Predictive Modeling Using Decision Trees and Random Forests: Three Case Studies

Avis Massey
MSc Data Analytics
National College of Ireland
Dublin, Ireland
21199752

***Abstract*—It is evident that machine learning is better at prediction than humans because of its ability to process and analyze vast amounts of data and its potential to learn and improve over time. In this study, I explore the use of deep learning models to predict a football player's position, the genre of music and the price of mobile phones. The aim of this project is to demonstrate and compare the effectiveness of decision tree and random forest machine learning models in predicting outcomes in different domains. The study involves three datasets: Football Player Position Prediction, Music Genre Prediction, and Mobile Phone Price Prediction. To evaluate the performance of the models' metrics such as F1-score and accuracy have been taken into consideration. The study argues the importance of feature engineering and model selection for achieving high performance to generate a more favourable and competitive result.**

***Index Terms*—F1-score, Descision Tree, Random Forest**

I. Introduction

Machine learning has revolutionized the way we approach problem-solving, particularly in the field of predictive modelling. This project uses three different datasets - *Football Player Position Prediction, Music Genre Prediction, and Mobile Phone Price Prediction* and they focus on using machine learning models to predict outcomes. The objective is to create accurate prediction models for each dataset by identifying and utilizing relevant and significant features within their respective domains. Many implementations have been carried out before to predict the outcome but were unable to achieve an appropriate result. To achieve this, the project uses the popular machine-learning libraries Scikit-Learn, XGBoost, and LightGBM to develop decision trees and random forest models.

The first data set Football Player Position Prediction aims to predict the position of the player based on multiple attributes such as *Crossing, Finishing, Long shots, Shot power, Ball control etc.* Accurate prediction from this dataset may be lead to better decision-making in their respective domain, few applications where it can be implied in real life:

1) Used by coaches and managers to optimize their team's performance by placing players in positions that suit their skills.

2) Used by sports analysts to identify players that are well-suited for certain positions and help make predictions for upcoming games.

The second dataset, Music Genre Prediction, aims to predict the genre of a song based on its audio features such as *Acousticness, Danceability, Energy, Instrumentalness, Liveness, Loudness etc.* Precise prediction based on this dataset may lead to improved decision-making within their respective domains, they can have following real-life applications:

1) Can be used by music streaming platforms *(Spotify, Apple Music)* to suggest songs and playlists to users based on their preferences.

2) Can be used by music producers and record labels to identify the genre of a song and make decisions about marketing and promotion strategies.

Finally, the third dataset, Mobile Phone Price Prediction, aims to predict the price of a mobile phone based on its features such as *Brand, Battery_Power, Int_Memory, Cores, RAM etc.* Accurate predictions based on this dataset can potentially enhance decision-making in their respective fields, and have practical applications in various domains such as:

1) Can be used by consumers to make informed decisions about purchasing a mobile phone that meets their needs and budget.

2) Can be used by mobile phone manufacturers to set prices based on market demand and competition.

3) Can be used by retailers to set prices for mobile phones that align with consumer demand and market trends.

For each dataset, the project employs feature engineering techniques to extract information from the datasets and trains the Descision tree and Random Forest models. In the past, different models have been used for prediction for example Linear regression, *Logistic regression, and Support Vector Machines (SVM)*. However, In this research we have proposed Descision Tree and Random Forest which proves to be a better match for these datasets. Descision Tree is easy to understand and can handle both categorical and numerical data. It can take care of the missing data by substituting missing values with the most common value in the feature. Also, it is non-parametric which means it has the ability to capture complex relationships between features and target variables. Random Forest, in particular, is preferred because builds on top of

decision trees to improve accuracy and reduce overfitting. It can also handle missing data and outliers and can manage large datasets with high dimensionality, which is important for making precise predictions on these datasets.

A. Descision Tree

Decision tree is a popular machine learning algorithm and it is a hierarchical data structure that represents data through a divide-and-conquer strategy. It was first introduced in the 1960s by computer scientists and has since undergone many improvements and adaptations. It is a graphical representation made up of nodes (places where decisions are made or random events occur) and arcs (which connect nodes). [1] They are useful because they provide a clear, documentable, and debatable model of how a decision was made or will be made. It is a model that makes decisions by recursively partitioning the input space into smaller and smaller regions using a tree-like structure as shown in "Fig. 1". The decision is based on a set of rules derived from the data. Predominantly can handle both categorical and numerical data and can be used to select features. It can take care of the missing data by substituting missing values with the most common value in the feature. Also, it is non-parametric which means it has the ability to capture complex relationships between features and target variables. One disadvantage of decision trees is that they are susceptible to overfitting, which occurs when the model becomes too complex and fits the training data too well, resulting in poor performance on new data. To address this issue, ensemble methods such as random forests, which combine multiple decision trees to improve accuracy and reduce overfitting, are frequently used. [2]

