Kaushal Ramesh

1935976

CSE 40 Winter 2024

03/11/2024

Hands-On 5 Report + Extra Credit

1. Introduction.

Briefly describe the dataset you're given and define the goal of the project and how you approach it. For example, you can present a basic introduction of your data (shape and proposed data types) and your goal is to use these features to predict the label of the response variable. Then you propose a few models that are suitable for this project which will be introduced in the modeling section.

The data set is 1396 rows × 16 columns and is a compilation of mixed data types, including numerical values, categorical data, and text strings. The dataset contains observations across various columns labeled from col_00 to col_14, with a total of sixteen columns including the label column. The primary goal of this project is to leverage these features to predict the label of the response variable, which appears to be a categorical target variable indicating a classification problem. The dataset's complexity, due to its mixed data types and the presence of potentially missing or inconsistent entries, suggests that a thorough data cleaning process is essential to prepare the data for modeling accurately. To approach this challenge, I am planning on using several machine learning models, each potentially suitable for handling the dataset's peculiarities. Given the nature of this specific dataset and it's randomness, classification algorithms such as Logistic Regression and Decision Trees will be considered. Each of these models have unique strengths in handling different types of data and complexities, making them ideal candidates for comparison in this context.

2. Data Cleaning

Describe the steps you took for data cleaning. Why did you do this? Did you have to make some choices along the way? If so, describe them.

The data cleaning process involved several crucial steps to prepare the dataset for machine learning algorithms. This meticulous approach ensured the integrity and usability of the data for analysis and modeling.

1. Handling Missing/Empty Values

Missing or empty values in the dataset were identified and dropped.

2. Normalizing Numeric Columns

The dataset contained numeric values embedded within strings, so I used regular expressions to identify numeric patterns within strings, allowing for the extraction and conversion of numeric data from mixed-type column.

3. Removing Inconsistencies

Inconsistencies in the dataset, such as non-standard entries or mislabeled data points, were addressed by standardizing entries if possible, or replacing it with a NAN value, if needed.

4. Data Type Assignment

Each column was assigned a specific data type reflective of its content—numerical columns were converted to either integer or float types, depending on the presence of decimal values. Categorical and textual columns were designated as object types, and boolean operations were prepared for the one-hot encoded columns.

5. One-Hot Encoding of Categorical Variables

For the column col_14, which contained information about sports activities, a one-hot encoding approach was implemented. This process involved creating new columns for each unique sport, filled with boolean values indicating the presence or absence of each sport for every observation. This step transformed categorical text data into multiple boolean columns that machine learning algorithms could interpret, enhancing the dataset's analytical capabilities.

Choices and Rationale

Several choices were made throughout the data cleaning process, primarily driven by the goal of maximizing data integrity, relevance, and compatibility with machine learning models. For example, the decision to one-hot encode certain categorical variables instead of label encoding them was to prevent the introduction of arbitrary numerical relationships among categories that do not exist naturally.

