**Name:** Fares Islam Mansour

**Id:** 211001846

Data Analysis and Machine Learning on Match Result Prediction Report.

**Introduction:** Football is a passion for many, and many fans enjoy predicting match outcomes as to who will win or score. And when they predict the winner of the match or whether it is a draw by using evidence, like historical matches, and teams' performance in home or away games. Or the team’s stats like the clean sheets, red cards, goals conceded, wins, losses, and draws. There are many machine-learning models. These models can perform diverse tasks. This report aims to compare different models to predict the winner of a match between 2 teams one home and one away. This will be done by using different models such as (Logistic Regression, Random Forest, Gradient Boosting, Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, Adaboost, Xgboost, and Decision Tree). By evaluating them, the best model will be used to predict the match result.

**Data Background:** The data was acquired from the Premier League website and is representative of seasons 2006/2007 to 2017/2018.

**First file = results.csv**

**About this file:** Results of 4560 Premier League matches - 380 matches over 12 seasons from 2006/2007 to 2017/2018

**Second file = stats.csv**

**About this file:** Statistics collected from Opta (official stats collector of the Premier League) of each team in every season (season totals) from 2006/2007 (i.e. the season from which the collection of these detailed stats began) to 2017/2018.

**Steps:** Developing the model was done by performing 5 major steps, that included many steps in each one.

**1- Data Exploration and Preprocessing:**

This step is taken to fully comprehend the data. and to clean it up in case the data has any problems. Several steps are taken in the preprocessing stage to ensure that the model works with the best possible data. This covers computing z-scores, addressing missing values, and handling outliers. Then, to thoroughly examine the data and obtain an understanding of its properties and connections, exploratory data analysis, or EDA, is carried out. Since these are the most effective EDA kinds, univariate, bivariate, and multivariate studies were conducted. Using the right plots to visualize the data is the final step in the data exploration process.

**2- Feature Engineering:**

Categorical variables were present in this data, which will have an impact on the accuracy of the model and which certain models are unable to handle. Consequently, by converting the numerical variables from the category variables. As a result, the dataset's predictive power increased. One-hot encoding was sufficient for the data; further forms of feature engineering were not required. One-hot encoding's function was completed. However, the codes do not use the one-hot encoding line. As a result, I changed the season, home team, away team, and result categories to replace the category variables with the desired numbers. For example, the ‘result’ column had the variables (H, A, D) where replaced (1, 2, 0) respectively.

**3- Machine Learning Model Development:**

The classification problem is the most appropriate machine learning challenge, given the nature of the dataset and the task itself. following scaling and imputing and following the division of the data into training and testing sets. These datasets were employed to create several machine-learning models. The most accurate model was found by comparing and evaluating nine different models. Logistic Regression, Random Forest, Gradient Boosting, Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, Adaboost, Xgboost, and Decision Tree were the models that were employed in the comparison. KNN, Random Forest, and Support Vector Machine (SVM) were the most accurate models after all the models were trained and evaluated, and their assessment metrics were satisfactory. Accuracy, precision, F1 score, confusion matrix, recall, and the mean of the cross-validation score were the evaluation metrics employed. Upon comparing every model, some demonstrated good accuracy while others did not.

**4- Model Evaluation and Fine-tuning:**

Use the following hyperparameters after testing the models using the metrics discussed: n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf. The process of determining a model's optimal hyperparameters through the application of the grid search function is known as "model fine-tuning." In machine learning, fine-tuning settings is crucial. It can have a significant impact on a model's functionality and adaptability. However, optimizing these settings requires thorough testing and comprehension of the issue you're attempting to resolve.

**5- Model Deployment:**

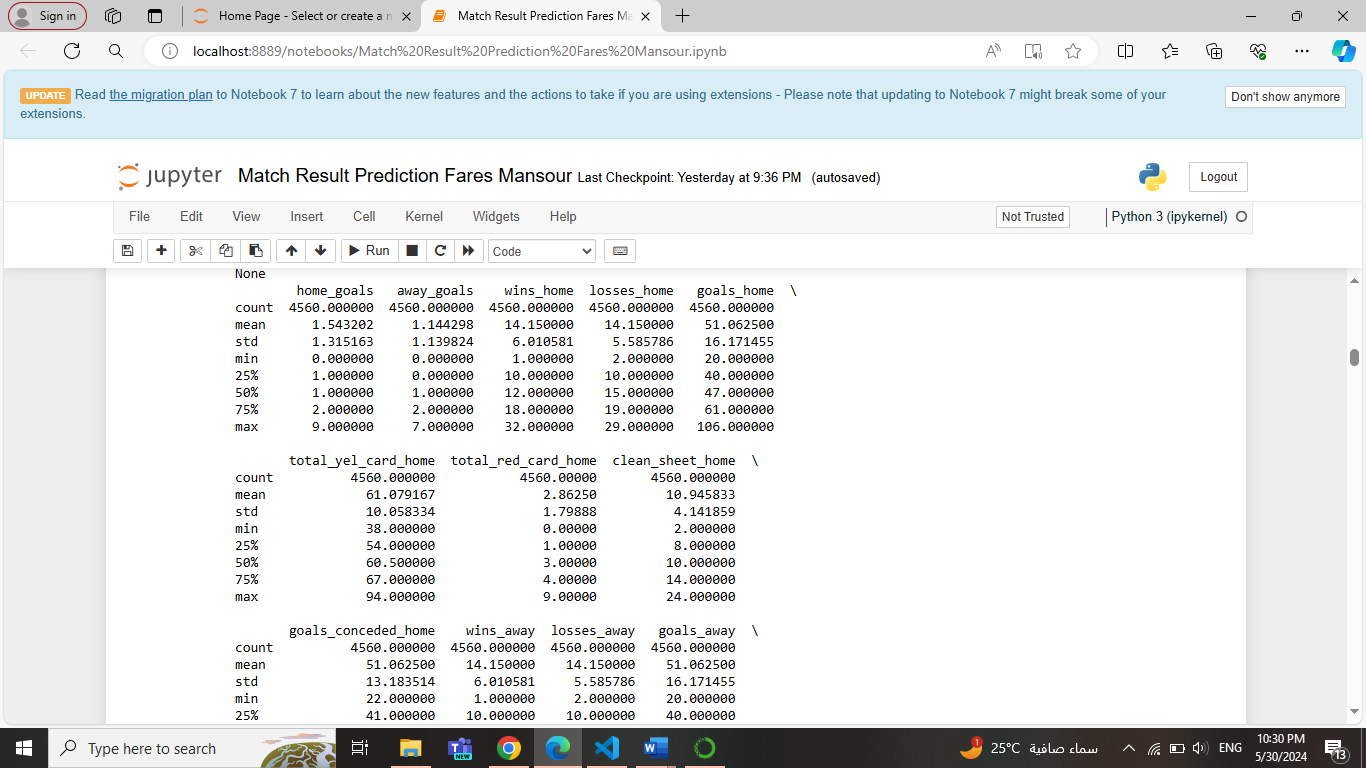
Four models functioned successfully once all the model development stages were completed, and the outputs were interpreted. To construct the deployment model, one of them was taken. When Random Forest was applied, accurate predictions were produced. Utilizing the features and utilities of the Streamlit web app, the model was deployed. It was successful to test the model on the local host that Streamlit supplied. The algorithm predicts which side will win when the user enters various team statistics, such as goals, wins, losses, yellow/red cards, clean sheets, and many more features. Building the deployment model was made simpler by the fact that all of the code needed to do so had already been developed. The notebook was simply added to the programs that created the Streamlit interface, and the deployment model was completed.

