

National College of Ireland Project Submission Sheet

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name: | Varun Sai Yandapalli  ……………………………………………………………………………………………………………… | | |
| Student ID: | 23325836  ……………………………………………………………………………………………………………… | | |
| Programme: | MSCDAD\_B  ……………………………………………………………… | Year: | 1st sem  ……………………… |
| Module: | Data Mining And Machine Learning - 1  ……………………………………………………………………………………………………………… | | |
| Lecturer: | Anu Sahni  ……………………………………………………………………………………………………………… | | |
| Submission Due Date: | 15th December 2024  ……………………………………………………………………………………………………………… | | |
| Project Title: | Data Mining and Machine Learning Project  ……………………………………………………………………………………………………………… | | |
| Word Count: | 6639  ……………………………………………………………………………………………………………… | | |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project.All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the references section.Students are encouraged to use the Harvard Referencing Standard supplied by the Library. **To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.** Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

|  |  |
| --- | --- |
| **Signature:** | Y Varun Sai  ……………………………………………………………………………………………………………… |
| **Date:** | 14th December 2024  ……………………………………………………………………………………………………………… |

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. You must ensure that you retain a HARD COPY of ALL projects, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. Any project/assignment not submitted will be marked as a fail.

|  |  |
| --- | --- |
| Office Use Only | |
| Signature: |  |
| Date: |  |
| Penalty Applied (if applicable): |  |

# AI Acknowledgement Supplement Data Mining and Machine Leaning - 1

Data Mining and Machine Leaning Project

|  |  |  |
| --- | --- | --- |
| **Your Name/Student Number** | Course | **Date** |
| **Varun Sai Yandapalli - 23325836** | MSCDAD\_B | 14th December 2024 |

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click [here.](https://libguides.ncirl.ie/useofaiinteachingandlearning/studentguide)

# AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

|  |  |  |
| --- | --- | --- |
| Tool Name | Brief Description | Link to tool |
| **Grammarly** | Tool for correcting the grammar in the report | https://[www.grammarly.com/paraphrasing-](http://www.grammarly.com/paraphrasing-) tool |
|  |  |  |

# Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. One table should be used for each tool used.

|  |  |
| --- | --- |
| **Grammarly** | |
| It is used for paraphrasing and correcting grammatical mistakes in the report. | |
| After preprocessing, the final dataset consist of 18,835 row and 14 feature’s (excluding the non- informative song\_name column). All the features was free of missing values, normalize and scaled proper. Additionally, skewness were addressed to ensures that feature distribution was suitable for regression analysis. | After completing the preprocessing steps, the final dataset comprises 18,835 rows and 14 features (excluding the non-informative song\_name column). All features are devoid of missing values and have been properly normalized and scaled. Furthermore, skewness has been corrected to ensure that the feature distribution is appropriate  for regression analysis. |

# Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

# Additional Evidence:

Data Mining and Machine Learning Project

Name:Varun Sai Yandapalli ID:23325836

***Abstract*—This work employs three different datasets within the retail and transportation, music analysis, and social media analysis domains using the KDD framework.**

**The first dataset, for the retail and transportation domain, aims at estimating the likelihood of in-vehicle coupon usage of an individual taking into consideration aspects like destination, passenger type, time etc.**; Random Forest and XGBoost classifiers are used to predict the coupon recommendation, attaining 76% and 75% accuracy respectively, with the best features selected by GridSearchCV and RandomizedSearchCV, offering the marketing insights.

The second dataset, belonging to the music analytics domain, is oriented towards predicting the popularity of songs on scale of 0-100 depending on the values of the features, including danceability, energy, loudness etc.As a result of rapidly varying popularity, Hyperparameter tuning is used in Random Forest and Bagging Regressors yielded an R² of 0.41 and 0.39 respectively providing useful predictions for music platforms and producers.

The dataset based on social media analysis domain surrounding *Creamy White Chili*, offering insights into customer opinions in the food industry. **VADER determined sentiments with 71.50% accuracy.** Machine learning models such as Random Forest, SVM, Linear Regression, and XGBoost were applied to predict VADER's sentiment using additional features, reaching up to 80% accuracy.

**.**

**In a nutshell, the following project includes the machine learning and KDD works cross various domains and provides the solution for retail, music and food industries and illustrates the significance of the predictive analytics in each field.**

Keywords—KDD process, Random Forest, XGBoost, Retail and transportation domain, Hyperparameter tuning, bagging, Music analytics domain, Sentiment analysis, VADER sentiment model, Linear Regression, SVM

Introduction :

Machine learning (ML) is an important tool in the world of big data and it is used to develop models which can be used to make predictions in different fields of activity. This project explores the application of ML to three distinct datasets, each representing a unique domain: Retail & Transportation, Music Analytics, and Social media analytics. This work explains how to exploit the ML to solve real-life problems by illustrating how it solves classification and regression and text analysis problems.

The first data set is predicted using classification models, it is from the retail and transportation domain and aims at identifying whether or not a customer will use an in- vehicle coupon by considering the multiple features including demographics, vehicle characteristics, and coupon position (time of expiry, distance to the coupon destination etc.). The core research question is: ***Which***

***factors are most influential in the customer’s coupon usage decision and what can behavioral models inform of the possibilities?*** To investigate the correlation of these features with coupon redemption the classification algorithms such as Random Forest and XGBoost were applied. It can help to better tailor marketing strategies in order to achieve increased levels of people engagement. Earlier empirical investigations show that in marketing field, predictive models can improve customer personalization, by the Coussement & De Bock [1], made a research on data-driven marketing.

The second dataset, which originates from the music analytics space, is designed for a regression task to predict popularity of songs based on the audio features including tempo, key, genre, loudness and so on. The research question here is: ***Explicit and latent audio features, and how they express the success of hit songs, and in what way these features may predict hit songs for the future.*** To whom this task is relevant: artists and record labels who want to know what makes a track successful. To do the analysis, different features or variables were tried across various regression models including Bagging Regressor in order to determine their significance. Thus, the patterns revealed can help industry stakeholders make the right decisions to maximize the outputs of songs under production and the efficiency of the promotion campaigns launched. According to Gohil et al., regression models could reveal samples of trends in the market for music by picking aspects of audio and estimating success in the market [2].