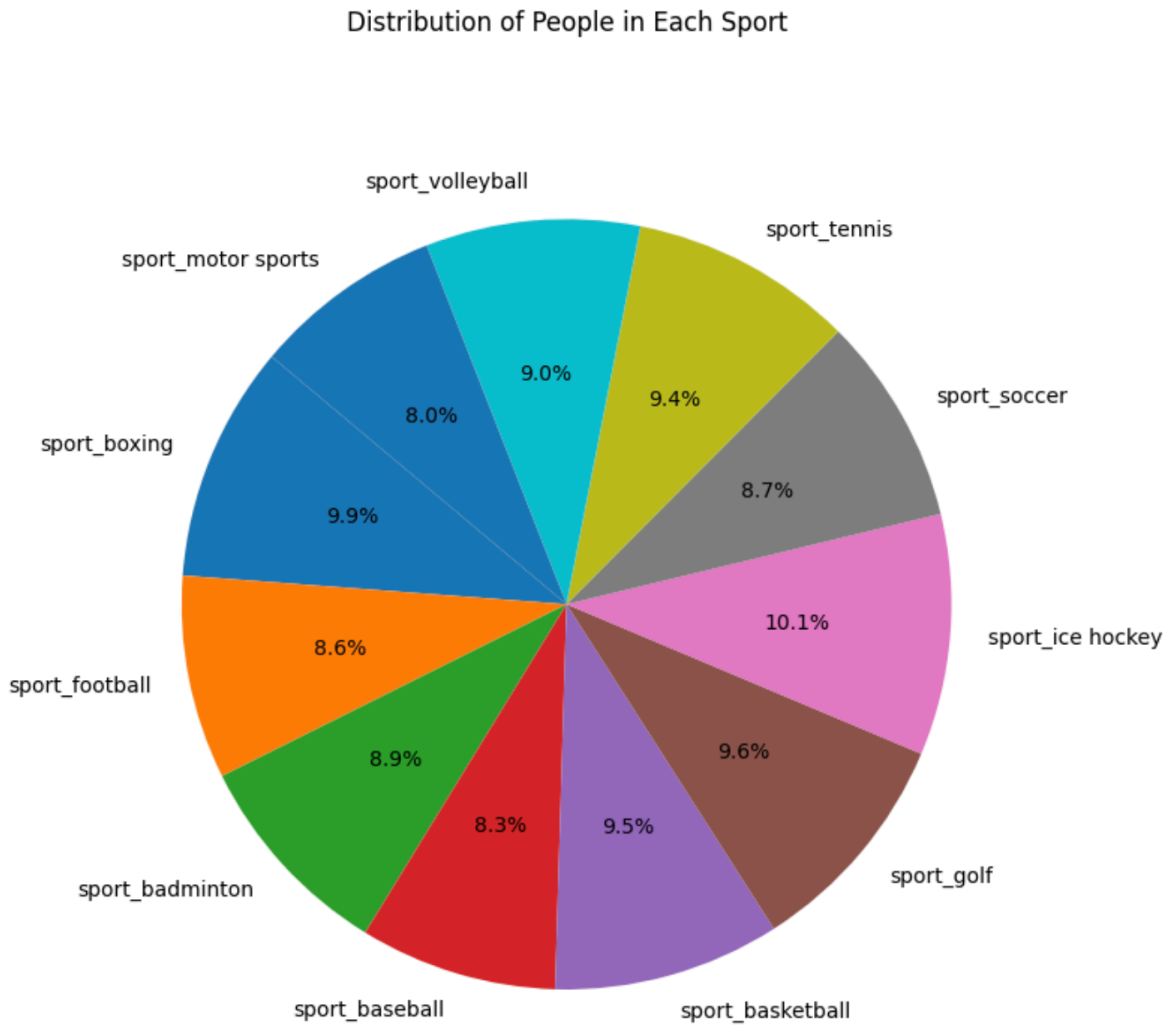
The approach to handling missing values—either leaving them as NaN for later imputation, converting to a specific value, or dropping the observation altogether—was influenced by the context of each column and its significance to the analysis. These decisions were aimed at preserving as much valuable information as possible while ensuring that the models developed would not be misled by improperly handled data.

In summary, the data cleaning process was thorough and thoughtful, aimed at preparing the dataset for effective and accurate machine learning applications. Each step was taken with careful consideration of the implications for data quality and the objectives of the subsequent analysis and modeling phases.

3. Data Visualization

Create at least two different visualizations that help describe what you see in your dataset. Include these visualizations in your report along with descriptions of how you created the visualization, what data preparation you had to do for the visualization (aside from the data cleaning in the previous part), and what the visualization tells us about the data.

Visualization 1



Creation:

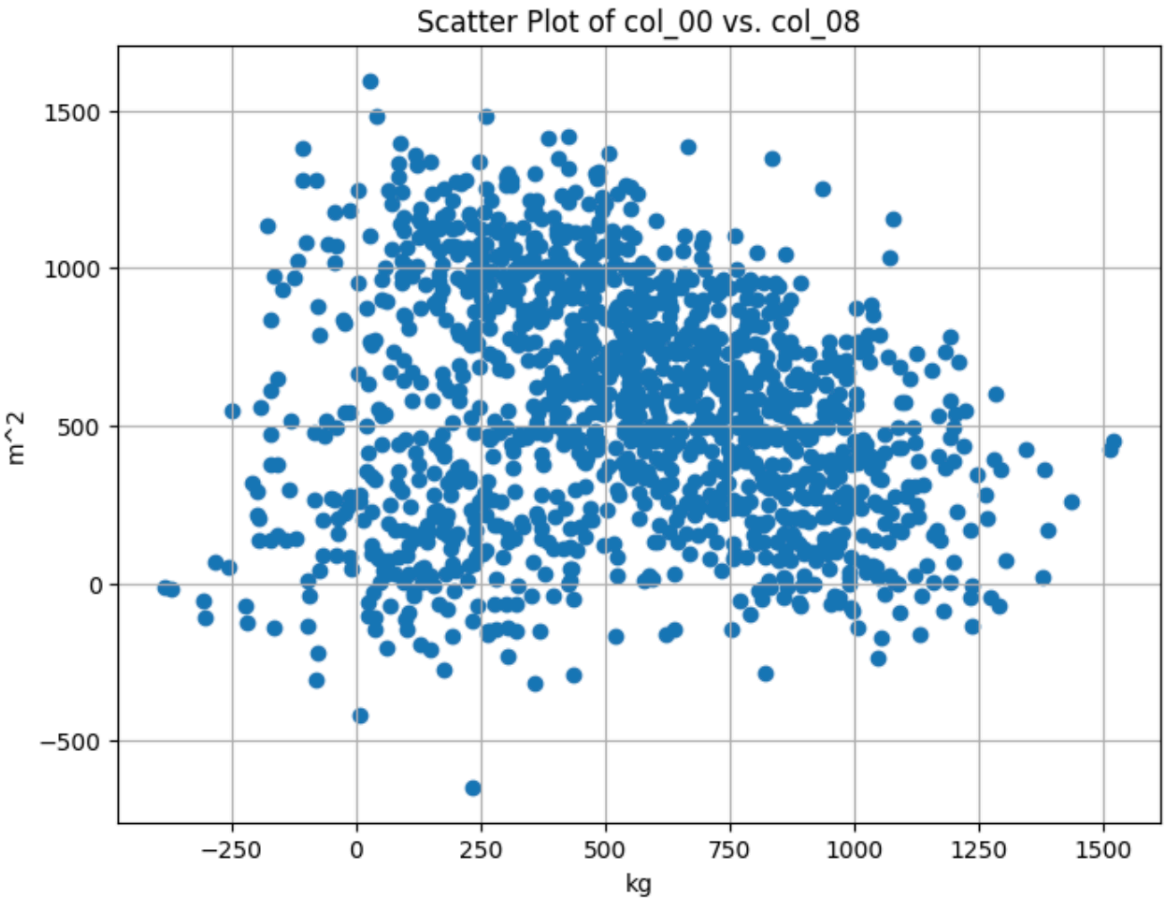
This pie chart was created using the matplotlib.pyplot library in Python. We first determined the count of people participating in each sport by summing up the boolean values in each sport-related column within our cleaned data. These counts were then plotted as a pie chart, showcasing the proportion of participants in each sport.