**Results:**

A screenshot of a computer

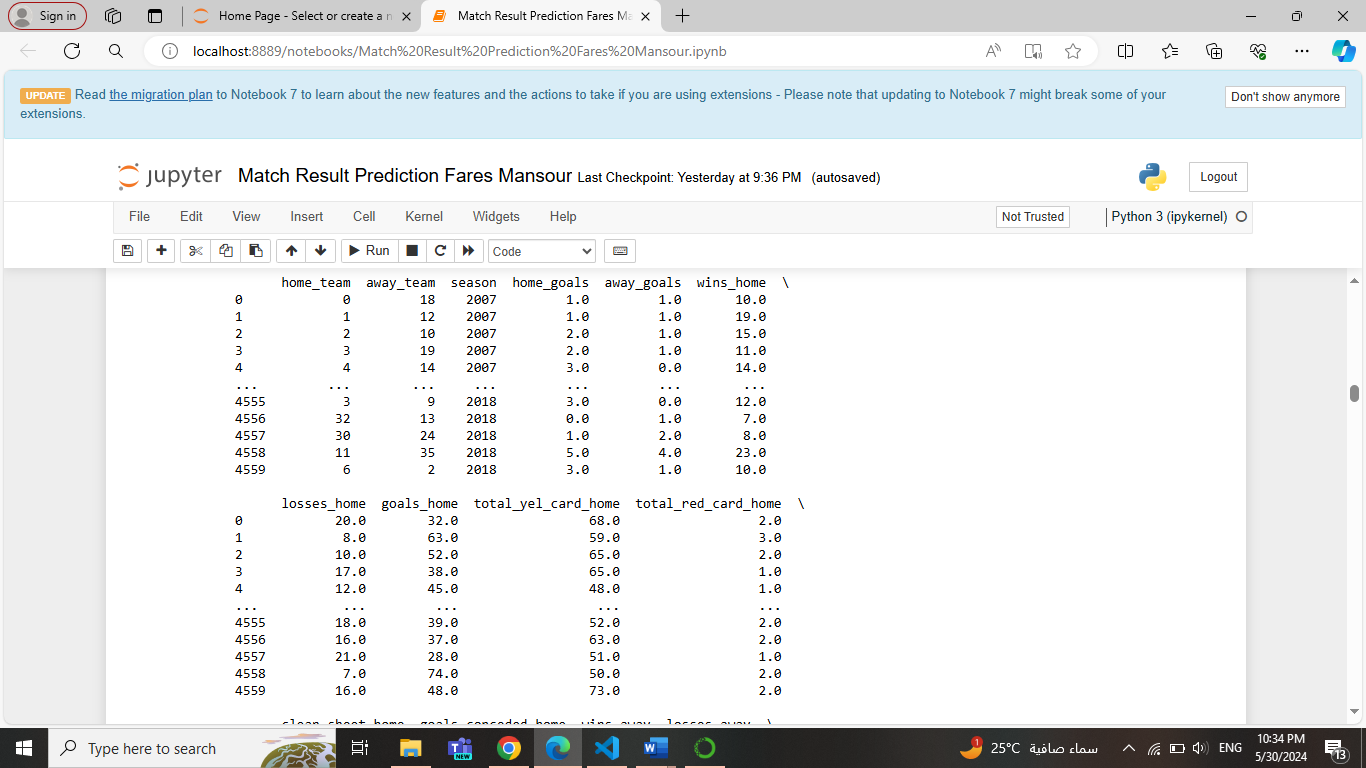
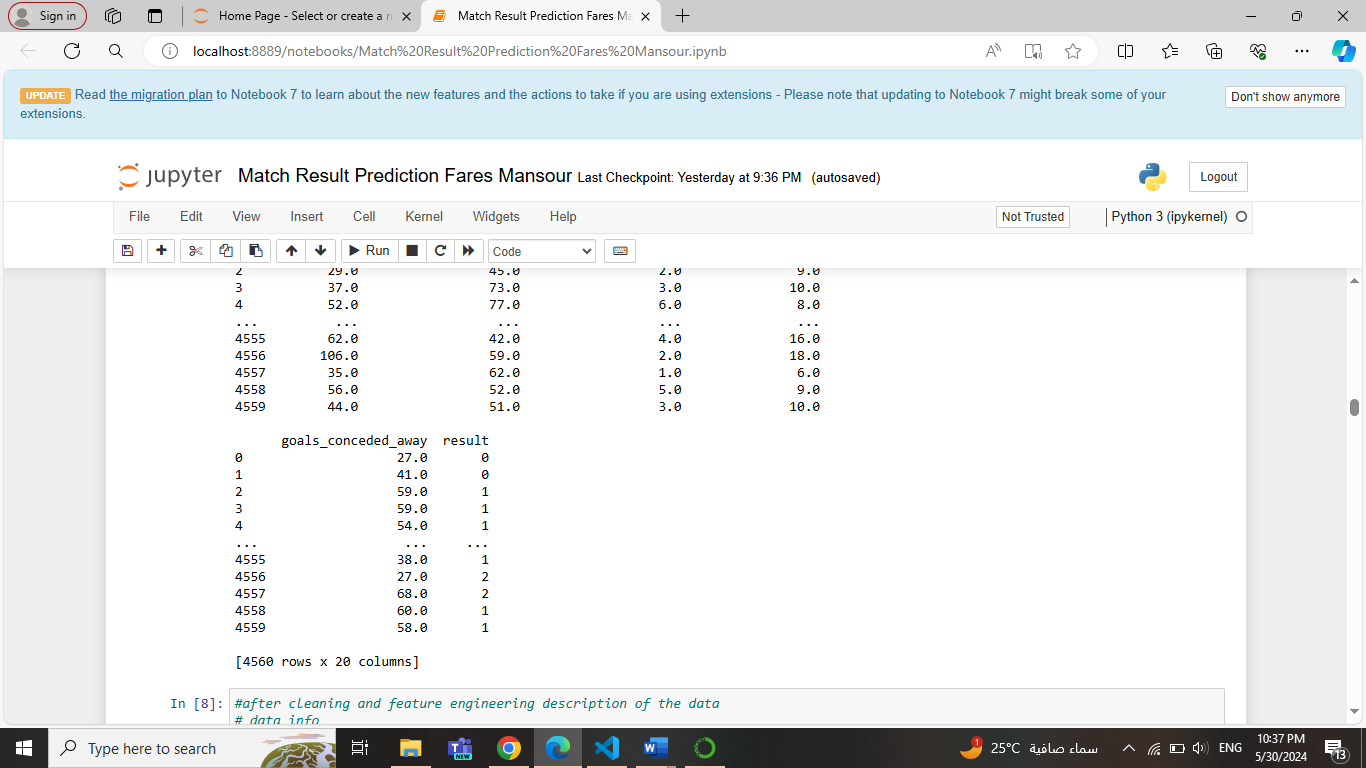
Description automatically generated**1- Data Summary:**

This is the first output which summarizes the data. There are 4560 entries, a total of 19 columns with their indexes. No null values are present in the rows. Also, the data type of every column is defined. Lastly, the memory usage is defined.