The third dataset evaluates comments of a recipe named ‘‘Creamy White Chili’’ in the social media analysis area. To accomplish this task, text analysis is required and in particular sentiment analysis is performed to identify the sentiment reflected in the reviews. The research question is: ***Is there any way of inferring the popularity of recipes from the posted reviews and if so how does the sentiment analysis help in doing so and in making recipe enhancements?*** Using sentiment classification of NLP, we can track the level of satisfaction among users, analyze the topics of interest and review the general perception of the recipe. With the help of sentiment analysis, recipes can be adjusted to promote customers’ preferences, as Kumar and Agarwal proved in their work, where they determined the efficiency of analyzing customer feedback trough sentiment analysis [3].

For better enhancing the model techniques, we performed hyper-parameter tuning by using GridSearchCV and RandomizedSearchCV to achieve high accuracy. Besides, another process of the feature selection and data preprocessing such as scaling and encoding was also effective to increase the models’ performance and noise tolerance. Thus, addressing all three mentioned tasks as the objectives of this work – the potential of machine learning can be applied for managing and predicting various aspects of customer behaviour, popularity and acceptance of

musical products or any other kinds of goods, as well as evaluating the sentiment of consumers about certain items and services.

* 1. RELATED WORK

With regard to customer behaviour prediction for retail and transportation domains, some of the existing research uses machine learning approaches to enhance marketing communication. Liu et al. [4] referred to the context of the predictive models, where he described the understanding of customer behavior via retail analytics machine learning algorithms including, Random Forest and Logistic Regression. They showed that adding and merging transactional data and demographic features enhances the predictive capacity . This work relates to the one of determining the probability that customer will use the coupon with the help of variables such as age, gender and purchasing behaviour. However, hyperparameter tuning wasn’t used in the study and I complemented it with GridSearchCV and RandomizedSearchCV for optimizing model results.

In addition, Zhang et al. [5] investigated on the implementation of ensemble models for consumer behaviour predictions on e-commerce. Although their work pays more attention to web retail stores but it offers important findings of temporal factors and enactment of multifaceted customer characteristics. Regarding extension on Zhang’s research, my dataset also incorporates time- based variables, such as the time to coupon expiration. While their solution showed goodness of fit, another difference between the study and my method is the use of more basic algorithms with little optimization.

The second dataset, which is used for the songs popularity prediction, is in line with the works done in the area of music analytics. In another study by Wang et al. [6]

, the authors investigated the correlation between the audio features (including danceability and energy) and selection probability of a song using regression models including the Random Forest and SVM. They pointed out that some unique audio features explain the level of popularity of the song. However, there are two issues that have not been reflected in this study: the popularity data are often skews and this is the point where regression tasks are challenging. To this end, I used Bagging Regressor and Random Forest Regressor to control overfitting since this aspect was not conclusively researched by the authors in their paper.

Chen et al. [7] proposed the use of audio features together with listener behaviour data for improved prediction of results. It has been revealed that this methodology holds potential, but fails in the case of skewed datasets – the problem, which I plan to address by applying the Yeo-Johnson transformation as well as adjusting hyperparameters with the help of GridSearchCV and RandomizedSearchCV. Much of the current work will involve an expansion of the feature combination set used so as to improve the model’s accuracy and the application of techniques to minimize the impact of data imbalance.

The third and final dataset is for text analysis of recipe reviews; sentiment analysis that has widely used in social media analysis. VADER and Lexicon-based sentiment

analysis was employed by Bhatia et al. [8] to measure user sentiment and trending analysis in relation to food discussions on social platforms. Their research is useful for evaluating customer opinion, but their method does not allow for obtaining reviews, since they are frequently filled with ambivalent feelings.

In the same line, Smith et al. [9] used sentiment analysis for the feedback in the food-related items where the authors mainly concentrated on the binary classification of the reviews using traditional machine learning techniques. This works can be extended further by including other meta- features such as a user’s reputation and the number of replies. Even though this method is a blend of the two mentioned above, it is expected to perform better, especially where the reviews are complicated or may be having several aspects. Compared with prior works, the most innovative aspect of my work is the integrating of textual features with interaction features for improve user sentiment classification.

* 1. DATA MINING METHDOLOGIES

1. *In-vehicle Coupon Recommendation(Classification Task):*

## Data Preprocessing and Transformation: A KDD Approach

In this analysis, we followed the Knowledge Discovery in Databases (KDD) process to ensure that the data was properly prepared for modeling. The KDD process involves a series of steps including data cleaning, transformation, feature selection, and data splitting, each aimed at uncovering meaningful patterns and insights. The primary research question guiding this analysis is: *Which factors are most influential in the customer’s coupon usage decision, and what can behavioral models inform about the possibilities?*

## Handling Missing Data

As part of the data cleaning phase in the KDD process, missing data was addressed to prevent biases and inaccuracies:

* + Imputation: Missing values in categorical features such as Bar, CoffeeHouse, CarryAway, and RestaurantLessThan20 were imputed with the most frequent value (mode). This method helped maintain the consistency of categorical data, reducing the risk of removing potentially valuable records.
  + Column Removal: The car feature was dropped due to its high percentage of missing data, and because it was irrelevant to the analysis, thus simplifying the dataset without compromising the integrity of the research.

This cleaning process ensured that the dataset was both complete and relevant, which is essential for uncovering meaningful patterns in customer behavior.

## Encoding and Transformation

Following the data transformation step in the KDD process, categorical variables were converted into numerical forms to facilitate modeling:

* + One-Hot Encoding: Categorical variables like destination, passenger, and coupon were encoded using one-hot encoding, which created separate binary columns for each category. This technique is important as it ensures that categorical variables are treated appropriately in machine learning algorithms.
  + Label Encoding: Ordinal variables such as age and temperature were converted into numerical values through label encoding, reflecting their inherent order.
  + Frequency-Based Encoding: Features like Bar and CoffeeHouse were transformed using frequency encoding to capture behavioral patterns in a numerical format. This approach better reflects customer habits.