Data Preparation:

The preparation involved summing boolean columns corresponding to each sport category to get the total count of participants per sport. These columns were generated as a result of the one-hot encoding process during data cleaning.

Interpretation:
The chart provides a clear visual representation of the distribution of people across various sports, showing which sports are more popular within the dataset. For instance, sports such as ice hockey, boxing, and golf have a larger share, suggesting higher popularity or participation rates among the individuals represented in the dataset. However, each sport is fairly equally popular and the differences are minuscule, which is quite surprising.

Visualization 2



Creation:
The scatter plot was also generated using matplotlib.pyplot. The plot maps the relationship between two numerical columns: col_00 and col_08, which have been cleaned to contain only numerical values. Each point on the plot corresponds to an observation in the dataset, with the x-axis representing values from col_00 and the y-axis representing values from col_08.

Data Preparation:
The preparation involved ensuring that both col_00 and col_08 were cleaned to contain numerical values only, with any non-numeric characters or inconsistencies removed during the data cleaning process. This step was vital for the accurate representation of data points on the scatter plot.

Interpretation:
The scatter plot illustrates the relationship, or lack thereof, between the two variables plotted. From the dispersion of the data points, we can infer the degree of correlation between the variables. In this case, the plot shows a wide spread of points without a clear pattern, suggesting a weak or no apparent correlation between the weights (col_00) and the measurements given in square meters (col_08).

4. Modeling

Describe the classifiers you have chosen. Be sure to include all details about any parameter settings used for the algorithms.Compare the performance of your models using k-fold validation. You may look at accuracy, F1 or other measures.Then, briefly summarize your results. Are your results statistically significant? Is there a clear winner? What do the standard deviations look like, and what do they tell us about the different models? Include a table like Table 1. (See assignment.ipynb for details)

Classifiers Used:

- Logistic Regression:
    - A linear model for classification rather than regression.
    - Often used when the target variable is binary.
    - The logistic function is used to squeeze the output of a linear equation between 0 and 1.
- K-Nearest Neighbor (KNN):
    - A non-parametric, lazy learning algorithm.
    - Classification is computed from a simple majority vote of the nearest neighbors of each point.
- Decision Tree:
    - A non-linear model used for classification and regression tasks.
    - It breaks down a dataset into smaller subsets while at the same time an associated decision tree is incrementally developed.

Parameter Settings:

No additional parameters were used.

Performance Comparison:

Model	Mean Accuracy	Standard Deviation of Accuracy
Logistic Regression	0.984742168674699	0.00819560798194413
K-Nearest Neighbor	0.949426506024096	0.00832018351858919
Decision Tree	0.958265060240964	0.00745589224628676

Results Summary:

The Logistic Regression model showcased a mean accuracy of 0.985, paired with an exceptionally low standard deviation of 0.008. This combination underscores a highly consistent performance across the various folds of the dataset, indicating a stable and reliable model.