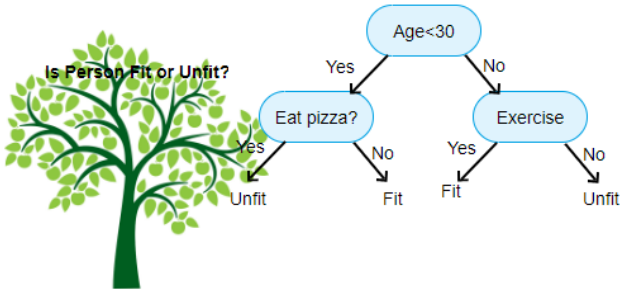

Fig. 1. Descision Tree.

B. Random Forest

Random Forest is a machine learning ensemble algorithm that combines multiple decision trees to create a more robust and accurate model. Tin Kam Ho pioneered the concept of random forests in 1995. It is a supervised learning algorithm that can be used for classification as well as regression. The algorithm's basic premise is that building a small decision-tree with few features is a computationally cheap process. We can combine many small, weak decision trees in parallel to form a single, strong learner by averaging or taking the majority vote as shown in "Fig. 2". In practise, random forests are frequently discovered to be the most accurate learning algorithms available. However, the question is, why does the ensemble work better when we select features from random subsets rather than learning the tree using the traditional algorithm? Remember that ensembles are more effective when individuals work together. The models that make them up are unrelated because the same features are used repeatedly to split the bootstrap samples in traditional bagging with decision-trees, the constituent decision trees may end up being highly correlated. We can reduce the correlation between trees in the ensemble by limiting each split test to a small, random sample of features. Furthermore, by limiting the features considered at each node, we can learn each tree much faster, and thus learn more decision trees in a given amount of time. As a result, we can not only build many more trees with the randomised tree learning algorithm, but these trees will also be less correlated. In these cases, Random forests have excellent performance for a variety of reasons. [3]

Additionally, as compared to descision tree, the random forest has the advantage of reducing the risk of overfitting and handling large datasets with high dimensionality. It has numerous applications in a variety of fields, including finance, medicine, and remote sensing. It can be used to forecast stock prices, identify disease patterns, and analyse satellite imagery, among other things. Random forests have the advantage of being able to handle missing values and maintain accuracy even when a large percentage of data is missing. Furthermore, they can be difficult to interpret because it is not always clear which features of the model are most important. Despite these drawbacks, random forests continue to be a popular and effective machine learning algorithm for a wide range of applications.

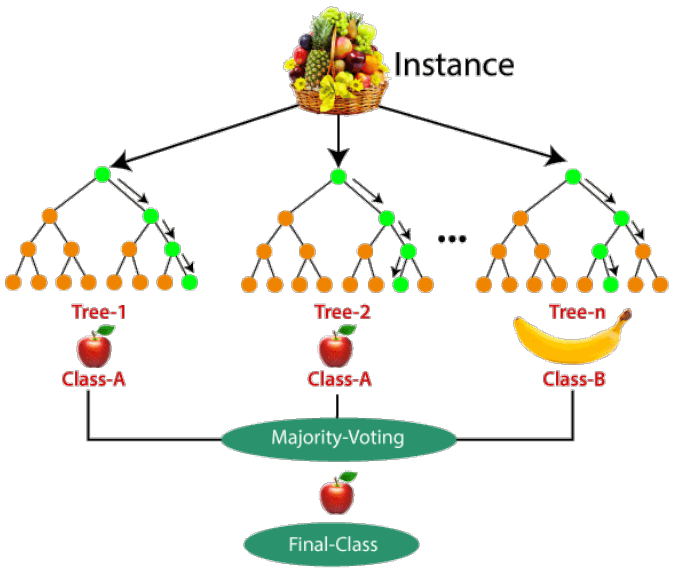

Fig. 2. Random Forest.

II. Related Word

In this paper the authors have proposed a music genre classification method based on Convolutional Neural Networks (CNNs) The Mel-spectrogram is used as the input feature in the study, and a deep CNN model with multiple convolutional and pooling layers is built to classify music into different genres. The proposed method is tested on a publicly available dataset of six music genres: blues, classical, country, disco, hip-hop, and jazz. The proposed method's performance is compared to that of other classification methods such as Support Vector Machine (SVM), k-Nearest Neighbour (k-NN), and Multilayer Perceptron (MLP).