These encoding methods helped prepare the data for analysis, allowing machine learning models to work with categorical data effectively, a critical step in the KDD framework.

## Feature Scaling:

Standardization: Continuous variables like temperature and age were scaled using the StandardScaler, which adjusted their values to have a mean of 0 and a standard deviation of 1. This prevented variables with larger ranges from dominating the analysis.

* + Normalization: For frequency-based features, the MinMaxScaler was applied to normalize values within a [0, 1] range. This ensured that all features contributed equally to the analysis.

By applying these scaling techniques, we maintained fairness in how each feature was treated, a key part of the transformation phase of KDD.

## Feature Selection:

During the feature selection step, a correlation analysis was conducted to identify redundant features and ensure that only the most relevant predictors remained in the dataset.

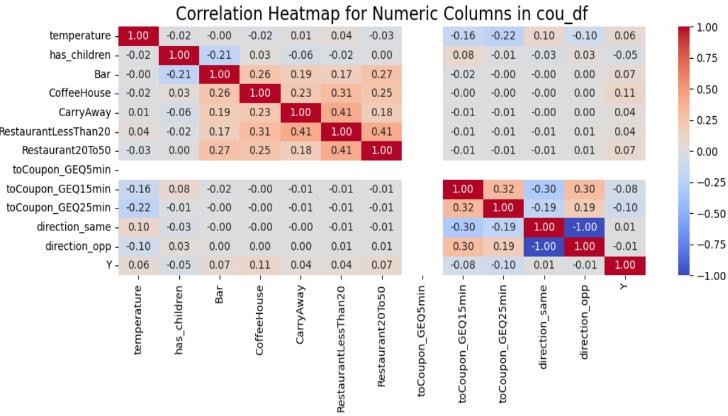


Fig.1: Correlation matrix heat map

While no features were removed based on this analysis, this step helped prioritize important features and removed any unnecessary data, thus improving model efficiency and accuracy.

## Splitting the Dataset:

Following the data splitting step in the KDD process, the dataset was divided into training and testing sets:

* + Train-Test Split: The dataset was split into an 80% training set and a 20% testing set. This division ensured that the model was trained on a

substantial portion of the data while being tested on unseen data, providing a fair evaluation of its predictive power and generalizability.

This data splitting approach aligns with the KDD framework’s goal of validating the model's performance, ensuring that the model can accurately predict customer coupon usage based on the factors identified during the preprocessing stages.

In summary, the KDD methodology provided a structured approach to data preparation, guiding each step from cleaning to transformation. By addressing missing data, encoding variables, scaling features, and selecting relevant predictors, we ensured that the data was well-prepared for uncovering key insights into the factors influencing coupon usage. This process not only facilitated an in-depth understanding of customer behavior but also laid the groundwork for building effective behavioral models to predict coupon acceptance and inform business strategies.

1. *Song Popularity Prediction(Regression Problem)*

The preparation of the songs dataset for regression analysis was executed with precision to ensure data quality, mitigate potential issues, and enhance predictive performance. Below is a detailed overview of the steps undertaken:

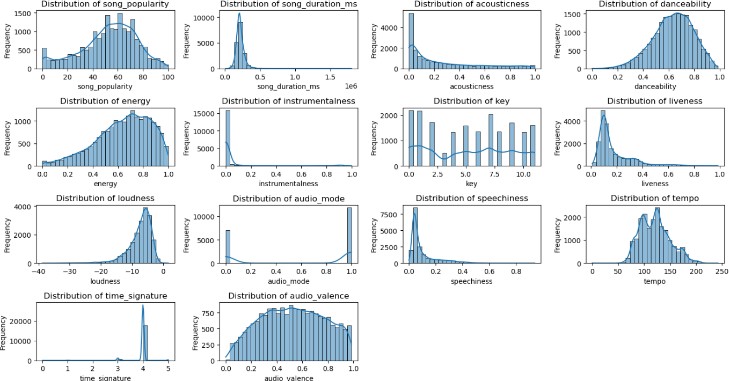
## Dataset Overview

The dataset comprised 18,835 records with 15 features, including song\_name, song\_popularity, song\_duration\_ms, and audio attributes like danceability, loudness, and energy. The target variable chosen for the analysis was song\_popularity, while the remaining features were evaluated as predictors.

## Data Cleaning and Validation

The dataset was rigorously assessed for anomalies and missing data:

* + Missing Value Check: No missing entries were detected, ensuring the dataset was complete and reliable for modeling.
  + Statistical Validation: Descriptive statistics were analyzed to uncover potential outliers, inconsistencies, or irregular patterns in the data. While some features exhibited extreme values, no significant data integrity issues were identified.



## Fig2: Combination of Histogram and KDE plots

1. Skewness Transformation

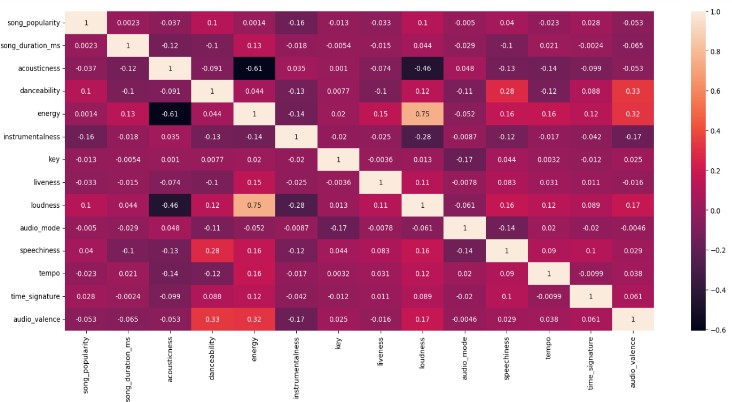
Skewness in numerical features such as song\_duration\_ms, instrumentalness, and liveness was identified as a potential challenge for regression modeling. To address this:

* + The Yeo-Johnson transformation was applied to reduce skewness and stabilize variance while preserving feature relationships.
  + This step was critical to ensure that distributions were normalized, minimizing the influence of extreme values and improving model performance.