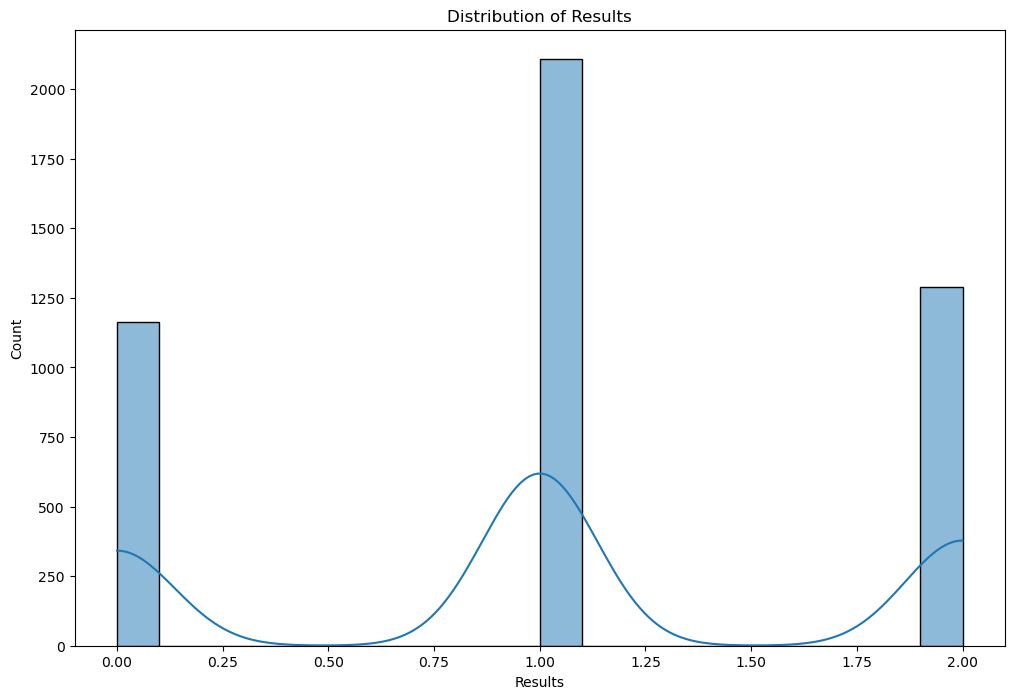
**2- Descriptive Statistics:**

This is a sample of the output that displays the descriptive statistics of the dataset used. As the count, mean, standard deviation, the IQR ranges, and the minimum and maximum numbers.

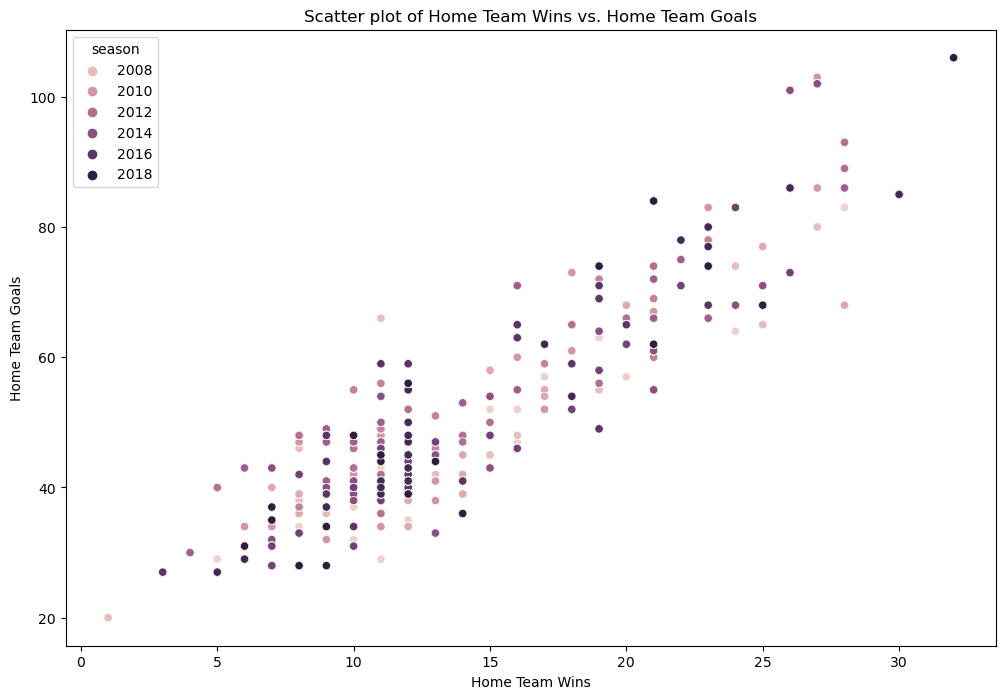
* **Data After Feature Engineering:**

****

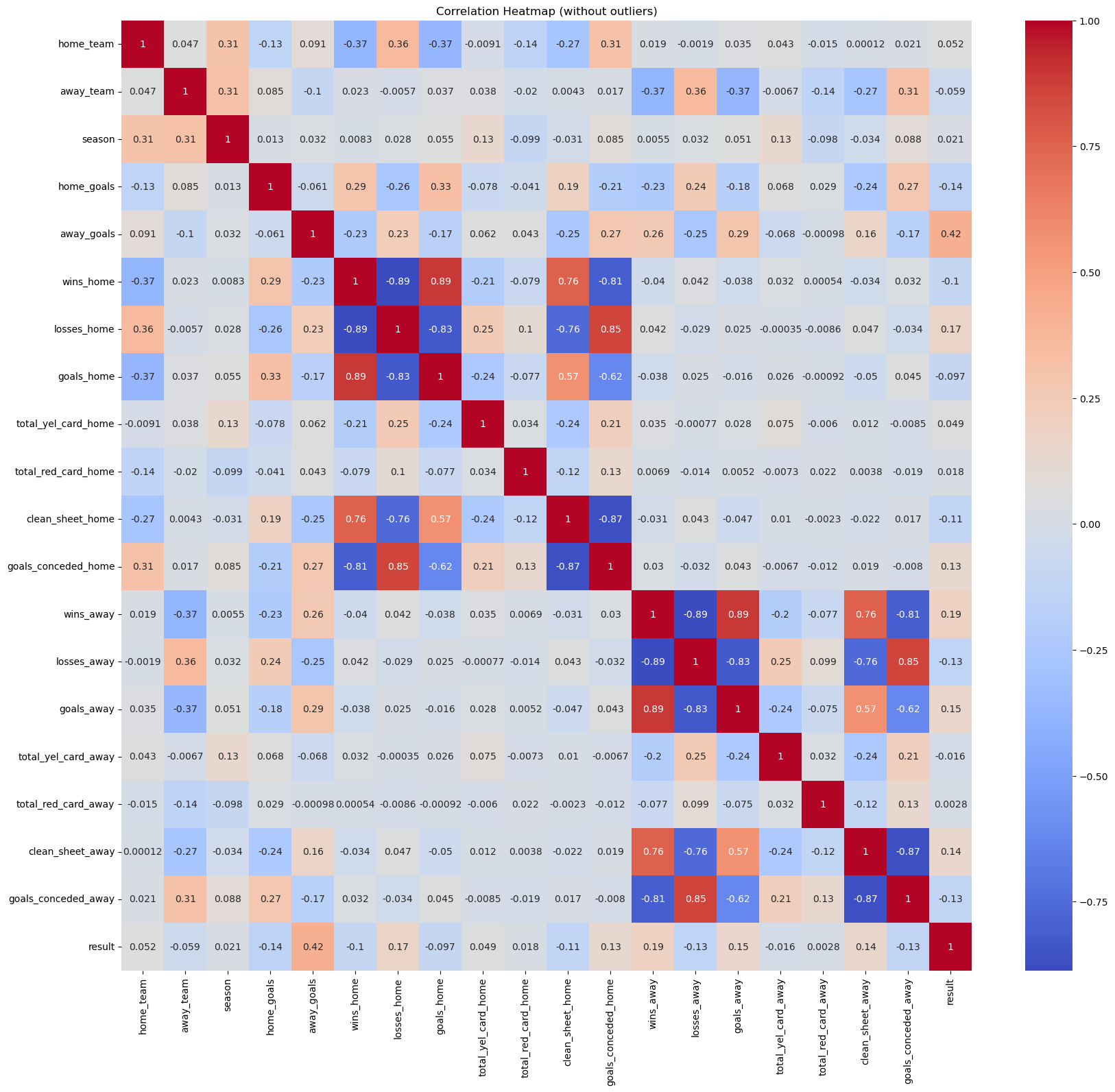
This shows the data after transforming all the categorical variables to numerical, and regarding the home/away team columns, the team names were sorted, and every team had a unique number to be assigned.

