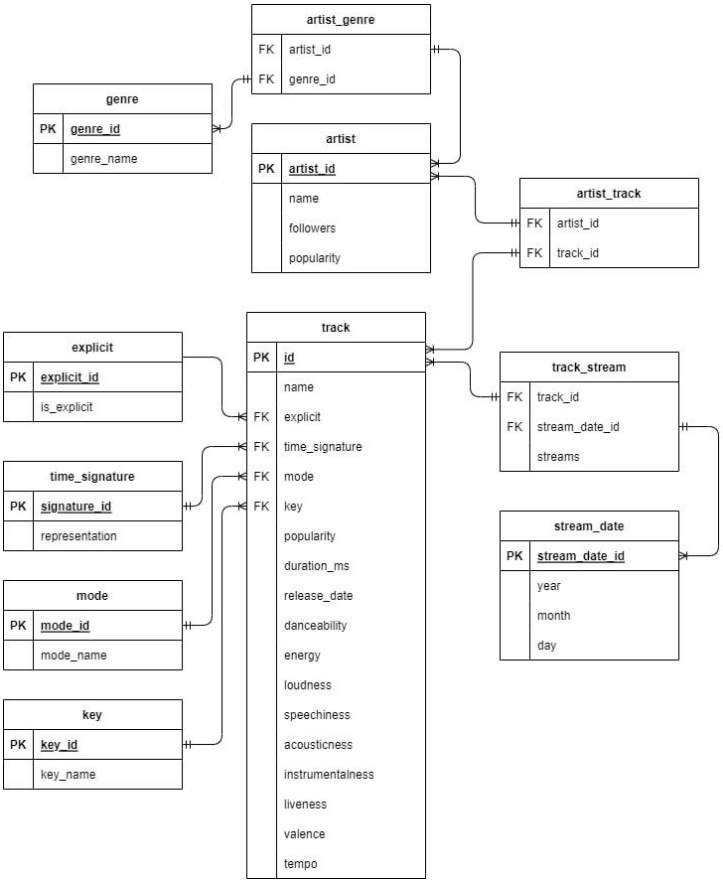


Рисунок 1 – Модель бази даних

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Дані було розбито на 3 класи за рівнем популярності (0 – зовсім непопулярний, 1 – середньої популярності, 2 – хіт) (рис.2,3).