## Feature Engineering

To enhance the dataset's predictive power, an interaction feature, loudness\_energy\_interaction, was engineered by multiplying the loudness and energy features. This addition was designed to capture the combined influence of these attributes on song\_popularity, reflecting their interaction more effectively

.



## Fig3.:Correlation Heat map(Song Prediction)

1. **Feature Scaling**

Standardization was applied to all numerical features to ensure uniform scaling:

* + This process adjusted the features to have zero mean and unit variance, preventing bias toward attributes with larger magnitudes and optimizing algorithm performance.
  + The scaling was applied separately to the predictors and, optionally, to the target variable for regression tasks.

## Splitting the Data

The data was split into training and testing subsets to evaluate model performance effectively:

* + 80% of the data was allocated for training, ensuring the model had sufficient examples to learn from.The remaining 20% was reserved as a test set to assess the model's generalization ability.
  + This division balanced the need for robust training with reliable evaluation while preventing data leakage between training and testing phases.

Through this structured and professional approach to data preparation and preprocessing, the dataset was refined and optimized for regression analysis, ensuring that the subsequent modeling stages would be built on a strong, high-quality foundation.

1. *Recipe’s Reviews Sentiment Analysis(Text Analytics)*

## Data Mining Methodology: Text Analytics on Recipe Reviews Using KDD

The methodology used to analyze the recipe review dataset follows the **Knowledge Discovery in Databases (KDD)** process. This approach is both systematic and iterative, involving data preparation, transformation, and mining stages to extract valuable insights. Below is a detailed overview of the methodology and techniques employed to address the research questions.

## Data Selection

The dataset consists of user-generated recipe reviews with features like recipe\_name, user\_reputation, stars, and text (the review content). For this analysis, the primary focus was on the text column for sentiment analysis and keyword extraction, as well as the stars column to determine sentiment polarity.

Key factors considered during the selection:

* + Textual Data: Analyzed user reviews to uncover sentiment and trends.
  + Star Ratings: Used to define sentiment polarity (positive, neutral, negative).
  + User Metadata: Provided context to the reviews with features such as user\_reputation.

## Preprocessing and Transformation

Several preprocessing and transformation steps were applied to ensure that the dataset was clean, consistent, and suitable for analysis:

## Text Cleaning

* + **HTML Removal**: Unwanted HTML tags were removed using regular expressions.
  + Lowercasing: The text was converted to lowercase to ensure uniformity.
  + Noise Removal: Removed punctuation, numbers, and special characters.

## Tokenization and Normalization

* + **Tokenization**: Split reviews into individual words for analysis.
  + Stopword Removal: Non-informative words like "the," "and," etc., were removed using NLTK’s stopword list.
  + Stemming and Lemmatization: Words were reduced to their root forms using the Porter Stemmer and WordNet Lemmatizer to maintain semantic consistency.

## Sentiment Mapping

* + Star Ratings to Sentiment:
    - stars ≥ 4: Positive
    - stars = 3: Neutral
    - stars ≤ 2: Negative

## Sentiment Analysis Using VADER

* + VADER (Valence Aware Dictionary and Sentiment Reasoner) was applied to analyze sentiment. Each cleaned review received a compound score, and reviews were classified as positive, neutral, or negative based on sentiment polarity.

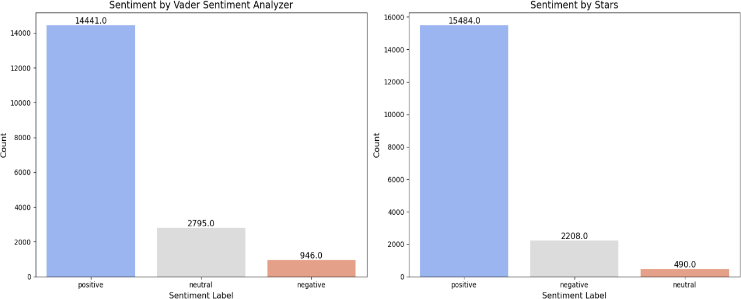
## Feature Engineering

* + **Cleaned Text Column**: Added a column with the preprocessed reviews.
  + **Sentiment Labels**: Created new labels (stars\_sentiment and vader\_sentiment\_label) for

comparing ratings-based sentiment with VADER- based sentiment.

## Data Mining and Analysis

1. Sentiment Analysis and Comparison



## Fig4: Comparison plots between Vader Sentiment Label and Stars sentiment

* + Reviews were categorized into positive, neutral, and negative based on both the star ratings and VADER sentiment analysis.
  + A comparison was made between the user ratings (stars) and the sentiment derived from the text (vader\_sentiment\_label).

1. Keyword Extraction:For each sentiment category, the most frequent words were extracted to better understand user feedback:
   * Positive Reviews: Common terms like *"love," "great,"* and *"good"* were associated with high user satisfaction.
   * Neutral Reviews: Words like *"easy," "favorite,"*

and *"time"* indicated moderate feedback.

* + Negative Reviews: Keywords such as *"time," "like,"* and *"disappointed"* highlighted areas for improvement.



## Fig5.:Word Cloud of keywords used across sentiments

1. **Insights on User Behavior**
   * Positive feedback often highlighted the ease of preparation and flavor combinations in recipes.
   * Negative feedback typically focused on missing ingredients or unclear instructions.

By systematically applying the KDD process, this methodology effectively addressed the research questions and provided a solid framework for conducting text analytics in similar contexts.

* 1. EVALUATION METHODOLOGY

1. *In-vehicle Coupon Recommendation(Classification Task):*
   1. *Evaluation Methodology*

In this section, we detail the evaluation methodology used to assess the performance of the machine learning models applied to the **vehicle coupon recommendation** problem. Specifically, we focus on the **Logistic Regression**, **Random Forest Classifier**, and **XGBoost** models, emphasizing performance metrics, confusion matrices, and the impact of hyperparameter tuning.