**3- Univariate Analysis:**

This is the first output plot from the EDA analysis, which is the univariate analysis. It shows the distribution of the result column in the dataset. This plot reveals that home wins results are the most frequent in the premier league, followed by away wins and the least are draws.

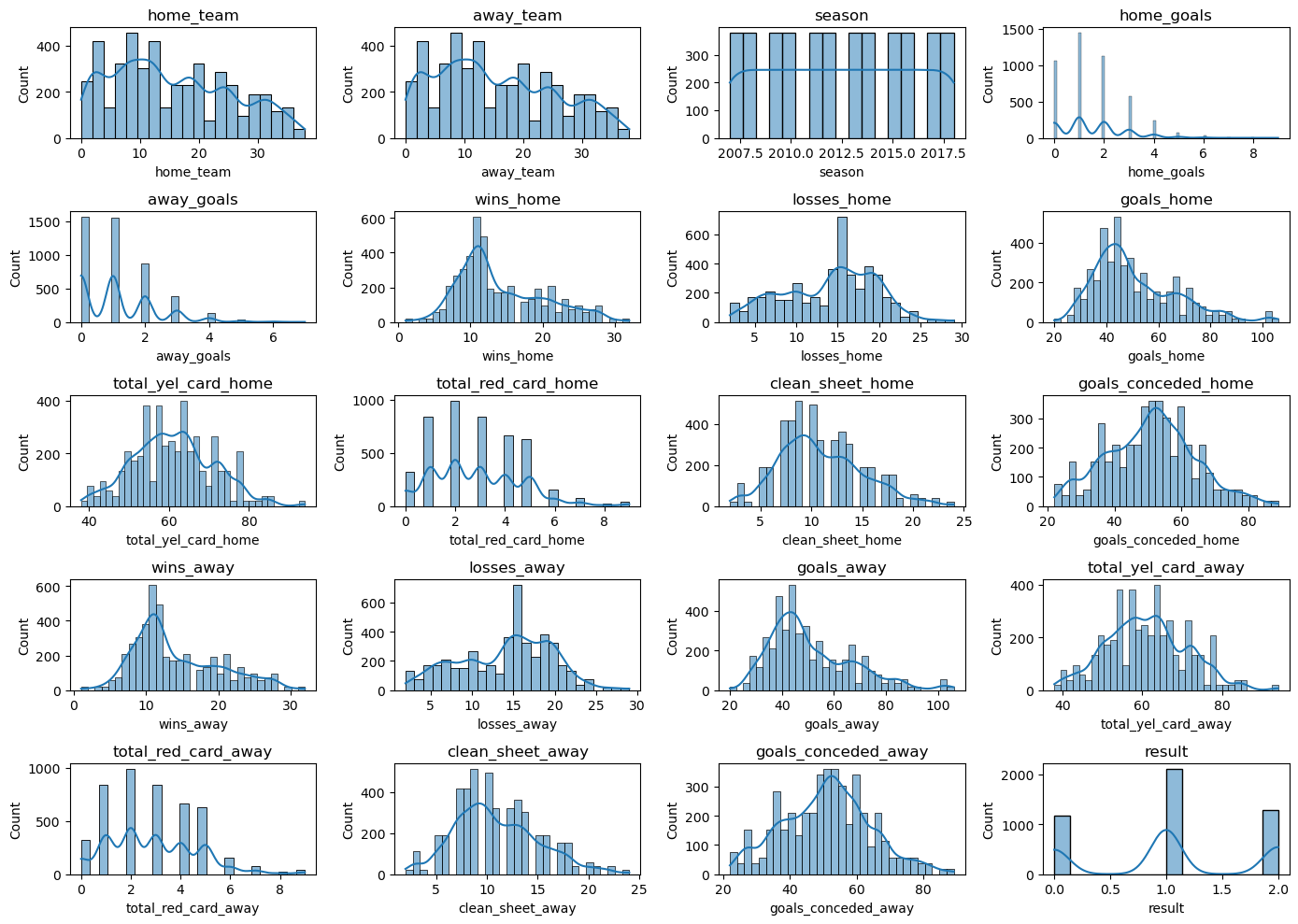
**4- Bivariate Analysis:**

This is the second output plot from the EDA analysis, the bivariate analysis. It shows a scatter plot that contains the 2 columns Home Team Wins and Home Team Goals. They are plotted against each other, and the ball’s color is based on season.

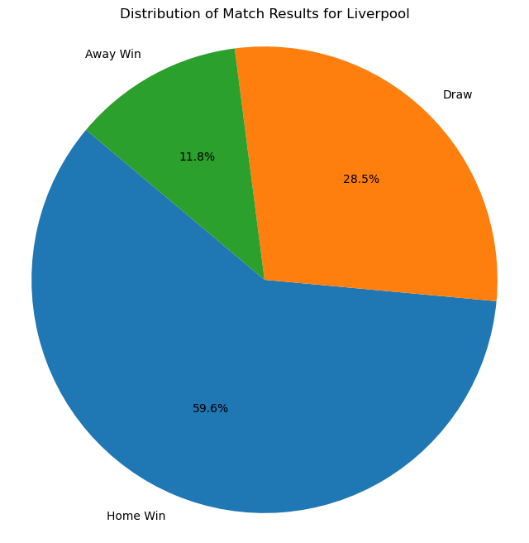
**5- Multivariate Analysis:**

This is the last output plot from the EDA analysis, which is the multivariate analysis. It shows a heatmap plot, that plots all the data’s columns against each other. If the correlation between the variables increases, the box's color becomes darker, the light boxes indicate low correlation.

**6- Histograms for All the numerical features:**

****

This histogram shows the distributions of all the numerical features of the data. Taking the home/away team's histograms, as they show the same numbers and the same plot, shows how accurate and unique the indexes given to them were.