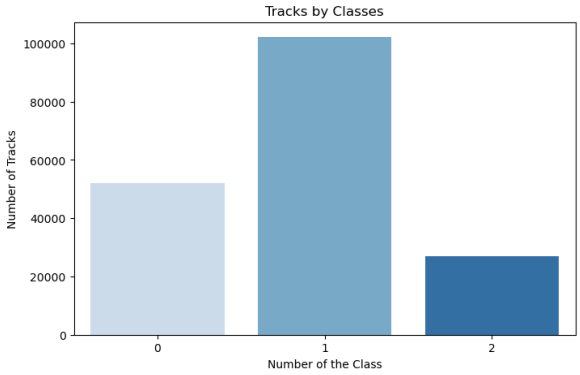


Рисунок 2 – Кількість треків для кожного з класів

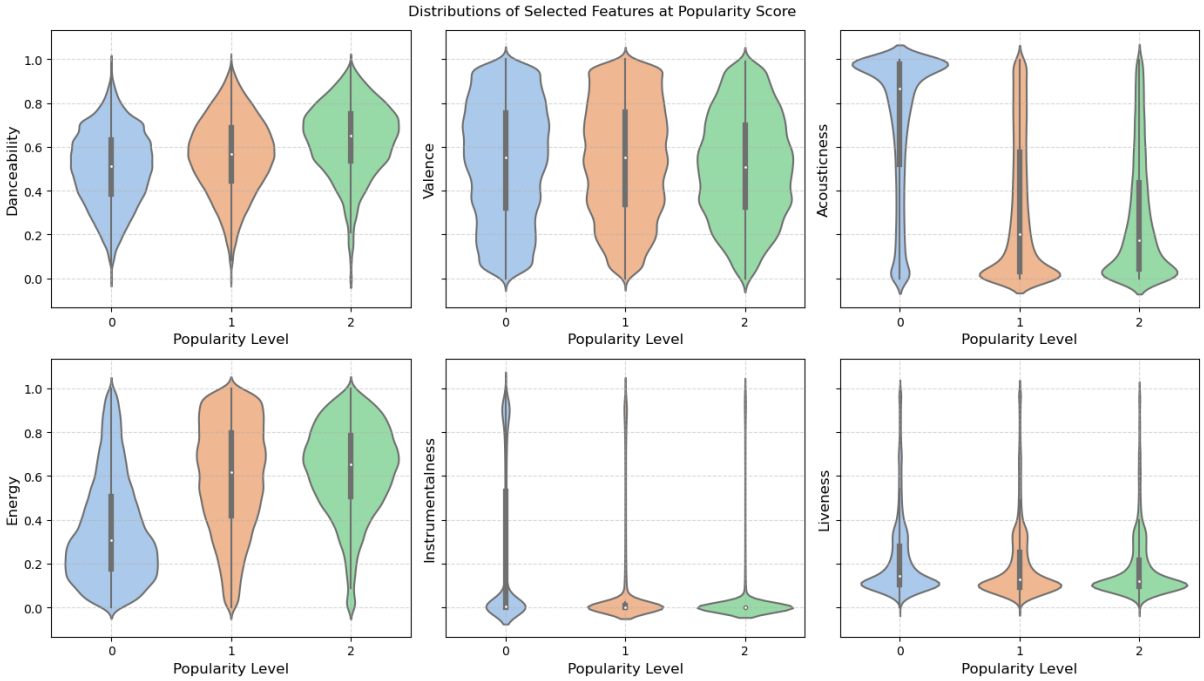


Рисунок 3 – Розподіл рівнів популярності музичних треків між метриками

Результати та основний матеріал дослідження. Для обробки інформації та виявлення в ній моделей та тенденцій для прийняття рішень використано інтелектуальний аналіз даних. Спираючись на числові характеристики інших музичних треків, за допомогою математичних методів можна оцінити майбутні показники популярності, що є актуальним для виявлення тенденцій не тільки для платформи Spotify, а й для багатьох інших музичних платформ та сервісів. В якості моделей для проведення прогнозування популярності музичних треків було обрано наступні: Decision Tree Classifier [3], KNeighbors Classifier [4], XGBoost Classifier [5, 6], RandomForest Classifier [7]. Вибір був зроблений на основі суттєвих факторів, що полегшують проведення класифікації. Дані моделі дозволяють аналізувати великі обсяги даних, виявляти складні залежності та

взаємозв'язки між характеристиками треків, такими як акустичність, темп, валентність, живість тощо.

Для тренування та оцінювання моделей була створена загальна функція, яка спрощує цей процес для різних моделей класифікації. Ця функція приймає наступні параметри: модель, її назву у текстовому вигляді, тренувальну вибірку незалежних змінних, тренувальну вибірку залежної змінної, тестову вибірку незалежних змінних та тестову вибірку залежної змінної. Далі описано кроки, які виконує ця функція:

– модель навчається на тренувальних даних;

– після навчання моделі, на тестових даних виконуються прогнози;

– виконується обчислення метрик якості моделі: accuracy, precision, recall, та f1-score. Метрики обчислюються для кожного з класів окремо, а також розраховуються середні значення для останніх трьох метрик;

– функція виводить обчислені метрики в консоль, графічно зображає їх та виводить матрицю невідповідностей (confusion matrix) для тестових та передбачених даних;

– функція додає обчислені метрики до результуючого масиву. Після тренування всіх моделей та обчислення їх метрик, загальні результати візуалізовані на графіку для порівняння ефективності моделей та вибору найкращої серед них.