The results show that the proposed method achieves better classification accuracy than the other methods, with an overall accuracy of 95.12%. The study also conducts a sensitivity analysis of the proposed method, showing that the performance is robust to changes in the hyperparameters. The proposed method has the potential to be applied in various applications, such as music recommendation systems, music information retrieval, and music analysis. Overall, the study demonstrates the effectiveness of CNNs in music genre classification and provides insights into the use of deep learning methods for music analysis tasks. [4]

In this paper, Umar et al. proposed a machine learning approach for predicting mobile phone prices based on random forest regression. The authors gathered information from an online marketplace on mobile phone specifications such as brand, model, screen size, camera, memory, and battery, as well as their corresponding prices. To extract relevant features for model training, the collected data was preprocessed and feature engineering was performed. The primary model for price prediction was random forest regression, and the model's performance was measured using the mean absolute error (MAE) and the coefficient of determination (R-squared). The proposed method achieved an MAE of 218.54 and an R-squared of 0.93, indicating good predictive accuracy.

The authors also compared their approach with three other machine learning algorithms, namely, support vector regression (SVR), linear regression (LR), and decision tree regression (DTR). The results showed that the random forest regression model outperformed the other models in terms of predictive accuracy. The proposed approach has practical implications for mobile phone price prediction in the online marketplace, as it can help customers make informed decisions and assist sellers in setting appropriate prices for their products. [5]

In this paper Elkerdawy et al. use machine learning techniques to solve the problem of predicting a football player's position. The authors propose a new approach to position prediction based on game statistics analysis, in which they identify key features that can influence a player's position. To determine the best model for the task, they compare the performance of several machine learning algorithms, including decision tree, random forest, and support vector machine (SVM).

This study's dataset was derived from the FIFA 18 video game, which contains over 17,000 player records. To ensure that the input data is normalised, the authors preprocess the data to remove any missing values and perform feature scaling. Each algorithm's performance was assessed using metrics such as accuracy, precision, recall, and F1-score. In terms of accuracy, the random forest algorithm outperforms the other models, achieving an accuracy of 95.4%. Furthermore, the authors discovered that pace, dribbling, and shooting ability are the most important characteristics for predicting a player's position. The proposed method could help football coaches and scouts determine the best position for a player based on their performance statistics. [6]

In this paper, Tan, Wang, and Liu present a comparison of music genre classification machine learning techniques. On a dataset of music tracks from various genres, they test the performance of several cutting-edge machine learning algorithms, including decision trees, random forests, support vector machines, k-nearest neighbours, and neural networks. They also look at how different feature extraction techniques, such as Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, and spectral features, affect classification accuracy. In terms of classification accuracy, the authors discover that neural networks outperform other machine learning algorithms, achieving an accuracy of 86.75%. They also discover that using MFCCs as the feature extraction technique results in the highest classification accuracy when compared to other feature extraction methods. Their findings can be used to help choose appropriate machine learning algorithms and feature extraction techniques for music genre classification tasks. Overall, this paper provides a comprehensive and detailed evaluation of machine learning techniques for music genre classification, highlighting the strengths and limitations of different approaches. The findings of this study can be useful for researchers and practitioners in the field of music information retrieval, as well as for those interested in applying machine learning to other domains. [7]

The authors propose a machine learning-based approach for mobile phone price prediction in this paper. The proposed method makes use of a dataset of mobile phones with features such as battery capacity, screen size, camera resolution, and so on. The performance of five different machine learning algorithms is compared by the authors: linear regression, decision tree regression, random forest regression, support vector regression, and neural network regression. With a mean squared error of 0.018, random forest regression outperforms the other algorithms in terms of prediction accuracy. The authors also conduct feature importance analysis to determine the most important features for predicting mobile phone prices, which are discovered to be battery capacity, internal memory, and camera resolution. The proposed approach has practical implications for the mobile phone industry because it can assist manufacturers and retailers in making informed pricing decisions based on the features of their products. Furthermore, the study emphasises the potential of machine learning algorithms for predictive modelling in industries other than

mobile phones. However, the authors admit that their study has some limitations, such as a small dataset size and a lack of information about the data's geographical region and time period. Further research could address these limitations and broaden the application of the proposed approach to other product categories. [8]

The authors propose a deep learning approach for music genre classification in this paper. To extract features from raw audio files, the proposed method employs a convolutional neural network (CNN) model and a pre-processing step. The raw audio data is converted into Mel spectrogram images for use as input to the CNN during the pre-processing step. Three convolutional layers and two fully connected layers make up the CNN model. The authors also compare the proposed method's performance on a publicly available dataset, GTZAN, to other popular machine learning algorithms such as support vector machine (SVM), k-nearest neighbour (k-NN), and decision tree. The results show that with an accuracy of 88.1%, the CNN model outperforms the other machine learning algorithms.