**7- Pie Chart of a random team (Liverpool):**

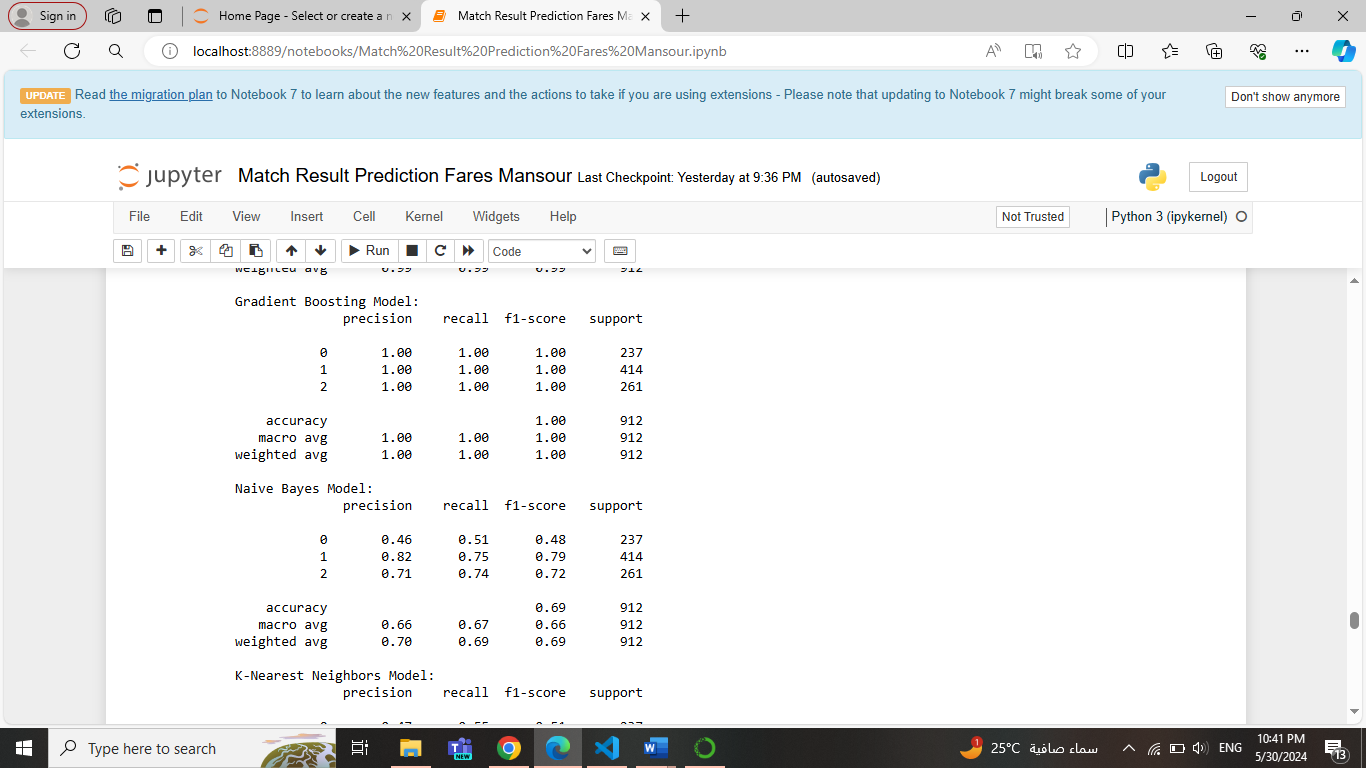
This pie chart shows the results for the Liverpool team only, where their home wins are the biggest portion of their results. Followed by draws. And the least were away wins.

**8- Classification Reports for the models:**

**A screenshot of a computer

Description automatically generated**

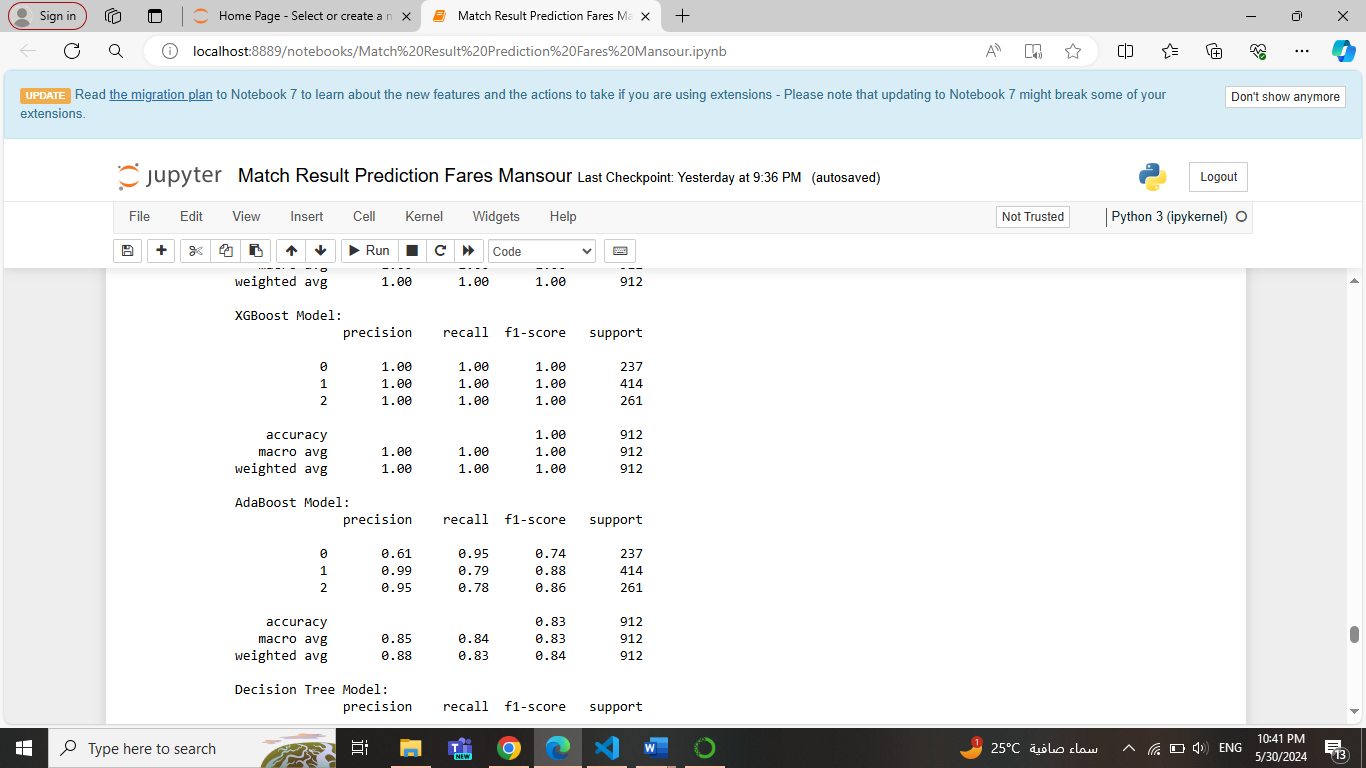
These are the outputs for the classification report for Logistic Regression and Random Forest models. The Precision, recall, and f1-score are better in Random Forest than in Logistic Regression as it gave in all 1 which is not true. Precision means that out of all the instances, your model predicts as positive how many are correct. Recall is about capturing all the positive instances. It measures how many of the actual positive instances your model can identify. And the F1 score is a way to balance precision and recall. It's the harmonic means of precision and recall. It gives you a single number that represents both precision and recall.