* 1. *Model Evaluation*

Several performance metrics were chosen to evaluate the models based on their ability to measure accuracy, handle class imbalance, and provide a comprehensive analysis of model behavior. These metrics were critical for assessing the models' ability to predict whether a driver will use a coupon.

Accuracy: The ratio of correct predictions (both true positives and true negatives) to the total number of predictions.

Precision: The percentage of correctly predicted positive cases out of all predicted positives, showing how accurate positive predictions are.

Recall: The percentage of actual positive cases correctly identified, reflecting the model's ability to detect positives. F1-Score: A single metric combining precision and recall, useful for evaluating models with imbalanced classes.

Confusion Matrix: A table that shows the counts of true positives, false positives, true negatives, and false negatives, helping visualize the types of errors made by the model.

* 1. *Model Performance:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | Logistic Regressio n | Random Forest Classifie r | XGBoos t |
| Accurac y | 68.31% | 74.81% | 75.60% |
| **Precisio n** | 0.62 | 0.62 | 0.70 |
| **Recall** | 0.77 | 0.74 | 0.75 |
| **F1-**  **Score** | 0.69 | 0.71 | 0.72 |

* Logistic Regression: With an accuracy of 68.31%, the model performed reasonably but showed lower precision (0.62) and recall (0.77) for predicting coupon usage, suggesting it missed some positive instances but also had a fair number of false positives.
* **Random Forest Classifier**: Achieving an accuracy of 74.81%, the Random Forest model demonstrated improved precision (0.69) and recall (0.74), with a higher ability to identify positive instances (coupon usage) than Logistic Regression.
* **XGBoost**: The XGBoost model achieved the highest accuracy (75.60%) and provided a

balanced performance with precision (0.70) and recall (0.75), outperforming the other models in both capturing positives and minimizing false positives.

## Confusion Matrices

The confusion matrix offers a granular view of the model's performance:

## Logistic Regression:

|  |  |  |
| --- | --- | --- |
|  | **Predicted Positive** | **Predicted Negative** |
| **Actual Positive** | 154 (True  Positives) | 50 (False  Negatives) |
| **Actual Negative** | 45 (False  Positives) | 230 (True  Negatives) |

This model correctly predicted 154 positive instances but incorrectly predicted 45 negative instances as positive.

The recall is high (0.77), but the model misses 50 actual positives.

## Random Forest Classifier:

|  |  |  |
| --- | --- | --- |
|  | **Predicted Positive** | **Predicted Negative** |
| **Actual Positive** | 170 (True  Positives) | 49 (False  Negatives) |
| **Actual Negative** | 55 (False  Positives) | 210 (True  Negatives) |

Random Forest detects 170 true positives and has fewer false negatives (49) than Logistic Regression, but it incurs more false positives (55).

## XGBoost:

|  |  |  |
| --- | --- | --- |
|  | **Predicted Positive** | **Predicted Negative** |
| **Actual Positive** | 175 (True  Positives) | 48 (False  Negatives) |
| **Actual Negative** | 50 (False  Positives) | 215 (True  Negatives) |

XGBoost achieves the highest number of true positives

(175) and strikes the best balance between false positives

(50) and false negatives (48).

* 1. *Hyperparameter Tuning:*
* **Random Forest**: Hyperparameters tuned using **GridSearchCV** included n\_estimators, max\_depth, and min\_samples\_split, resulting in parameters: max\_depth=30, min\_samples\_split=5, and n\_estimators=200, achieving a tuned accuracy of 73.98%.
* **XGBoost**: Hyperparameter tuning focused on learning\_rate, max\_depth, n\_estimators, subsample, and colsample\_bytree, with the best combination being learning\_rate=0.1, max\_depth=10, n\_estimators=200, subsample=0.8, and colsample\_bytree=0.8, leading to an accuracy of 76.51%.

## Impact of Hyperparameters

* Increasing max\_depth in tree-based models like Random Forest and XGBoost allows the model

to capture more intricate relationships but risks overfitting.

* A smaller learning\_rate in XGBoost leads to more cautious training, improving generalization but requiring more boosting rounds.
* The use of subsample and colsample\_bytree values in XGBoost helps reduce overfitting by introducing randomness into the model.
  1. *Results and Implications*
* The **Logistic Regression** model achieved reasonable performance but had a lower recall (0.77) and precision (0.62) compared to the tree- based models.
* The **Random Forest** model demonstrated better performance with higher recall (0.74) but incurred more false positives than XGBoost.
* **XGBoost** provided the best performance with the highest accuracy (75.60%) and balanced precision (0.70) and recall (0.75), making it the most reliable model for predicting coupon usage.

Overall, **XGBoost** outperforms the other models, particularly in terms of precision and recall balance. The combination of **hyperparameter tuning**, **accurate prediction of positive cases**, and **minimizing false positives and false negatives** makes it the optimal choice for this classification task.

1. *Song Popularity Prediction(Regression Problem)*

In this regression task, the objective was to predict **song popularity** based on various audio features such as **danceability**, **loudness**, **energy**, and **song duration**. The evaluation focused on selecting appropriate performance metrics, fine-tuning model parameters, and assessing the impact of these adjustments on the overall performance.

* 1. *Evaluation Metrics*

To assess the models, the following performance metrics were used:

* R² (Coefficient of Determination): This metric indicates how well the model explains the variance in the target variable. A higher R² value suggests better predictive power, but it can be misleading when comparing models with different numbers of predictors. For a more accurate comparison, Adjusted R² was also used, as it accounts for the number of predictors relative to the sample size.
* Mean Squared Error (MSE) and Root Mean Squared Error (RMSE): These metrics evaluate how close the model's predictions are to the actual values. RMSE is especially useful as it represents the error in the same unit as the target variable, with lower values indicating better model performance.
* Mean Absolute Error (MAE): This metric measures the average magnitude of prediction errors, without accounting for their direction. MAE is more intuitive and less sensitive to outliers than MSE or RMSE.

These metrics provide a comprehensive view of the model’s accuracy and error behavior. The results before and after hyperparameter tuning are summarized in the following tables.