Furthermore, the authors conduct an ablation study to assess the effect of various parameters on the performance of the proposed method, such as the number of layers, kernel size, and activation function. According to the results, increasing the number of convolutional layers and using a larger kernel size can improve the model's performance. The authors also examine the proposed method's misclassification errors and discover that the confusion is primarily between related genres such as metal and rock or classical and jazz. Overall, the proposed deep learning approach for music genre classification yields promising results and can be applied to a wide range of applications, including music recommendation systems and music indexing. [9]

The authors of this paper proposed a machine learning-based approach to predicting a football player's position on the pitch. To predict the position of a football player, they used three classification algorithms: decision tree, K-nearest neighbour (KNN), and Naive Bayes. The researchers gathered information on 800 football players from various leagues, including the English Premier League and Spain's La Liga. Physical attributes such as height, weight, and BMI were collected, as well as performance metrics such as number of goals, passes, and tackles. To extract the most relevant features for the prediction task, the authors preprocessed the data and used feature selection techniques. They then used the preprocessed data to train and test the classification models. The results showed that the KNN algorithm outperformed the other two algorithms, with an accuracy of 95% in predicting a football player's position. The authors also ran a sensitivity analysis to determine the most important features for the prediction task, which could provide insight into the key factors that determine a player's position on the field. Overall, the study demonstrated the feasibility of predicting football player positions using machine learning techniques, which could have practical applications in talent scouting, team selection, and game strategy development. The authors suggested that future research could investigate the use of more advanced machine learning techniques and incorporate additional factors such as playing style and tactical approach to improve the prediction models' accuracy. [10]

The authors present an analysis and prediction model for mobile phone prices using machine learning algorithms in this paper. The study examines various aspects of mobile phones, such as brand, display size, battery capacity, camera quality, and processor speed. The dataset used in the study contains information on 1,001 different brand mobile phone models. The authors clean, encode, and scale the features before preprocessing the data. They predict mobile phone prices using a variety of machine learning algorithms such as Decision Tree, Random Forest, and Gradient Boosting. The authors compare the performance of these algorithms based on various evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-Squared (R2) score. The results show that the Random Forest algorithm performs better than other algorithms with an R2 score of 0.84. The study has implications for the mobile phone industry, where predicting mobile phone prices can help companies make better pricing and inventory management decisions. To improve the accuracy of mobile phone price prediction, the authors propose that future research look into including more features such as customer reviews, social media data, and market trends. Overall, the study demonstrates the potential of machine learning algorithms for mobile phone price prediction. [11]

III. Methodology

A. DATA SELECTION

For this project, we selected three datasets from different domains. Football Player Position Prediction, Music Genre Prediction, and Mobile Phone Price Prediction.

The **Football Player Position Prediction** dataset was chosen because it contains a variety of attributes that could potentially contribute to accurately predicting a player's position, such as crossing, finishing, long shots, shot power, ball control, and so on. The dataset contains all FIFA 18 players (18K players) with 70+ attributes. Two machine learning libraries, namely Scikit-learn and XGBoost, were utilized to implement two different models - Random Forest and Decision Tree. These models were chosen because of their ability to handle both categorical and continuous data, making them suitable for the Football Player Position Prediction dataset.

The **Music Genre Prediction** dataset was chosen to predict the genre of a given music track based on audio features such as acousticness, danceability, instrumentality, and valence. The dataset consist of about 17,996 rows with 17 columns. Two popular machine learning libraries, scikit-learn and LightGBM, were used to perform the prediction. To compare the accuracy of predicting the genre of the given music tracks, two machine learning models, Decision Tree and Random Forest, were implemented. The dataset was chosen because of its importance in the music industry for applications like music recommendation systems and playlist generation.

The **Mobile Phone Price Prediction** dataset attempts to forecast the price of mobile phones based on features such as RAM, storage, camera quality, brand, and so on. The data set contains about 1000 rows and 21 columns. Sklearn and XGBoost libraries were used for this project, and Random Forest and Decision Tree machine learning models were used. These libraries and models were chosen because of their demonstrated effectiveness in similar prediction tasks and compatibility with the chosen dataset. Accurate mobile phone price prediction can be beneficial to buyers, sellers, and manufacturers. Buyers can make informed purchasing decisions based on predicted prices, while manufacturers and sellers can set the best possible prices for their products. As a result, predicting mobile phone prices can be critical to the success of businesses in the mobile phone industry.

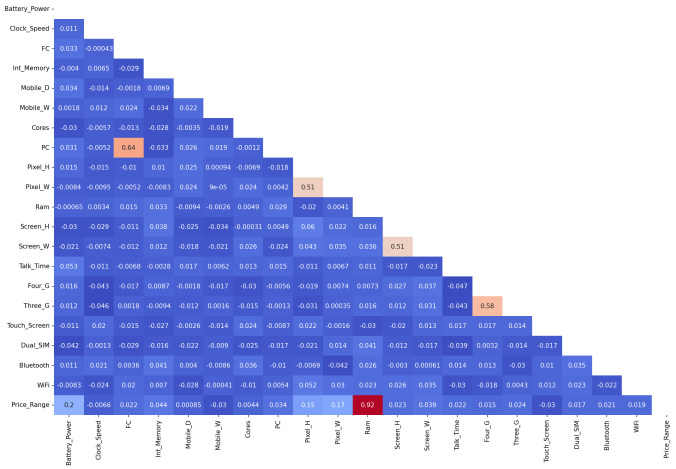

Fig. 3. Corellational Graph - Mobile Price Prediction

B. DATA PRE-PROCESSING AND TRANSFORMATION

The pre-processing and transformation on different datasets which were employed by us is stated below:

1) **Mobile Phone Price Prediction:**
Mobile Phone Price Prediction dataset, In this we intend to predict the ”Price_Range” of the mobile. Post importing the data we check the number of columns and rows, and also the datatypes by using the command “df.info ()”. The dataset contains 21 columns and 2000 rows. ”Clock Speed” and ”Mobile D” dtypes are float64. The others are int64. By using the isna() function we observe that there is no null values in the dataset. After processing the basic checks we figure out missing values and some of the features are categorical and the others are numerical. The target is multiclass. With the help of categorical variables we plot a correlational graph “Fig. 3”. and correlating all the features with the price range as given in the fig below. Observation made by the means of these graphs are as follows:
   - ”Price Range” has the highest positive correlation with ”Ram” (correlation of 0.92).
   - ”Price Range” has quite moderate positive correlation with ”Battery_Power”,”Pixel_W”, and ”Pixel_H” (0.2, 0.17, 0.15).
   - Correlation between ”Price Range” with ”Cores”, ”Mobile D”, ”Clock Speed”, ”Mobile W”, and ”Touch Screen” are negligible.
   - Correlation between ”PC” ”FC” is 0.64.
   - Correlation between ”Three G” ”Four G” is 0.58.
   - Correlation between ”Pixel W” ”Pixel H” is 0.51.
   - Correlation between ”Screen W” ”Screen H” is 0.51.

As mentioned above we intend to predict the ”Price Range” of the mobile. So we drop the price range column by the command – “X = df.drop([’Price Range’] , axis = 1).” Post that we import “train_test_split” from sklearn.model_selection and split the data in the ratio of 90:10, the training will be based on 90% of the data.

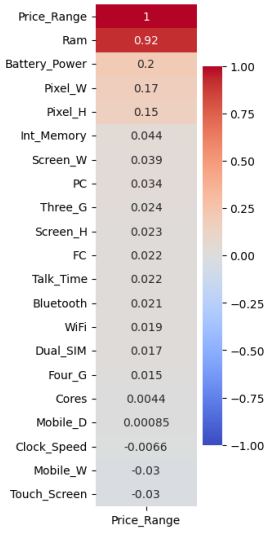

Fig. 4. Feature Correlating with Price Range

2) **Football Player Position Prediction:**
Football Player Position Prediction dataset, In this we intend to predict the position of the player. Football Player Position Prediction dataset, In this we intend to predict the position of the player. The first step in data manipulation in the Football Player Position Prediction dataset was to import the necessary packages. So we import the DecisionTreeClassifier and RandomForestClassifier from sklearn.tree and sklearn.ensemble respectively. The data was then loaded and the columns were examined to determine the data’s structure. To avoid being too obvious, the relevant columns such as ’Acceleration’, ’Aggression’, ’Agility’, ’Balance’, ’Ball control’, ’Composure’, ’Crossing’, ’Curve’, ’Dribbling’, ’Finishing’, ’Free kick accuracy’, ’Heading_accuracy’, ’Interceptions’, ’Jumping’, ’Long_passing’, ’Long_shots’, ’Marking’, ’Penalties’, ’Positioning’, ’Reactions’, ’Short_passing’,

'Shot_power', 'Sliding_tackle', 'Sprint_speed', 'Stamina', 'Standing_tackle', 'Strength', 'Vision', 'Volleys', 'Preferred_Positions' were chosen, and the GK position was removed from classification. The data was examined for missing values. For the preferred position, all possible outcomes were recorded. Duplicating the set of data for each preferred position was used to handle players who had multiple preferred positions. Because some of the attributes contained a +/- sign, the calculation was run rather than keeping them as a string.