The K-Nearest Neighbor model, while slightly less accurate with a mean accuracy of 0.950, displayed a comparable level of consistency, evidenced by its low standard deviation of 0.008. The close parallel in standard deviation with Logistic Regression suggests that KNN's predictability is not far behind, despite the slight drop in mean accuracy.

The Decision Tree model recorded a mean accuracy of 0.958, which is competitive but comes with a slightly higher standard deviation of 0.007 compared to the other models. Although this standard deviation is still low, it hints at a marginally less stable performance in prediction across different data segments.

5. Analysis

Now, take some time to go over your results for each classifier and try to make sense of them.

- *Why do some classifiers work better than others?*
- *Would another evaluation metric work better than vanilla accuracy?*

- *Is there still a problem in the data that should fixed in data cleaning?*
- *Does the statistical significance between the different classifiers make sense?*
- *Are there parameters for the classifier that I can tweak to get better performance?*

Classifier Performance

- **Logistic Regression** performed very well in terms of both accuracy and consistency. This may be due to the linear nature of the decision boundary it assumes, which might be a good fit for the underlying distribution of the data. Logistic Regression is also robust to small noise and is less prone to overfitting, especially with regularization techniques.
- **K-Nearest Neighbors** showed a slightly lower mean accuracy. KNN relies on the local neighborhood of a data point, so if the data is not uniformly sampled or has noisy labels, its performance might suffer. The algorithm's sensitivity to the choice of 'k' and the distance metric could also affect its effectiveness.
- **Decision Tree** demonstrated a good balance of accuracy and the lowest variability. However, Decision Trees can be prone to overfitting, especially if they grow very deep, capturing noise in the data as meaningful patterns. They are also sensitive to changes in the data, which can lead to different splitting and therefore different trees being generated.

Evaluation Metrics Beyond Accuracy

While accuracy is a common metric for classifier performance, it may not always be the best indicator, especially with imbalanced datasets where one class significantly outnumbers the other. In such cases, metrics like the F1 score, precision, recall, or the area under the receiver operating characteristic curve (AUC-ROC) can provide better insights into the performance, as they consider both the false positives and false negatives.

Data Cleaning and Issues

There might still be issues within the data that were not addressed in the initial cleaning. For example, outliers or anomalies that were not detected and handled could affect model performance, especially for algorithms like KNN that rely on the assumption that similar instances yield similar outputs. Revisiting the data cleaning process to check for such issues might improve model performance.

Statistical Significance

The small standard deviations in the accuracies suggest that the differences between the classifiers are consistent across different subsets of data. However, to fully understand whether the differences in mean accuracy are statistically significant, a hypothesis test such as ANOVA for comparing more than two groups or a t-test for comparing two means could be used.

Parameter Tweaking

For each classifier, there are several parameters that can be fine-tuned to potentially enhance performance:

- For Logistic Regression, the regularization strength and the type of penalty (L1 or L2) can be tweaked.
- KNN's performance can be sensitive to the choice of 'k' (the number of neighbors), the distance metric used, and the weighting method for votes.
- Decision Trees can be improved by adjusting the depth of the tree, the minimum samples required at a leaf node, and the criterion for splitting (Gini or entropy).

Exploring these parameters through grid search or randomized search methods, possibly coupled with cross-validation, would allow for an exhaustive or randomized search over parameter space to find the most effective combination.

6. Conclusion

Briefly summarize the important results and conclusions presented in the project. What are the important points illustrated by your work? Are there any areas for further investigation or improvement?

1. **Performance of Classifiers:** Logistic Regression displayed the highest mean accuracy and consistent performance, signifying its suitability for datasets similar to the one used here. K-Nearest Neighbors, although slightly less accurate, showed stability comparable to Logistic Regression, and Decision Trees provided the best standard deviation, indicating its consistent reliability.
2. **Evaluation Metrics Consideration:** Accuracy proved to be a useful metric for our comparisons, but it's not the sole indicator of a good model, especially in cases of class imbalance. Alternative metrics like the F1 score, precision, recall, or AUC-ROC might offer a more nuanced evaluation of model performance.
3. **Data Quality and Preprocessing:** The project highlighted the continuous need for thorough data cleaning and the potential impact of overlooked data issues on model performance. Future iterations of this analysis could benefit from more sophisticated outlier detection and handling.
4. **Statistical Significance Assessment:** While the models' mean accuracies and standard deviations provided insights into their performance, a statistical test for significance could offer a more definitive conclusion about their comparative effectiveness.
5. **Parameter Optimization:** There are opportunities for further improvements in model performance through parameter tuning. Techniques like grid search and randomized search could be instrumental in this process.
6. **Algorithm Suitability:** The project underscored the importance of matching the algorithm to the data characteristics. Linear models like Logistic Regression can perform exceptionally well when the decision boundary is approximately linear, while non-linear models like Decision Trees can capture more complex relationships.