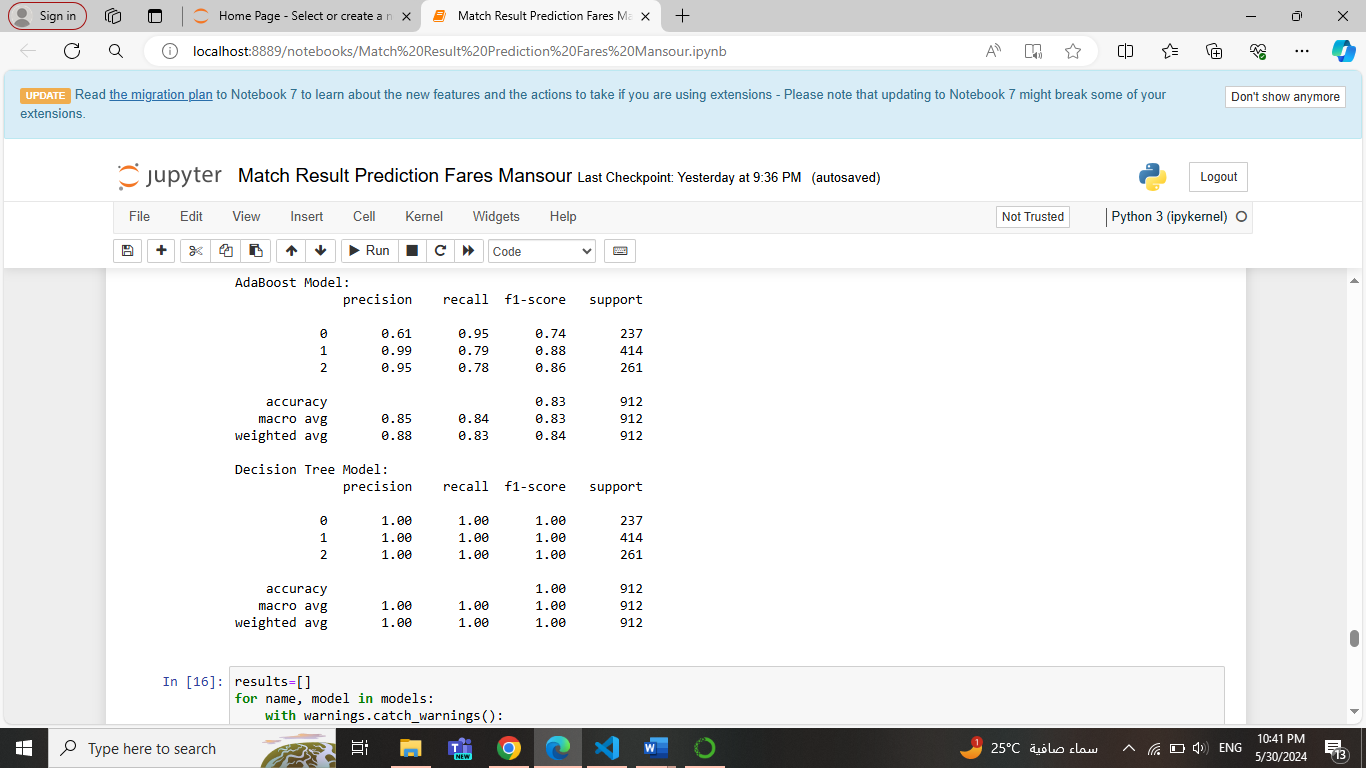


These are the outputs for the classification report for Gradient Boosting and Naïve Bayes models. The Precision, recall, and f1-score are lower in Naïve Bayes than in Gradient Boosting.

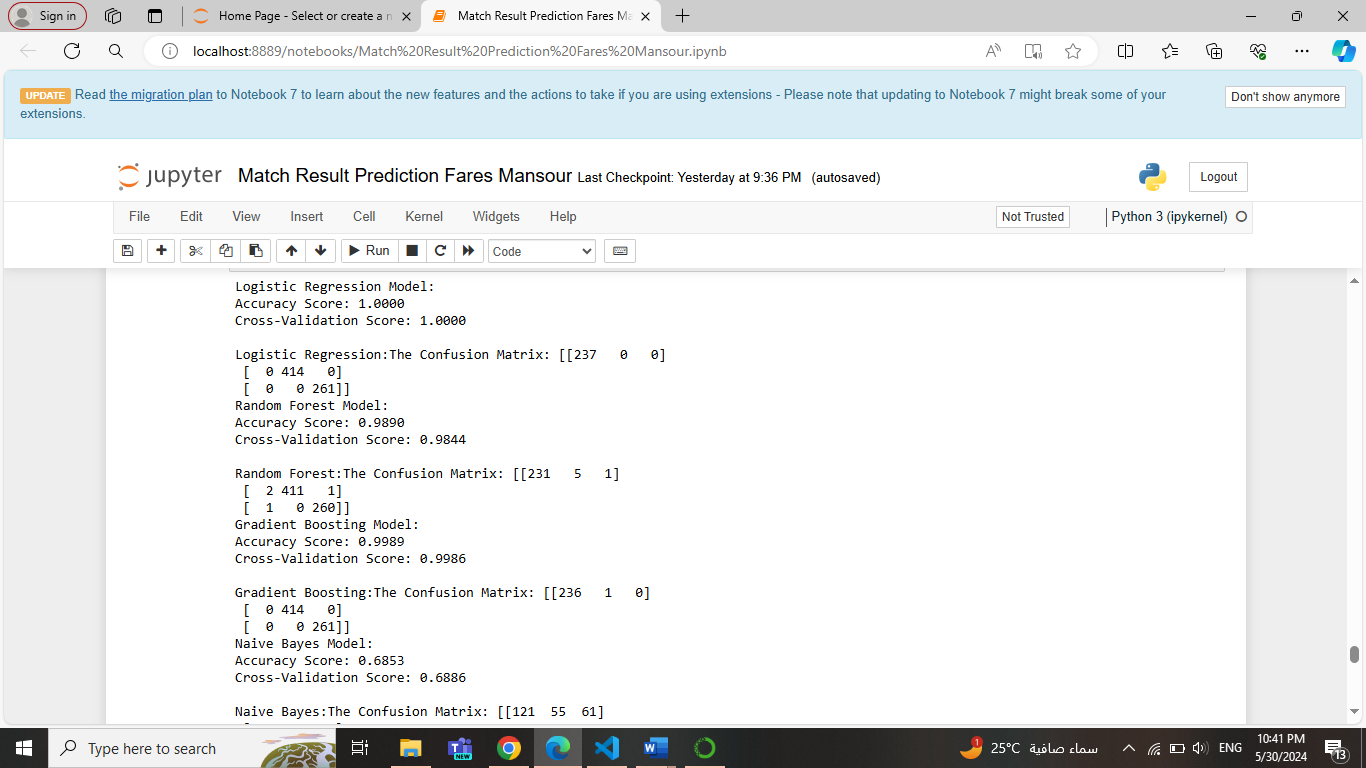
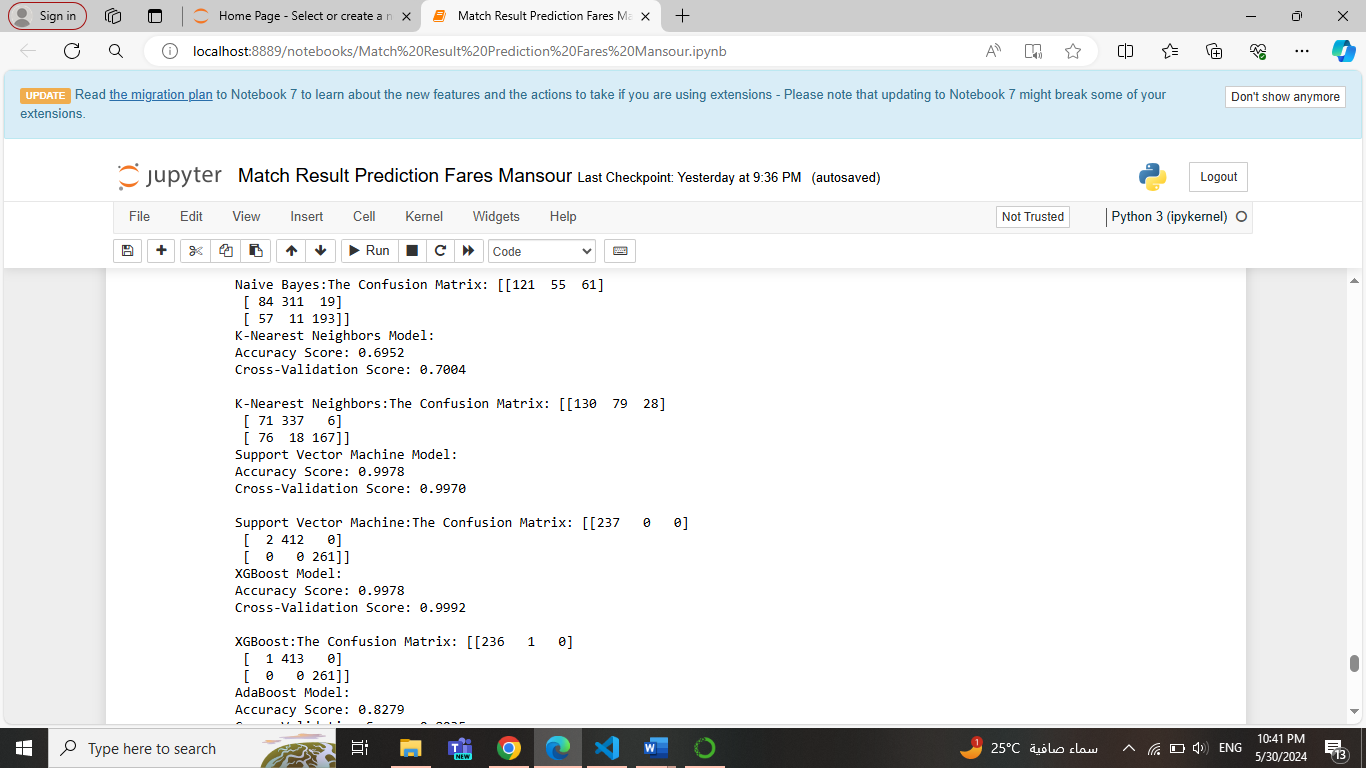
A screenshot of a computer