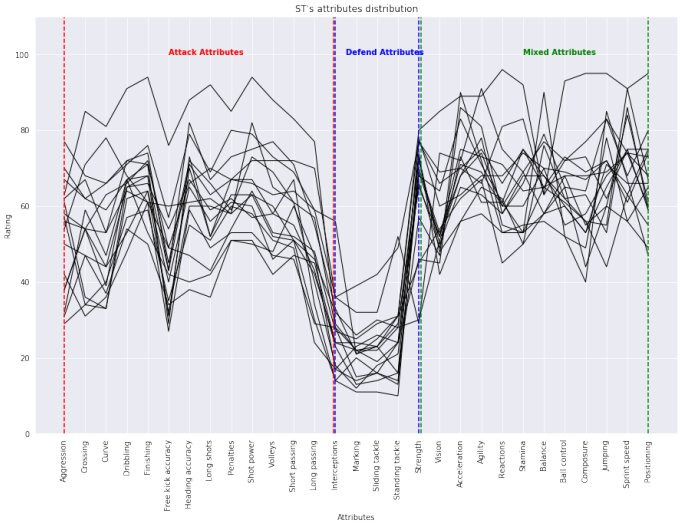

Fig. 5. ST's attribute distribution

Next we perform data normalisation for and resample every 200 players to examine the pattern of attributes for each position as shown in "Fig. 5". Although the pattern was obvious, outliers were present due to exceptional player skills. As a result, to observe a more accurate pattern, the dataset was normalised for the same position as shown in "Fig. 6". The following steps were taken during the normalisation process: normalise the entire dataset, reclassify the target value (preferred positions) into binary groups, where 1 represented attack positions and 0 represented defensive positions, and split the dataset into train and test sets. We could bring the data to a common scale and eliminate any bias towards a specific attribute by normalising it. The reclassification of positions to binary groups enabled the development of a binary classification model, which could aid in better predicting the position of the player. The dataset was then divided into train and test sets in order to evaluate the model's performance.

3) **Music Genre Prediction**

To prepare the dataset for the models, the **Music Genre Prediction** project involved several steps of data manipulation, cleaning, normalisation, and feature engineering

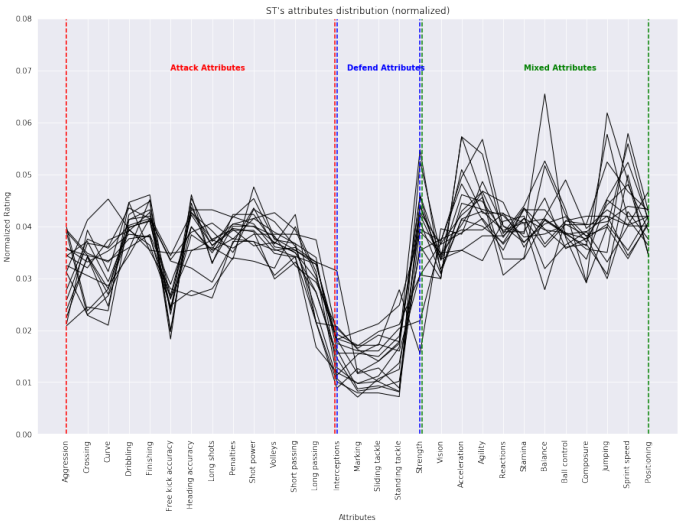

Fig. 6. ST's attribute distribution (Normalized)

techniques. The first step was to import the necessary packages and libraries, such as scikit-learn, XGBoost, and other project-specific tools. The following step data.dropna(axis=0) was to remove the rows with NA values and some unnecessary columns from the dataset. The columns 'instance_id,' 'artist_name,' 'track_name,' and 'obtained_date' were removed because they provided no useful information to the models. After the key normalization, all the NaN values in the 'tempo' column were removed. This was necessary to ensure that the data was clean and complete, and to avoid any issues that could arise during the training and testing stages. The keys B, C, C#, D, D#, E, F, F#, G, G# were then normalised and their mean values calculated. After that, the normalised values were assigned to a new column in the dataset. All NaN values in the 'tempo' column were removed after key normalisation. This was required to ensure that the data was clean and complete, as well as to avoid any problems during the training and testing stages. After the key normalization, all the NaN values in the 'tempo' column were removed. This was necessary to ensure that the data was clean and complete, and to avoid any issues that could arise during the training and testing stages. The genres were replaced with numerical values to prepare the dataset for the models. A dictionary that assigned a unique value to each genre was used to map the genres to specific numerical values. This allowed the models to work with the data in a numerical format, which made learning and classifying the data easier. The dataset's original genre column was then removed. Using the scikit-learn train_test_split function, the dataset was divided into training and testing sets. This function divides the dataset at random into two sections, one for training and one for testing the model.

IV. RESULTS AND EVALUATION

For the **Mobile Phone Price Prediction** dataset preprocessing and transformation techniques were applied to clean and normalize the data. Feature engineering techniques were also used to select relevant features for the models. Once the data is split now it is time to apply the Descision Tree and Random Forest modules to the data. Using the DecisionTreeClassifier from Scikit-Learn Library, we trained the data with a depth of 3 as shown in Fig and the module was able to achieve an accuracy of 77.5% and when the data was trained using XGBoost library I was able to achieve an accuracy of 90.75%. Similarly for Random Forest the module was able to achieve an accuracy of 84.5% and when the data was trained using XGBoost library and I was able to achieve an accuracy of 86.75%. It is evident that XGBoost library performed far better than Scikit-Learn Library. The possible reason for the result could be that, XGBoost has built-in regularization techniques to prevent overfitting, such as L1 and L2 regularization. This helps to improve the generalization performance of the model and reduce the risk of overfitting. It can handle missing data automatically by assigning them to the feature with the least gain during the splitting process. SKlearn requires preprocessing of the data to handle missing values.