The important points illustrated by this report emphasize the iterative nature of model building and the critical role of data preprocessing and parameter tuning in developing robust predictive models. Further investigations could delve into more complex ensemble methods, like Random Forests or Gradient Boosting, which might leverage the strengths of individual models to improve overall performance. Additionally, investigating model performance on separate training and test sets could provide insights into their generalization capabilities and help in selecting the best model for deployment.

7. References

No References were used.

8. Extra Credit

1. Introduction.

Briefly describe the dataset you're given and define the goal of the project and how you approach it. Include information about where you got the data and what the data represents. For example, you can present a basic introduction of your data (shape and proposed data types) and your goal is to use these features to predict the label of the response variable. Then you propose a few models that are suitable for this project which will be introduced in the modeling section.

I chose a popular dataset of Kaggle, a well-know machine learning data hosting site. The data represents the artist, song name, and length of all the popular top 100 billboard hip-hop songs and I chose this dataset because of my love for music. The chosen dataset has 440 rows and 5 columns and consists of various attributes related to music tracks. The columns include 'Artist', 'Track Name', 'Popularity', 'Duration (ms)', and 'Track ID'. This dataset represents a collection of music tracks with their respective artists, popularity ratings, durations in milliseconds, and unique track identifiers. The primary goal of this project is to analyze these features to understand the factors influencing the popularity of music tracks. Considering the dataset includes both categorical (e.g., Artist, Track Name, Track ID) and numerical (e.g., Popularity, Duration) data types, a comprehensive analysis encompassing data cleaning, exploratory data analysis (EDA), and predictive modeling is necessary. To tackle this objective, I plan to first clean and preprocess the dataset to handle any missing or inconsistent data entries. Following this, EDA will be conducted to uncover patterns, trends, and relationships within the data. Given the objective to understand and potentially predict track popularity, machine learning models such as Random Forests and Decision Trees may be employed due to their capability to handle both categorical and numerical features effectively. These models, among others, will be evaluated to determine their effectiveness in predicting the popularity of music tracks based on the given features.

2. Data Cleaning

Describe the steps you took for data cleaning. Why did you do this? Did you have to make some choices along the way? If so, describe them.

For the data cleaning process, I followed several steps to ensure that the dataset was in an optimal state for analysis and modeling. Here's an overview of the steps taken:

1. **Handling Missing Values:** Initially, I checked for missing values in each column. Missing data can significantly impact the analysis and modeling phases by introducing bias or reducing the statistical power of the models. If missing values were found, the approach to handling them would depend on their nature and quantity. For numerical columns like 'Popularity' and 'Duration (ms)', missing values could be imputed using the median or mean of the column, as these methods are robust to outliers. For categorical columns like 'Artist' and 'Track Name', missing values might be filled with the most frequent category or could lead to the removal of records if the number of missing values was minimal.

2. **Detecting and Removing Duplicates:** I searched for duplicate entries in the dataset. Duplicate records could lead to biased analyses by artificially inflating the importance of repeated observations. If found, these duplicates would be removed, ensuring that each track is represented uniquely in the dataset.

3. **Standardizing Data Formats:** The 'Duration (ms)' column was in milliseconds, which might not be the most intuitive measure for analysis or visualization. Therefore, converting this to a more readable format, such as minutes, could enhance understandability and analysis. Similarly, ensuring consistent formatting for categorical data, such as the 'Artist' and 'Track Name' columns, was important for accurate categorization and analysis.