Такий підхід дозволяє ефективно проводити порівняльний аналіз моделей класифікації, обраних для прогнозування популярності музичних треків. Результати моделювання представлені на рис.4-7.

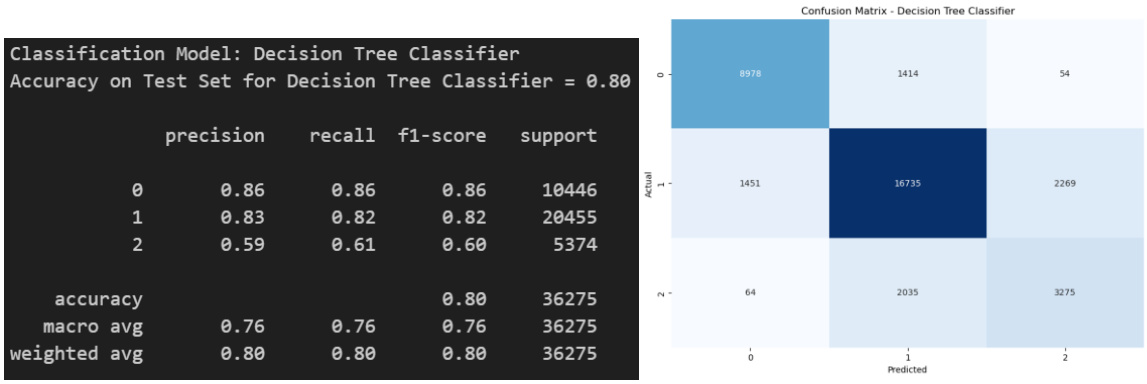


Рисунок 4 – DecisionTree Classifier

Модель показує досить хороші результати, з точністю близько 80% та значенням f1-score 0.76. Проте основною проблемою є її схильність до перенавчання, особливо на великих та складних наборах даних. Це може призводити до поганої генералізації на нових даних.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Classification Model: KNeighbors Classifier
Accuracy on Test Set for KNeighbors Classifier = 0.73

	precision	recall	f1-score	support
0	0.77	0.73	0.75	10446
1	0.73	0.82	0.77	20455
2	0.57	0.35	0.43	5374
accuracy			0.73	36275
macro avg	0.69	0.63	0.65	36275
weighted avg	0.72	0.73	0.72	36275

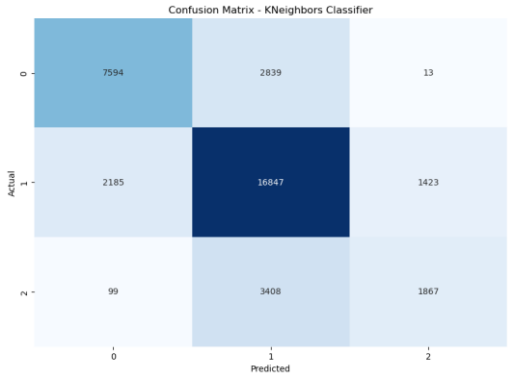


Рисунок 5 – KNearestNeighbors Classifier

Модель KNearestNeighbors Classifier показала середні результати з точністю близько 72.5% та f1-score близько 0.65. Основна проблема цієї моделі полягає в її чутливості до масштабування ознак та вибору значення параметра k (кількість сусідів). Модель може бути менш ефективною на великих наборах даних через високу обчислювальну складність.