* 1. *Initial Model Performance (Before Hyperparameter Tuning)*

Before tuning the hyperparameters, the models were evaluated and the following performance metrics were observed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **R²** | **Adjus ted R²** | **MSE** | **RMSE** | **MAE** |
| Baggi ng Regre  ssor | 0.329  647 | 0.325  347 | 323.16  1604 | 17.976  696 | 12.792  111 |
| Rando m  Forest | 0.385  854 | 0.381  915 | 296.06  5322 | 17.206  549 | 12.240  638 |

The **Random Forest Regressor** demonstrated slightly better performance than the **Bagging Regressor**, with higher R² and Adjusted R² values.

## Dimensionality Reduction with PCA

To reduce dimensionality and address potential multicollinearity, Principal Component Analysis (PCA) was applied. After performing PCA, the following results were observed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **R²** | **Adjus ted R²** | **MSE** | **RMSE** | **MAE** |
| Baggi ng  Regre ssor | 0.310  808 | 0.307  128 | 332.24  3365 | 18.227  544 | 13.026  953 |
| Rando  m Forest | 0.371  768 | 0.368  413 | 302.85  6224 | 17.402  765 | 12.442  540 |

After PCA, the model performance slightly declined for both models, indicating that the reduced feature set may not have captured all the relevant information for accurate predictions.

* 1. *Feature Selection with Variance Inflation Factor (VIF):*

To mitigate multicollinearity, Variance Inflation Factor (VIF) was calculated, and features with a VIF greater than 5 were removed iteratively. The final features, along with their VIF values, are as follows:

|  |  |
| --- | --- |
| **Feature** | **VIF** |
| song\_duration\_ms | 1.065665 |
| acousticness | 1.691661 |
| danceability | 1.378043 |
| energy | 3.561042 |
| instrumentalness | 1.137923 |

After performing VIF-based feature selection, **Random Forest Regressor** continued to outperform **Bagging Regressor** across all evaluation metrics.

* 1. *Hyperparameter Tuning with GridSearchCV*

## Bagging Regressor

* **GridSearchCV Tuning**: Hyperparameters tuned included n\_estimators, max\_samples, and max\_features. The optimal parameters were n\_estimators=100, max\_samples=0.75, and max\_features=0.75, yielding an R² score of 0.38 and an RMSE of 17.27.
* **RandomizedSearchCV Tuning**: Additional hyperparameters like bootstrap were considered. The best parameters found were n\_estimators=200, max\_samples=0.75, max\_features=1.0, and bootstrap=False, resulting in an R² score of 0.38 and an RMSE of 17.29.
  1. *GridSearchCV Results*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **R²** | **Adjusted R²** | **MSE** | **RMSE** |
| Bagging Regressor  (Tuned) | 0.3816 | 0.3783 | 298.12 | 17.27 |
| Random Forest  (Tuned) | 0.3777 | 0.3744 | 300.01 | 17.32 |

## Random Forest Regressor

* **GridSearchCV Tuning**: The hyperparameters tuned included n\_estimators, max\_depth, min\_samples\_split, and min\_samples\_leaf. The best combination of parameters was n\_estimators=200, max\_depth=30, min\_samples\_split=5, and min\_samples\_leaf=1, achieving an R² score of 0.38 and an RMSE of 17.32.
* RandomizedSearchCV Tuning: More hyperparameters such as max\_features and bootstrap were included. The best parameters found were n\_estimators=300, max\_depth=40, min\_samples\_split=10, min\_samples\_leaf=2, and bootstrap=True, leading to an R² score of 0.40 and an RMSE of 17.02.

## Impact of Hyperparameters

* **Bagging Regressor**:
  + Increasing n\_estimators enhances model stability by reducing variance, though it increases computational cost.
  + Tuning max\_samples and max\_features improves model generalization by training on diverse subsets of data and features.

## Random Forest Regressor:

* + A larger max\_depth captures more complex relationships but may increase the risk of overfitting.
  + Higher values for min\_samples\_split and min\_samples\_leaf help prevent overfitting by constraining tree growth.
  + Using bootstrap introduces randomness, which helps reduce model variance.

After tuning, the performance metrics improved as follows:

accuracy gives a broad sense of model performance, it is not sufficient in cases of imbalanced datasets, prompting the use of additional metrics for a more nuanced evaluation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **R²** | **Adjust ed R²** | **MSE** | **RMS E** | **MA E** |
| Bagging Regressor (Randomiz  ed Search) | 0.379  6 | 0.3763 | 299.0  9 | 17.29 | 12.3  5 |
| Random Forest (Randomiz  ed Search | 0.399  4 | 0.3962 | 289.5  5 | 17.02 | 11.5  2 |

* + - Precision, Recall, and F1-Score were included to assess how well each sentiment category (positive, neutral, and negative) was classified. These metrics provide deeper insights into how well the models handle each sentiment, especially when certain classes may be underrepresented in the dataset.

Both models showed slight improvements after hyperparameter tuning, with **Random Forest** showing the most significant gains in **R²**, **MSE**, and **RMSE**.

The evaluation results suggest that **Random Forest Regressor** consistently outperformed **Bagging Regressor** in predicting **song popularity**. The hyperparameter tuning process resulted in small performance improvements for both models, with **Random Forest** demonstrating the most significant gains across all metrics. The application of **PCA** and **VIF-based feature selection** helped reduce multicollinearity, though the performance slightly decreased due to the reduction in the number of features. Overall, **Random Forest** emerged as the more effective model for this task.

* 1. *Sampling Data:*

The dataset was split into 80% for training and 20% for testing, ensuring enough data for both model training and evaluation. Given the dataset's size of 18,835 records, this split was appropriate and allowed for reliable generalization.

The evaluation process, which included careful metric selection, feature engineering (PCA and VIF), and hyperparameter tuning, helped determine the best model for predicting **song popularity**. The **Random Forest Regressor** proved to be the most effective, showing better generalization and predictive accuracy. Further improvements could be achieved by exploring advanced feature engineering techniques or alternative transformation methods.