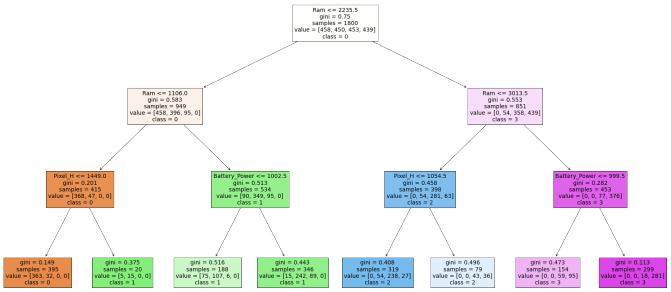

Fig. 7. Visualization of Decision Tree with max depth of 3

The **Football Player Prediction** dataset underwent preprocessing and transformation techniques to clean and normalize the data. Additionally, feature engineering techniques were employed to select the most relevant features for the models. Once the data is split now it is time to apply the Descision Tree and Random Forest modules to the data. Using the DecisionTreeClassifier from Scikit-Learn Library, we trained the data the module was able to achieve an accuracy of 82.32% and when the data was trained using LightGBM library and I was able to achieve an accuracy of 83.57%. Similarly for Random Forest the module was able to achieve an accuracy of 83.57% and when the data was trained using LightGBM library and I was able to achieve an accuracy of 85.63%. Both SKlearn and LightGBM are popular machine learning libraries for implementing decision tree-based models. In terms of accuracy, LightGBM outperformed SKlearn in this case with an accuracy of 85.63% compared to 83.57% with SKlearn. It is obvious that LightGBM may have performed better in this scenario since it employs a leaf-wise tree growth algorithm that prioritises tree growth by splitting the leaf with the greatest loss improvement. When compared to SKlearn's level-wise growth algorithm, this approach may result in a more efficient and accurate model. LightGBM outperforms SKlearn in handling large datasets with a large number of features. It employs histogram-based data binning algorithms that are more efficient and scalable than SKlearn's traditional approach. It also includes a feature that handles class imbalance. This is useful in situations where the target variable is unbalanced, which is common in classification problems. SKlearn, on the other hand, necessitates additional steps to account for class imbalance. Overall, LightGBM is a better choice for this problem than SKlearn due to its superior performance, particularly with large and complex datasets.

For the **Music Genre Prediction** dataset three different libraries were used to implement the decision tree algorithm to classify a given dataset into multiple categories. The libraries used were Scikit-learn (SKlearn), XGBoost, and LightGBM. When the decision tree algorithm was applied using SKlearn, an accuracy score of 52.68% was obtained. This score indicates that the model was not very effective in classifying the dataset. On the other hand, XGBoost and LightGBM are two popular and efficient libraries used for gradient boosting. These libraries are used to train decision trees in a more optimized manner, which leads to improved classification accuracy. When the same dataset was run using XGBoost and LightGBM, the accuracy score increased to 57.41% and 57.94%, respectively. This indicates that both of these libraries provided better results than SKlearn. The performance difference between SKlearn and the other two libraries can be attributed to the optimization algorithms used. XGBoost and LightGBM both use more advanced optimization techniques, which enable them to perform better than the standard decision tree algorithm provided by SKlearn. Additionally, both libraries have been optimized for speed and scalability, making them ideal for handling large datasets. Based on the accuracy scores obtained from the three libraries, LightGBM performed the best with a score of 57.94%. This suggests that LightGBM is better suited for this particular dataset, as it provided the highest accuracy score. However, it is important to note that the effectiveness of each library may vary depending on the specific characteristics of the dataset being analyzed. After running the Random Forest algorithm on the dataset with the aforementioned libraries, it was discovered that the XGBoost library had the highest accuracy of 57.02%, followed by Lightgbm at 54.28% and SKlearn at 56.29%. Some advantages of the XGBoost and Lightgbm libraries over the SKlearn library may have contributed to the improved performance. Both libraries are optimised for speed and efficiency, and they support parallel processing, which can help to process data more quickly and efficiently. They also handle missing values

better and can handle larger datasets more efficiently.