Description automatically generated

These are the outputs for the classification report for K-Nearest Neighbors and Support Vector Machine models. The Precision, recall, and f1-score are better in the Support Vector Machine than in K-Nearest Neighbors.



The best one among the last 3 models (XGBoost, AdaBoost, Decision Tree), is the Decision Tree model, followed by XGBoost.

**9- Models Evaluation Metrices:**

A screenshot of a computer

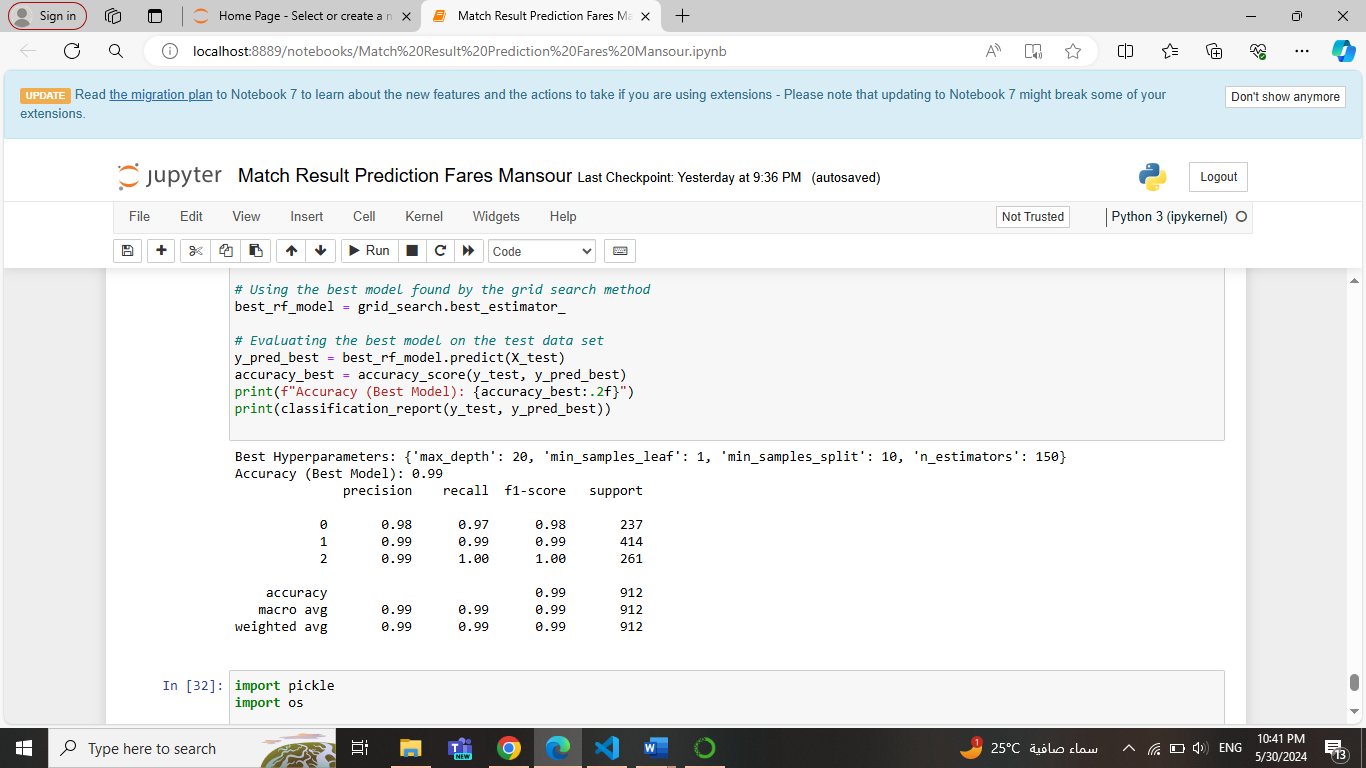
Description automatically generated

This output shows all the accuracies of the 9 models trained. SVM (support vector machine), Gradient Boosting, XGBoost, and Decision Tree were the highest among all with 99%. Followed by Random Forest with 98%. Then AdaBoost with 82%. And the other’s accuracies weren’t as good as the others.