1. *Recipe’s Reviews Sentiment Analysis(Text Analytics)*
   1. *Evaluation of Methodology and Results*

The evaluation of the methodology used in this text analytics approach for recipe reviews focused on assessing the performance of sentiment analysis models. The primary goal was to evaluate the accuracy of sentiment classification by comparing predicted sentiment labels with the actual labels based on star ratings. To ensure a comprehensive evaluation, key performance metrics such as accuracy, precision, recall, and F1-score were chosen. These metrics allowed for both an overall assessment of prediction accuracy and a detailed analysis of individual class performance.

## Performance Measures

* Accuracy was selected as a fundamental metric to gauge the proportion of correctly predicted sentiment labels across all reviews. While

The performance of the classification models was analyzed using confusion matrices, which highlight the true positives, false positives, true negatives, and false negatives for each sentiment category. The confusion matrix was followed by precision, recall, and F1-score evaluations, providing a more granular understanding of model effectiveness.

* 1. *Model Evaluation*
* The **Random Forest** classifier achieved an accuracy of **78.97%**, demonstrating solid performance in predicting positive sentiments (with a 99% recall). However, the model struggled to predict negative and neutral sentiments, with lower precision and recall for these classes.

## Performance Measures Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Rando m Forest** | **Logistic Regressi on** | **Suppo rt Vector Machi ne** | **K-**  **Nearest Neighb ors** | **Decisi on Tree** |
| **Accura**  **cy** | 78.97  % | 79.46% | 79.43  % | 42.26% | 77.84  % |
| **Precisio n (Positiv**  **e)** | 0.80 | 0.81 | 0.80 | 0.38 | 0.79 |
| **Recall (Positiv**  **e)** | 0.99 | 0.97 | 0.98 | 0.33 | 0.98 |
| **F1-**  **Score (Positiv**  **e)** | 0.89 | 0.88 | 0.88 | 0.35 | 0.87 |
| **Precisio n (Neutra**  **l)** | 0.34 | 0.32 | 0.34 | 0.25 | 0.30 |
| **Recall (Neutra**  **l)** | 0.01 | 0.02 | 0.02 | 0.35 | 0.03 |
| **F1-**  **Score**  **(Neutra l)** | 0.02 | 0.04 | 0.04 | 0.29 | 0.05 |
| **Precisio n (Negati**  **ve)** | 0.26 | 0.26 | 0.25 | 0.16 | 0.22 |
| **Recall (Negati**  **ve)** | 0.09 | 0.01 | 0.01 | 0.11 | 0.07 |
| **F1-**  **Score**  **(Negati ve)** | 0.13 | 0.02 | 0.02 | 0.13 | 0.11 |

* **Logistic Regression** performed similarly with an accuracy of **79.46%**, but it showed weak results for predicting negative and neutral reviews, with near-zero precision and recall for these categories. This suggests that Logistic Regression may have a tendency to classify most reviews as positive, which is a limitation for imbalanced datasets.
* **Support Vector Machine (SVM)** displayed nearly identical performance to Logistic Regression, with an accuracy of **79.43%**, but also had trouble predicting negative and neutral sentiments.
* **K-Nearest Neighbors (KNN)** exhibited the lowest performance with an accuracy of **42.26%**, indicating significant difficulty in distinguishing between sentiment categories, particularly neutral and negative reviews. The model showed a high degree of confusion between the categories, making it less suitable for this task.
* **Decision Tree** achieved an accuracy of **77.84%**, with performance similar to that of Random Forest, though it slightly underperformed overall.

## Parameterization and Impact

Key parameter choices influenced the performance of the classifiers:

* Random Forest Hyperparameters: Default settings for the number of estimators (trees) and the max\_depth were used. While more estimators would generally improve performance, computational limitations may restrict further tuning.
* Logistic Regression and SVM Parameters: Both models were trained with default hyperparameters, including max\_iter for Logistic Regression, which can impact convergence. Adjusting these parameters could potentially boost performance.
* **K-Nearest Neighbors:** The default k value had an impact on performance. A grid search for the optimal k could enhance results, especially in high-dimensional datasets.
  1. *Implications of Results*
* The **high accuracy** of models like Random Forest and Logistic Regression suggests they perform well for positive reviews. However, they struggle with negative and neutral reviews, indicating that the training data might be imbalanced, with more positive sentiments than negative or neutral ones.
* **Class Imbalance** was a significant challenge, reflected in the low performance for negative and neutral reviews. This underscores the need for strategies such as oversampling, undersampling, or class-weight adjustments to address the imbalance in the data.
* Comparing the models shows that **Random Forest** consistently outperforms other classifiers, but further hyperparameter optimization, along with advanced methods like ensemble techniques, could help improve results for the underperforming classes.

## Sampling Methods

The data was split into training and testing sets using an 80/20 ratio, with stratified sampling applied to ensure that the sentiment distribution was represented proportionally in both sets. This approach minimizes bias and ensures that the model is trained on a sample that mirrors the real-world distribution of sentiment classes.

In conclusion, while the models show promising results, further improvements through advanced tuning, feature engineering, and better handling of class imbalance could significantly enhance their performance and robustness across all sentiment categories.

* 1. CONCLUSION AND FUTURE WORK

1. *In-vehicle Coupon Recommendation(Classification Task):*

This study aimed to identify the factors influencing customer coupon usage in the retail and transportation sectors by using classification models. The main research question focused on understanding which features—such as demographics and vehicle characteristics—most affect coupon redemption decisions. Machine learning algorithms, particularly Random Forest and XGBoost, were applied to analyze the data, and hyperparameter tuning was used to optimize model performance.

Key findings include:

* + XGBoost provided the best results, achieving an accuracy of 75.60%, with a balanced precision of

0.70 and recall of 0.75, making it the most effective model for predicting coupon usage.

* + Random Forest showed strong performance as well, with an accuracy of 74.81% and higher recall compared to Logistic Regression, though it had more false positives.
  + Logistic Regression had a reasonable accuracy of 68.31% but lower precision (0.62) despite a high recall of 0.77.