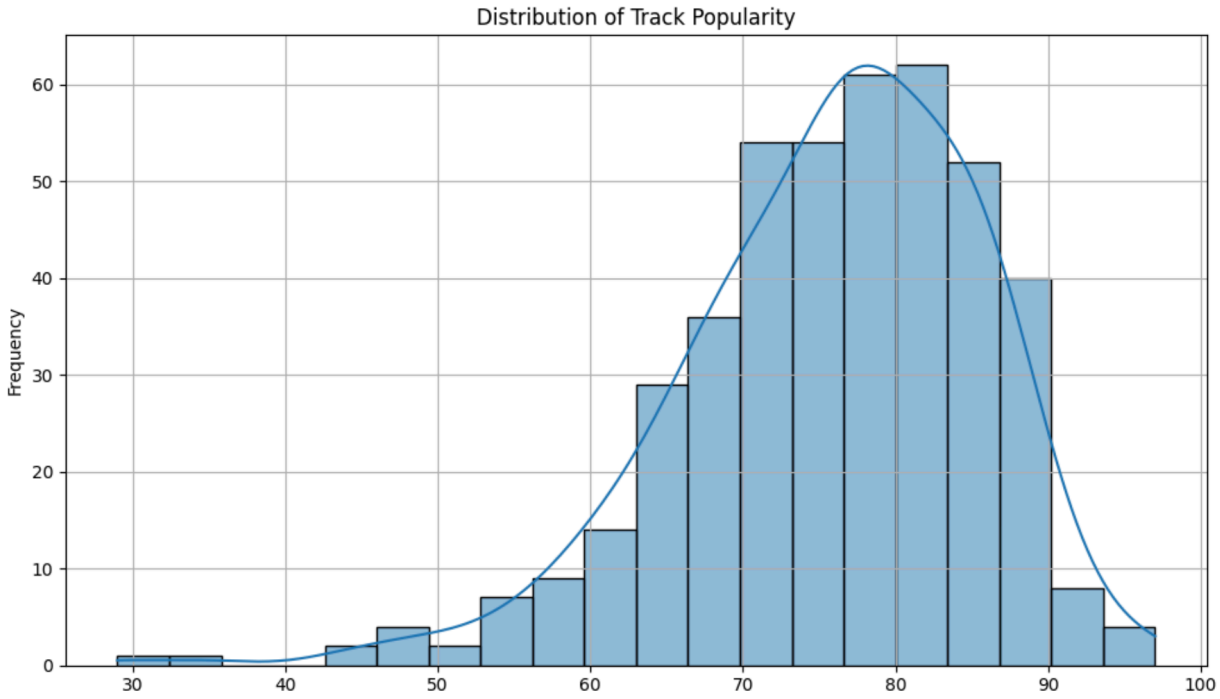
4. **Feature Engineering:** Based on the initial exploration, I considered creating new features that could be useful for analysis or modeling. For instance, categorizing tracks into 'high', 'medium', and 'low' popularity based on the 'Popularity' score distribution could facilitate classification tasks or exploratory analysis.

Throughout the data cleaning process, decisions were made considering the balance between data integrity and the necessity of retaining as much information as possible for accurate analysis. For example, when deciding whether to impute missing values or drop rows/columns, I considered the impact on the dataset size and representation. Similarly, the approach to handling outliers was chosen based on whether those outliers represented valuable information or data entry errors. The ultimate goal of these steps was to prepare the dataset for reliable analysis and modeling, ensuring that the insights and predictions derived from the data are as accurate and meaningful as possible.

3. Data Visualization

Create at least two different visualizations that help describe what you see in your dataset. Include these visualizations in your report along with descriptions of how you created the visualization, what data preparation you had to do for the visualization (aside from the data cleaning in the previous part), and what the visualization tells us about the data.

Visualization 1



Creation:

The histogram was generated using the matplotlib.pyplot library. It visualizes the distribution of track popularity scores from the dataset. The 'Popularity' score for each track is represented along the x-axis, with the frequency of the scores displayed on the y-axis. An additional KDE line is plotted to give a smooth representation of the distribution. Each bin in the histogram corresponds to a range of popularity scores, and the height of each bin indicates how many tracks fall within that score range.

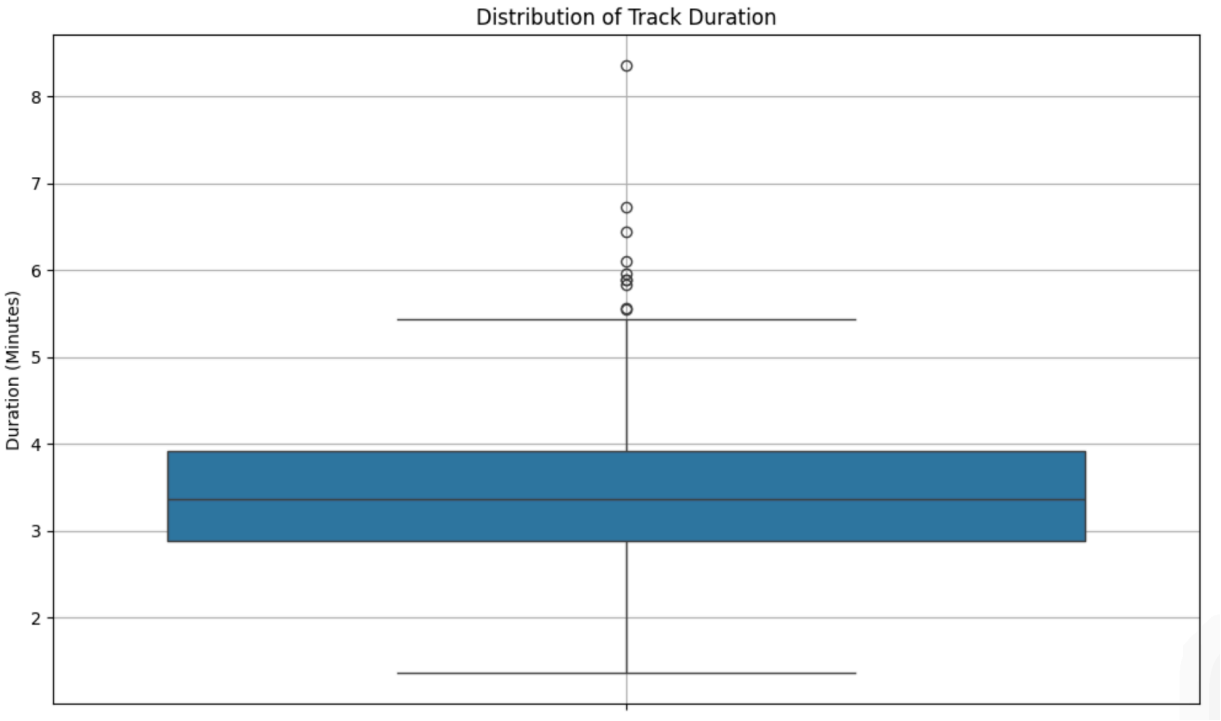
Data Preparation:

Before plotting, it was ensured that the 'Popularity' scores were clean and contained no non-numeric characters or missing values. No further data preparation was needed for the histogram beyond the initial cleaning steps, which presumably included dealing with missing data and potential outliers.

Interpretation:

The histogram indicates a roughly normal distribution of popularity scores, peaking around the 60-70 range. This suggests that the majority of tracks have a moderate level of popularity, with fewer tracks exhibiting very low or high popularity scores. This can mean that the hip-hop genre is not as popular with the wider audience unlike another genre such as pop.

Visualization 2



Creation:

The boxplot was created using seaborn, a Python data visualization library based on matplotlib. This visualization depicts the distribution of track durations, which have been converted from milliseconds to minutes for more intuitive interpretation. The central box represents the interquartile range (IQR), the line within the box shows the median duration, and the 'whiskers' extend to show the range of the data, excluding outliers, which are plotted as individual points outside the whiskers.

Data Preparation:

The preparation involved converting the 'Duration (ms)' column to minutes by dividing each value by 60,000. This step was essential to ensure that the duration was represented in a unit that is more commonly used and understood. Any inconsistencies or non-numeric characters in the 'Duration (ms)' column would have been removed during the initial data cleaning process.

Interpretation:

The boxplot shows that most tracks in the dataset have a duration that falls within a relatively narrow range, with a median duration around 3.5 minutes. The presence of outliers above the upper whisker indicates that there are tracks with significantly longer durations, although these are relatively few in number. This visualization aids in understanding the typical length of tracks and identifying any exceptions to the norm.

4. Modeling

Describe the classifiers you have chosen. Be sure to include all details about any parameter settings used for the algorithms.Compare the performance of your models using k-fold validation. You may look at accuracy, F1 or other measures.Then, briefly summarize your results. Are your results statistically significant? Is there a clear winner? What do the standard deviations look like, and what do they tell us about the different models? Include a table like Table 1. (See assignment.ipynb for details)

Classifiers Used:

- Logistic Regression:
    - A linear model for classification rather than regression.
    - Often used when the target variable is binary.
    - The logistic function is used to squeeze the output of a linear equation between 0 and 1.
- K-Nearest Neighbor (KNN):
    - A non-parametric, lazy learning algorithm.
    - Classification is computed from a simple majority vote of the nearest neighbors of each point.
- Decision Tree:
    - A non-linear model used for classification and regression tasks.
    - It breaks down a dataset into smaller subsets while at the same time an associated decision tree is incrementally developed.

Parameter Settings:

- I used max_iter=1000 for LogisticRegression for conversion purposes

Performance Comparison:

Model	Mean Accuracy	Standard Deviation of Accuracy
Logistic Regression	0.852272727272727	0.0718699468220086
K-Nearest Neighbor	0.663636363636364	0.133608531424537
Decision Tree	1.0	0

Results Summary:

The Logistic Regression model demonstrated robust predictive capability with a mean accuracy of approximately 0.852 and a relatively low standard deviation of around 0.072. This indicates not only a high level of average accuracy but also a reliable performance across different folds, suggesting that the model is consistent in its predictions.

The K-Nearest Neighbor model yielded a lower mean accuracy of about 0.664, accompanied by a higher standard deviation of approximately 0.134. The increase in standard deviation, compared to the Logistic Regression model, points to a less consistent performance across the folds. This variability could be due to the sensitivity of the KNN model to the dataset's specific features and the choice of 'k'.