Classification Model: XGB Classifier
Accuracy on Test Set for XGB Classifier = 0.87

	precision	recall	f1-score	support
0	0.89	0.92	0.90	10446
1	0.87	0.91	0.89	20455
2	0.83	0.62	0.71	5374
accuracy			0.87	36275
macro avg	0.86	0.81	0.83	36275
weighted avg	0.87	0.87	0.86	36275

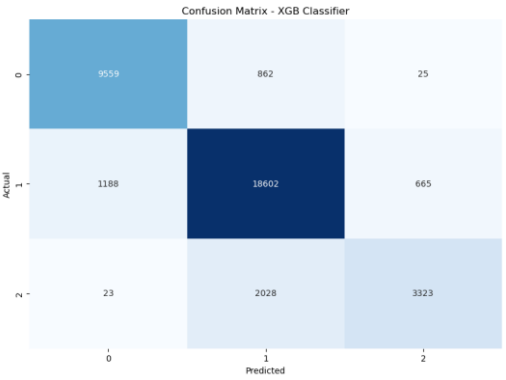


Рисунок 6 – XGBoost Classifier

XGBoost Classifier показав на даний момент найкращі результати серед усіх моделей, з точністю майже 87% та f1-score 0.81. І дійсно, модель має чудову здатність до генералізації та високу стійкість до перенавчання.

Classification Model: Random Forest Classifier
Accuracy on Test Set for Random Forest Classifier = 0.86

	precision	recall	f1-score	support
0	0.88	0.91	0.90	10446
1	0.86	0.90	0.88	20455
2	0.80	0.61	0.70	5374
accuracy			0.86	36275
macro avg	0.85	0.81	0.82	36275
weighted avg	0.86	0.86	0.86	36275

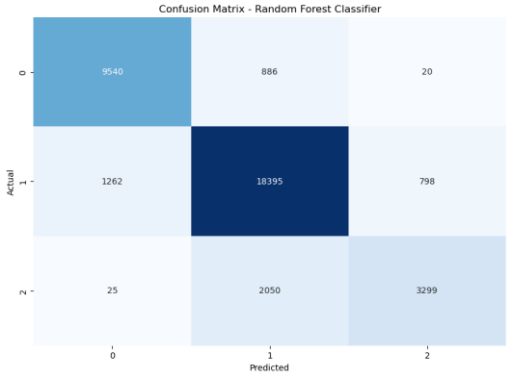


Рисунок 7 – Random Forest Classifier

Random Forest Classifier показав дуже високі результати, з точністю понад 86% та f1-score 0.81. Модель добре справляється зі складними наборами даних і є стійкою до

перенавчання. Проте, обчислювальна складність та час тренування можуть бути досить високими, особливо на великих наборах даних, порівняно з XGBoost Classifier.

Беручи за основу минулі результати було прийнято рішення про підбір параметрів для трьох моделей: KNeighbors Classifier, XGBoost Classifier, RandomForest Classifier. Чому не було включено DecisionTree Classifier замість KNeighbors Classifier? Все доволі просто, занадто велика чутливість до даних та ризик перенавчання робить цю модель поганим рішенням для задачі передбачення рівня популярності музичних треків на основі числових значень треків, адже в реальному світі сховище з музичними треками буде постійно поповнюватися, що може і буде впливати на результати моделі. Тобто навіть якщо зараз модель справляється доволі непогано, немає гарантій, що додавши ще декілька тисяч треків, вона не почне перенавчатися. Знову ж, хоча KNeighbors і не дуже підходить для великих вибірок даних, було прийнято зробити підбір параметрів, адже модель є більш стабільною за DecisionTree Classifier і в перспективі може її обігнати в результуючих метриках.

Для моделі KNeighbors Classifier основними параметрами є n_neighbors, weights, algorithm, leaf_size та p. Параметр n_neighbors визначає кількість сусідів, які враховуються при класифікації, що може впливати на точність і стабільність моделі. Вибір зважування (weights) впливає на вагомість близькості сусідів, тоді як алгоритм пошуку (algorithm) впливає на швидкість обробки даних. Параметр leaf_size визначає розмір листа, що впливає на продуктивність, а параметр p обирає метрику відстані між точками: 1 – Manhattan distance, 2 – Euclidean distance. Після підбору параметрів, були отримані наступні результати (рис.8).

```
Classification Model: Best KNeighbors Classifier
Accuracy on Test Set for Best KNeighbors Classifier = 0.75

              precision    recall  f1-score   support

           0       0.81      0.73      0.77     10446
           1       0.75      0.85      0.80     20455
           2       0.64      0.41      0.50      5374

    accuracy                           0.75     36275
   macro avg       0.73      0.66      0.69     36275
weighted avg       0.75      0.75      0.74     36275
```