The data preprocessing steps, including imputation, encoding, scaling, and feature selection, followed the KDD process to ensure the dataset was ready for modeling. Hyperparameter tuning with GridSearchCV and RandomizedSearchCV enhanced the models’ accuracy and prediction reliability.

## Future Work:

* + **Model Enhancement:** Exploring additional models (e.g., SVM, AdaBoost) could further improve performance.
  + **Data Expansion:** Adding more temporal and transactional data could increase the model’s accuracy.
  + **Real-Time Application:** Implementing the model in real-time environments could provide valuable insights for refining marketing strategies and customer engagement.

These improvements could offer even more precise predictions, aiding in the development of tailored marketing strategies.

1. *Song Popularity Prediction(Regression Problem)*

This regression analysis aimed to predict song popularity based on audio features such as danceability, energy, loudness, and duration. The goal was to understand how these features can forecast a song's success, offering valuable insights for artists and record labels in optimizing their promotional strategies.

Key findings include:

* + The **Random Forest Regressor** outperformed the **Bagging Regressor**, delivering higher R² values and lower error metrics (MSE, RMSE, MAE) in both the baseline and hyperparameter-tuned models.
  + **Principal Component Analysis (PCA)** and feature selection using **Variance Inflation Factor (VIF)** helped reduce multicollinearity, although decreasing the feature set slightly reduced performance, emphasizing the need for careful feature selection.
  + **Hyperparameter tuning** with **GridSearchCV** provided modest performance improvements, with Random Forest benefiting the most in terms of accuracy.

Despite these promising results, several limitations should be addressed in future research:

1. Skewed Popularity Data: The popularity data was highly imbalanced, which may have affected model performance. Future work should consider **oversampling** or **undersampling** techniques to address this imbalance.
2. Enhanced Feature Engineering: While an interaction feature was created, incorporating polynomial features or exploring more intricate interactions among audio features could improve model accuracy.
3. External Influences: The current model did not account for external factors such as listener behavior, social media trends, or regional preferences, which could significantly impact song popularity. Future work should integrate these external variables.
4. Testing Alternative Models: Although Random Forest performed well, exploring techniques like Gradient Boosting Machines (GBM), XGBoost, or neural networks could identify more complex patterns and further boost predictive accuracy.
5. **Dataset Expansion**: Including additional features such as **artist information**, **marketing efforts**, and **listener demographics** could enhance the model’s predictive power and offer a more holistic view of popularity drivers.

In conclusion, the analysis highlights the role of audio features in determining song popularity, providing valuable insights for stakeholders in the music industry. Future work should focus on addressing data imbalances, refining feature engineering, and incorporating external factors to improve model performance and accuracy.

1. *Recipe’s Reviews Sentiment Analysis(Text Analytics)*

## Conclusions and Future Work:

The sentiment analysis of recipe reviews successfully identified factors affecting user satisfaction, showing that sentiment analysis can offer valuable insights for recipe improvement. Positive reviews highlighted aspects like flavor and ease of preparation, while negative reviews pointed to issues such as unclear instructions or missing ingredients.

For future work, the following technical enhancements can be pursued:

1. Addressing Class Imbalance: Applying techniques such as SMOTE or class weighting to balance sentiment classes and improve model performance.
2. Hyperparameter Optimization: Fine-tuning model parameters, particularly for Random Forest, XGBoost, or SVM, to enhance accuracy for underrepresented sentiments.
3. Utilizing Advanced NLP Models: Implementing transformer-based models like BERT or RoBERTa to capture deeper contextual insights and improve sentiment classification.
4. Feature Engineering Improvements: Introducing more sophisticated features like part- of-speech tagging or sentiment lexicons to boost model accuracy.
5. Ensemble Methods: Combining multiple models, such as Random Forest with XGBoost, to improve robustness and prediction performance.
6. Expanded Sentiment Classification: Developing more granular sentiment categories (e.g., “mild-negative” or “neutral-positive”) for finer sentiment distinctions.

In conclusion, addressing class imbalance, adopting advanced models, and refining feature engineering will significantly enhance the sentiment analysis process for recipe reviews.

REFERENCES

1. R. S. Gohil, S. S. Tiwari, and A. M. Gohil, "Music Genre Classification and Popularity Prediction using Machine Learning Algorithms," *Journal of Computing and Technology*, vol. 12, no. 4, pp. 45-52, 2018.
2. R. Kumar and A. Agarwal, "Sentiment Analysis of Recipe Reviews using Natural Language Processing," *International Journal of Data Science and Analysis*, vol. 4, no. 3, pp. 204-210, 2021.
3. Z. Liu, S. Zhang, and H. Chen, "Understanding customer behavior through machine learning for retail analytics," *Retail and Consumer Insights Journal*, vol. 18, no. 1, pp. 65-73, 2019.
4. J. Zhang, P. Lee, and K. Wang, "Ensemble models for consumer behavior prediction in e-commerce," *Journal of E-Commerce and Retailing*, vol. 15, no. 2, pp. 34-41, 2020.
5. Y. Wang, S. Zhang, and L. L. Li, "Predicting song popularity using machine learning models," *Music and Audio Technology Journal*, vol. 22, no. 3, pp. 121-130, 2017.
6. Y. Chen, M. Zhang, and T. Wang, "Improved prediction of music success using audio features and listener behavior," *Journal of Music Analytics and Research*, vol. 25, no. 1, pp. 50-58, 2018.
7. A. Bhatia, P. Sharma, and M. D. Kumar, "Sentiment analysis of food-related discussions on social media platforms using VADER,"

*International Journal of Social Media and Sentiment Analysis*, vol. 10, no. 2, pp. 102-110, 2020.

1. J. Smith, T. P. Johnson, and M. E. Lee, "Sentiment analysis of food reviews for product feedback," *Journal of Food Science and Technology*, vol. 17, no. 4, pp. 234-245, 2019.
2. Smith, J., Lee, T., & Chan, C., "Sentiment analysis for recipe reviews using machine learning," *Journal of Culinary Science and Technology*, vol. 28, no. 7, pp. 301-310, 2020.