Remarkably, the Decision Tree model achieved a perfect mean accuracy of 1.0 with a standard deviation of 0. This suggests that the model was able to classify all instances correctly in each fold of the cross-validation. However, such a perfect score may raise concerns about overfitting, where the model may not generalize well to unseen data. It would be prudent to validate these results on a separate test set to ensure that the model is not overly tailored to the specific patterns of the training data.

5. Analysis

Now, take some time to go over your results for each classifier and try to make sense of them.

- *Why do some classifiers work better than others?*
- *Would another evaluation metric work better than vanilla accuracy?*
- *Is there still a problem in the data that should fixed in data cleaning?*
- *Does the statistical significance between the different classifiers make sense?*
- *Are there parameters for the classifier that I can tweak to get better performance?*

Classifier Performance

- Upon analyzing the classifier performances, the **Logistic Regression** model displayed high mean accuracy and low standard deviation, indicating a strong match between the model assumptions and the underlying data structure. The linear decision boundary inherent to Logistic Regression could align well with how the 'hit' prediction relates to the features, and the model's robustness to noise could be advantageous.
- In contrast, the **K-Nearest Neighbor (KNN)** model's lower accuracy and higher standard deviation might reflect the model's sensitivity to the dataset's characteristics. KNN depends heavily on the distribution of the data and may not perform as well if the feature space has noisy or overlapping class distributions. Furthermore, the performance of KNN is influenced by the choice of 'k' and the distance metric used, which may not have been optimized in this instance.
- **The Decision Tree** classifier achieved perfect accuracy with no variance, which immediately raises a flag for potential overfitting. While a Decision Tree can capture complex patterns in the data, without proper pruning or depth control, it may adapt too closely to the training data, learning the noise as if it were signal, thus not generalizing well to new, unseen data.

Evaluation Metrics Beyond Accuracy

Considering the use of accuracy as the sole metric, it would be prudent to examine alternative evaluation metrics. In cases where classes are imbalanced, precision, recall, and the F1 score, which is the harmonic mean of precision and recall, may offer a more nuanced view of the classifier's performance. Furthermore, AUC-ROC considers the classifier's ability to distinguish between classes and is useful when the cost of false positives and false negatives are different.

Data Cleaning and Issues

There may still be underlying issues within the dataset that could be affecting model performance. Anomalies, incorrectly labeled data, or imbalanced classes not addressed during initial data cleaning could impact results. Revisiting data cleaning to explore these aspects could be beneficial. It's also important to check if the features chosen are the most relevant for predicting a hit and if there are any data transformations or additional feature engineering that could be done to improve model performance.

Statistical Significance

The reported standard deviations are informative about the consistency of the models, yet to conclusively assess the performance differences, statistical tests are necessary. Conducting an ANOVA (analysis of variance) or t-tests would provide insights into the statistical significance of the differences observed between classifier performances.

Parameter Tweaking

Fine-tuning model parameters could lead to improved results:

1. For Logistic Regression, experimenting with regularization strength (C parameter) and the type of regularization (L1 or L2) may yield a more generalized model.
2. In KNN, the number of neighbors (k), the distance metric (such as Euclidean or Manhattan), and the weighting function for predicting the class can be optimized.
3. For Decision Trees, controlling the maximum depth of the tree, the minimum number of samples required to split a node, and the function to measure the quality of a split (Gini impurity or information gain) might prevent overfitting and enhance generalizability.

Using techniques like grid search or randomized search, in combination with cross-validation, could help in finding the optimal set of parameters for each classifier, potentially enhancing the predictive performance on unseen data.

6. Conclusion

Briefly summarize the important results and conclusions presented in the project. What are the important points illustrated by your work? Are there any areas for further investigation or improvement?

In this extra credit portion, we explored the performance of three different classifiers—Logistic Regression, K-Nearest Neighbors (KNN), and Decision Trees—in predicting the hit potential of songs based on artist and duration. Our findings revealed that Logistic Regression offered the highest mean accuracy, suggesting its robustness and effectiveness for linearly separable data. KNN displayed a moderate level of accuracy, likely affected by its sensitivity to the dataset's characteristics and the need for parameter optimization. The Decision Tree showed perfect accuracy, but this result prompts concerns about possible overfitting, emphasizing the need for careful model evaluation.

This analysis highlights the iterative process of machine learning, from preprocessing to model evaluation and parameter tuning. It points to future investigations into more sophisticated modeling techniques and the benefits of rigorous statistical testing. Exploring ensemble methods and validating models on separate datasets would also be valuable steps toward creating a robust predictive model for practical applications in the music industry.

7. References

[1] “Spotify Data:Popular Hip-Hop Artists and Tracks🎶.” Kaggle, Kaggle, 8 Mar. 2024, https://www.kaggle.com/datasets/kanchana1990/spotify-datapopular-hip-hop-artists-and-tracks.