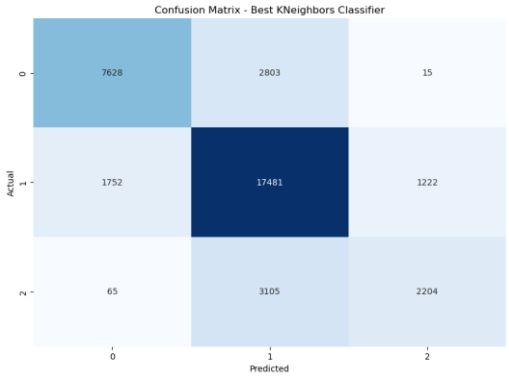


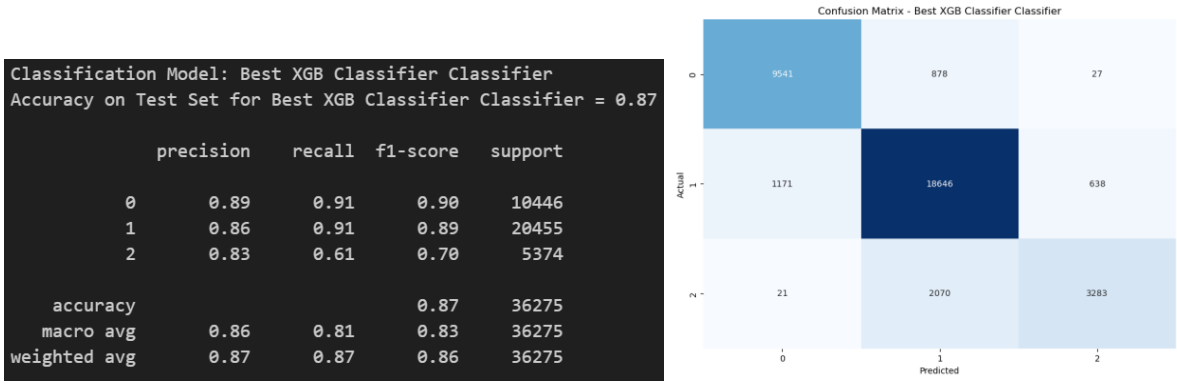
Рисунок 8 – Метрики KNeighbors Classifier з підібраними параметрами

Модель продемонструвала покращення в усіх метриках, що вказує на те, що налаштування параметрів позитивно вплинуло на її продуктивність. Але навіть у такому разі, продуктивність моделі все ще була нижчою порівняно з іншими моделями.

Для моделі XGBoost Classifier ключовими параметрами є n_estimators, max_depth, min_child_weight та gamma. Параметр n_estimators визначає кількість дерев у моделі, що покращує її продуктивність, але збільшує час навчання. Максимальна глибина дерева (max_depth) дозволяє моделювати складніші закономірності, але ризикує перенавчанням, тоді як мінімальна вага листа (min_child_weight) робить модель стійкішою до шуму. Параметр gamma регулює мінімальне зменшення втрат для розбиття вузла, що

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

робить модель більш консервативною і знижує ризик перенавчання. Після підбору параметрів, були отримані наступні результати (рис. 9):