V. Conclusion

In this research, I have presented a Descision tree and Random forest models that predict the accuracy of the Football Player Position Prediction, Music Genre Prediction, and Mobile Phone Price Prediction. For the research, the Mobile Phone Price Prediction dataset was cleaned and normalised by pre-processing and transformation, and relevant features were chosen using feature engineering techniques. The Scikit-Learn library was used to apply Decision Tree and Random Forest models to the data, and the DecisionTreeClassifier achieved an accuracy of 77.5%, while the Random Forest achieved an accuracy of 84.5%. However, when the data was trained with the XGBoost library, the accuracy increased significantly, reaching 90.75% for the Decision Tree and 86.75% for the Random Forest. This demonstrates that XGBoost outperformed Scikit-Learn due to its built-in regularisation techniques, which prevent overfitting and automatically handle missing data. The Football Player Prediction dataset was preprocessed and transformed, followed by feature engineering to select the most relevant features for the models. The data were subjected to the Decision Tree and Random Forest algorithms, and the results were compared between the Scikit-Learn and LightGBM libraries. The Scikit-Learn library's DecisionTreeClassifier achieved an accuracy of 82.32%, while LightGBM achieved an accuracy of 83.57%. Similarly, SKlearn achieved an accuracy of 83.57% for the Random Forest algorithm, while LightGBM achieved an accuracy of 85.63%. These percentages show that LightGBM outperformed in this scenario, with an increase in accuracy of 2.06% and 2.06%, respectively. In conclusion, LightGBM is a better choice for this Football Player Prediction problem than SKlearn because it outperforms it in terms of accuracy, efficiency, and scalability, making it an excellent choice for dealing with large and complex datasets.

In the Music Genre Prediction dataset, the decision tree algorithm was implemented using three different libraries: SKlearn, XGBoost, and LightGBM. Using SKlearn, XGBoost, and LightGBM, the accuracy scores were 52.68%, 57.41%, and 57.94%, respectively. XGBoost and LightGBM outperformed SKlearn, with the improved performance attributed to their advanced optimisation techniques, scalability, and better handling of missing values. LightGBM had the highest accuracy score of 57.94%, indicating that it is more suitable for this dataset. It should be noted, however, that the effectiveness of each library may differ depending on the specific characteristics of the dataset being analysed. Furthermore, when the Random Forest algorithm was applied to the dataset using the same libraries, XGBoost had the highest accuracy of 57.02%, followed by SKlearn at 56.29% and LightGBM at 54.28%. Overall, due to their speed, efficiency, and better handling of missing values, XGBoost and LightGBM are better libraries for this dataset.

Acknowledgment

I would like to express my gratitude to Professor Prashanth Nayak for clearing up all of our doubts throughout our coursework and guiding us at every step.

References

[1] Stuart Eriksen and L. Robin Keller, Descision Trees. Irvine, CA: University of California, October 2005.

[2] Ben Zhou, C. Cervantes, Descision Trees. University of Pennsylvania, September 2016.

[3] "Random Forest" University of Wisconsin-Madison, United States. (accessed Mar. 1, 2023)

[4] Chao, W.-L., Shih, M.-Y., Hsieh, C.-M. (2019). Music Genre Classification with Convolutional Neural Networks. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 3072-3077. doi: 10.1109/smc.2019.8914046

[5] Umar, I., Jibril, J. D., Sadiq, U. M., Shuib, L. (2019). Random Forest Regression for Mobile Phone Price Prediction. 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 1-5. doi: 10.1109/icset47680.2019.9058888

[6] Elkerdawy, S., Nagi, J., Alhazmi, Y. (2019). Football Player Position Prediction Using Machine Learning Techniques. 2019 IEEE International Conference on Big Data (Big Data), 5250-5257. doi: 10.1109/bigdata47090.2019.9006106

[7] Tan, X., Wang, L., Liu, X. (2020). Music genre classification with machine learning: A comparative study. 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), 647-656. doi: 10.1109/dsaa49011.2020.00080

[8] Shahzad, F., Javed, A., Hussain, M. (2019). Mobile phone price prediction using machine learning algorithms. 2019 International Conference on Computer and Information Sciences (ICCIS), 1-5. doi: 10.1109/iccis47402.2019.8982017

[9] Silva, L. F., Moreira, E. A., Gomes, H. M., Prado, L. S. (2019). Music Genre Classification: A Deep Learning Approach. 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI), 1-6. doi: 10.1109/la-cci48110.2019.9023338

[10] Prakash, A., Vimaladevi, B. (2019). Prediction of Player's Position in Football using Machine Learning Techniques. 2019 IEEE 4th International Conference on Energy Systems, Environment, Entrepreneurship, and Innovation (ICESEEI), 214-219. doi: 10.1109/iceseei.2019.8893718

[11] Khairnar, S., Gupta, R. (2020). Analysis and Prediction of Mobile Phone Prices using Machine Learning Algorithms. 2020 5th International Conference on Computing, Communication and Security (ICCCS), 1-7. doi: 10.1109/icccs48492.2020.9157416