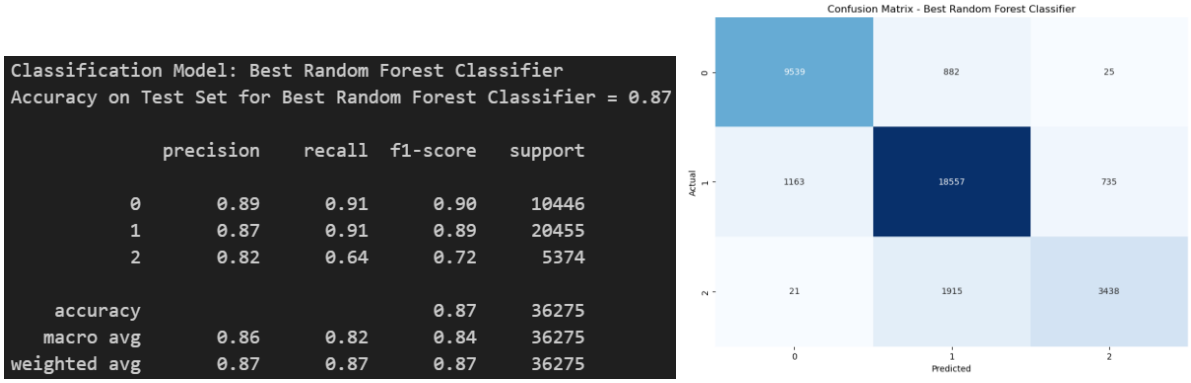
Classification Model: Best XGB Classifier Classifier
Accuracy on Test Set for Best XGB Classifier Classifier = 0.87

	precision	recall	f1-score	support
0	0.89	0.91	0.90	10446
1	0.86	0.91	0.89	20455
2	0.83	0.61	0.70	5374
accuracy			0.87	36275
macro avg	0.86	0.81	0.83	36275
weighted avg	0.87	0.87	0.86	36275

Рисунок 9 – Метрики XGBoost Classifier з підібраними параметрами

Налаштування гіперпараметрів для XGBoost Classifier також призвело до незначного покращення результатів, зробивши цю модель однією з найкращих для даної задачі. Точність склала 87%, а значення f1-score становить 0.84, що є доволі гарним результатом, враховуючи специфіку задачі.

Останньою моделлю, для якої було виконано підбір гіперпараметрів, була RandomForest Classifier. У випадку з цією моделлю, важливими параметрами є n_estimators, max_depth, min_samples_split, min_saples_leaf та bootstrap. Параметр n_estimators встановлює кількість дерев у лісі, що підвищує стабільність, але збільшує час обробки. Максимальна глибина дерева (max_depth) контролює перенавчання, дозволяючи моделі вловлювати складні закономірності або знижуючи ризик перенавчання. Мінімальна кількість зразків для розбиття вузла (min_samples_split) і мінімальна кількість зразків у листі (min_saples_leaf) впливають на гнучкість дерева, а параметр bootstrap визначає використання вибірки з поверненням для навчання кожного дерева, що впливає на різноманітність дерев. Після підбору параметрів, були отримані наступні результати (рис. 10):



Classification Model: Best Random Forest Classifier
Accuracy on Test Set for Best Random Forest Classifier = 0.87

	precision	recall	f1-score	support
0	0.89	0.91	0.90	10446
1	0.87	0.91	0.89	20455
2	0.82	0.64	0.72	5374
accuracy			0.87	36275
macro avg	0.86	0.82	0.84	36275
weighted avg	0.87	0.87	0.87	36275

Рисунок 10 – Метрики RandomForest Classifier з підібраними параметрами

Після налаштування гіперпараметрів для Random Forest Classifier, результати стали ще трохи кращими, що підкреслює високу ефективність цієї моделі для задачі кла-

сифікації популярності музичних треків. Незначне покращення присутнє в усіх метриках: accuracy, precision, recall та f1-score.

Для більш наглядного порівняння було створено графік ключових метрик, за якими проводилась оцінка моделей та ROC-крива для кожної з них (рис. 11-12):

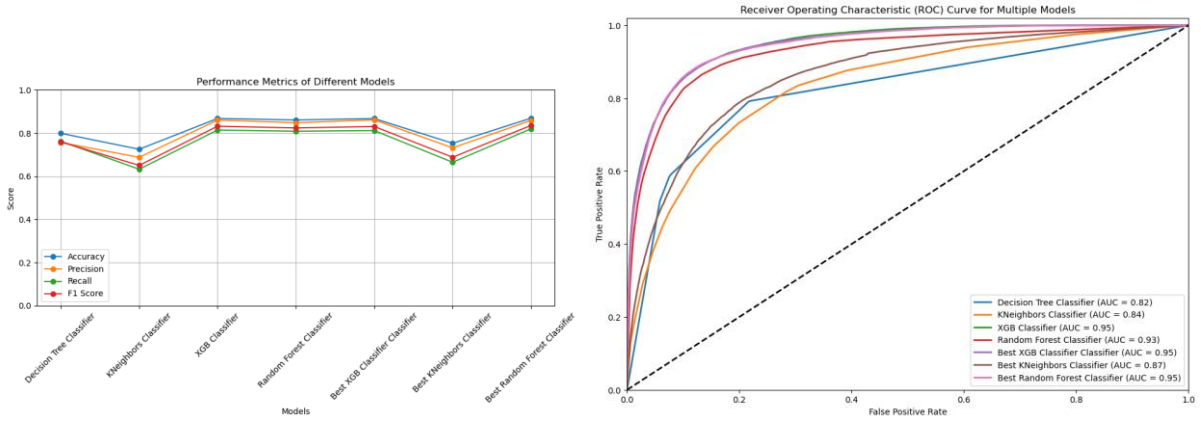


Рисунок 11 – Порівняння оцінок ключових характеристик моделей

Algorithm	Accuracy	Precision	Recall	F1 Score
Decision Tree Classifier	0.799118	0.75659	0.76234	0.759369
KNeighbors Classifier	0.725238	0.68784	0.632668	0.650443
XGB Classifier	0.867926	0.860383	0.814282	0.832014
Random Forest Classifier	0.861034	0.848267	0.808814	0.824178
Best XGB Classifier Classifier	0.86754	0.861327	0.811943	0.830737
Best KNeighbors Classifier	0.752943	0.731846	0.664987	0.688152
Best Random Forest Classifier	0.869304	0.859187	0.820043	0.835757

Рисунок 12 – Порівняння характеристик моделей

Після виконання налаштування гіперпараметрів, XGBoost Classifier та Random Forest Classifier продемонстрували найвищі показники метрик, що робить їх найефективнішими моделями для прогнозування популярності музичних треків у даній курсовій роботі.

Після цього було проведено ідентичний процес, але вже без використання атрибуту “year” для музичних треків. Цей атрибут сильно впливає на популярність треків на платформі Spotify, що є специфікою даної платформи. Після отримання гарних результатів в основній частині, стало цікаво провести схожий алгоритм дій без урахування специфіки платформи, що дозволить розширити застосування моделей на більшу кількість музичних платформ чи стрімінгових сервісів. На рис. 13 можна бачити метрики accuracy, precision, recall та f1-score для кожного з класів, для моделей DecisionTree

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Classifier, KNeighbors Classifier, XGBoost Classifier та RandomForest Classifier без урахування підбору гіперпараметрів.

Algorithm	Accuracy	Precision	Recall	F2 Score
Decision Tree Classifier	0.662578	0.605515	0.609909	0.607624
KNeighbors Classifier	0.554101	0.4446	0.421941	0.415558
XGB Classifier	0.75909	0.734379	0.670632	0.691422
Random Forest Classifier	0.737202	0.702892	0.650093	0.665126
Best XGB Classifier Classifier	0.757657	0.733415	0.668804	0.689604
Best KNeighbors Classifier	0.585665	0.506123	0.458361	0.466088
Best Random Forest Classifier	0.758401	0.741727	0.663554	0.686688

Рисунок 13 – Порівняння метрик оцінювання моделей,
без використання атрибуту “year”

З результатів видно, що при виключенні року виходу музичного треку з атрибутів, точність та f1-score моделей XGBoost Classifier та Random Forest Classifier залишаються високими. Це свідчить про їх стійкість і ефективність навіть без інформації про рік випуску треку. Натомість, моделі Decision Tree Classifier та KNeighbors Classifier демонструють помітно нижчі результати, що вказує на їхню більшу залежність від повного набору атрибутів. Таким чином, для задачі прогнозування популярності музичних треків у ситуаціях, коли рік виходу невідомий, найкращим вибором є моделі XGBoost Classifier та Random Forest Classifier.

Висновки. Загальний аналіз показав, що моделі XGBoost та Random Forest є найбільш ефективними для прогнозування популярності музичних треків. Вони демонструють високу точність і стійкість до змін у наборі атрибутів, що робить їх придатними для використання у реальних умовах.

ЛІТЕРАТУРА / REFERENCES

1. SQLite. SQLite Home Page. URL: https://www.sqlite.org/index.html (date of access: 26.05.2024).
2. Python. The official home page of the Python Programming Language. URL: https://www.python.org/ (date of access: 26.05.2024).
3. DecisionTree Classifier. scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (date of access: 26.05.2024).
4. KNeighbors Classifier. scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (date of access: 26.05.2024).

5. XGBoost Documentation. XGBoost Documentation. URL: https://xgboost.readthedocs.io/en/stable/index.html (date of access: 25.05.2024).
6. Tabulate Documentation. URL: https://pyneng.readthedocs.io/en/latest/book/12_useful_modules/tabulate.html (date of access: 26.05.2024).
7. RandomForest Classifier. scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (date of access: 26.05.2024).

Received 11.09.2024.
Accepted 13.09.2024.

Predicting the popularity of music tracks on Spotify based on numerical metrics

In today's world, music plays an important role in the lives of millions of people, and music streaming platforms such as Spotify have become an integral part of modern culture. The popularity of music tracks is of great importance to the music industry, affecting artists' incomes and trends in the music world. Predicting the popularity of music tracks is an important task that can help artists, producers, and platforms better understand listener preferences and optimize their strategies.

As part of this work, a data storage of music tracks on the Spotify platform has been developed, based on a physical model of the database, the functionality of which is implemented using SQL scripts. Working with the database is presented through the implementation of software for the implementation of ETL processes and intelligent analysis of selected data. The software allows you to classify tracks by the level of popularity (0 - not at all popular, 1 - medium popularity, 2 - hit) using numerical track metrics such as acousticness, tempo, valence, liveness, etc. The role of the data storage management system is SQLite, the programming language for implementing the application is Python.

Different machine learning models are used to predict track popularity, including KNeighbors, Decision Tree, Random Forest, and Extreme Gradient Boosting. Data mining software provides efficient track classification and graphical display, allowing users to easily interpret forecasting results. Libraries used in the work: pandas, numpy, seaborn, matplotlib, tabulate, xgboost, scipy, sqlite3.

The overall analysis showed that the XGBoost and Random Forest models are the most effective for predicting the popularity of music tracks. They demonstrate high accuracy and resistance to changes in the set of attributes, which makes them suitable for use in real conditions.

Keywords: intelligent data analysis, classification, KNeighbors, Decision Tree, Random Forest, Extreme Gradient Boosting.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Бур Антон Олександрович – студент кафедри інформатики та програмної інженерії, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Ліхоузова Тетяна Анатоліївна – к.т.н., доцент кафедри інформатики та програмної інженерії, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Олійник Юрій Олександрович – к.т.н., доцент кафедри інформатики та програмної інженерії, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Bur Anton – student, Department of Informatics and Software Engineering National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».

Likhouzova Tetiana – PhD, associate professor, Department of Informatics and Software Engineering National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».

Oliinyk Yurii – PhD, associate professor, Department of Informatics and Software Engineering National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».