Also, shows all the confusion matrices of the 9 models trained. Gradient Boosting and Decision Tree were the best among all. Followed by Random Forest and XGBoost. Then, AdaBoost. And the other’s confusion matrices weren’t as good as the others.

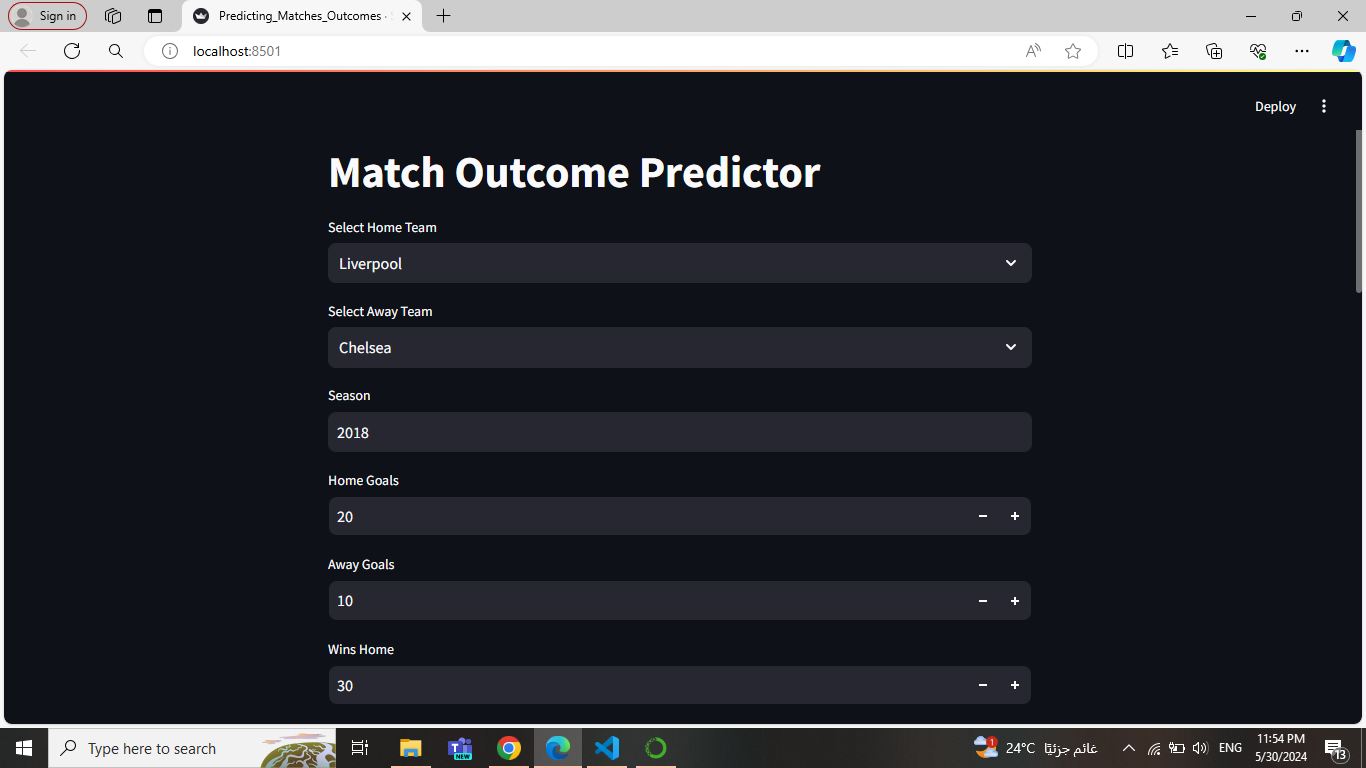
In Cross-Validation scores, Gradient Boosting and Random Forest are the highest. Followed by the others. Note that this output shows the mean cross validations not all of them.

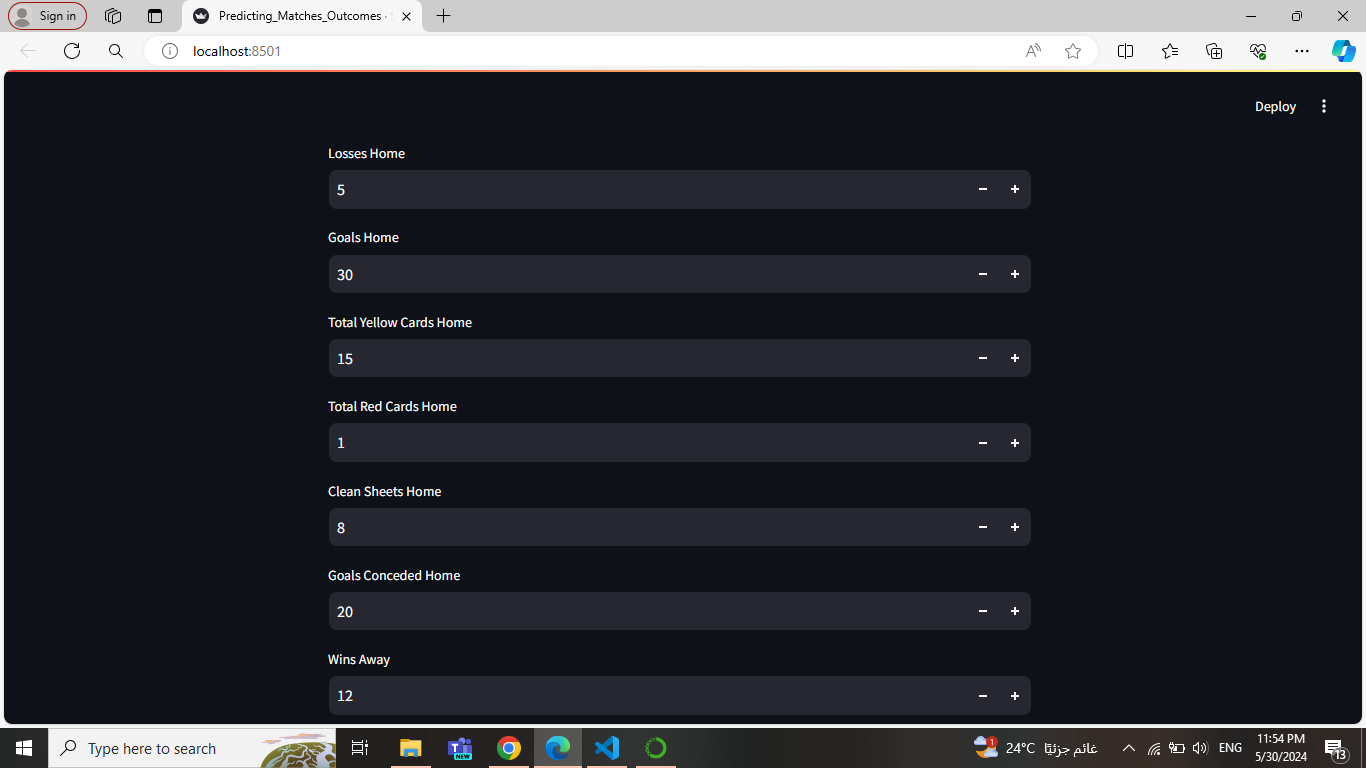
**10- Best Hyperparameters:**

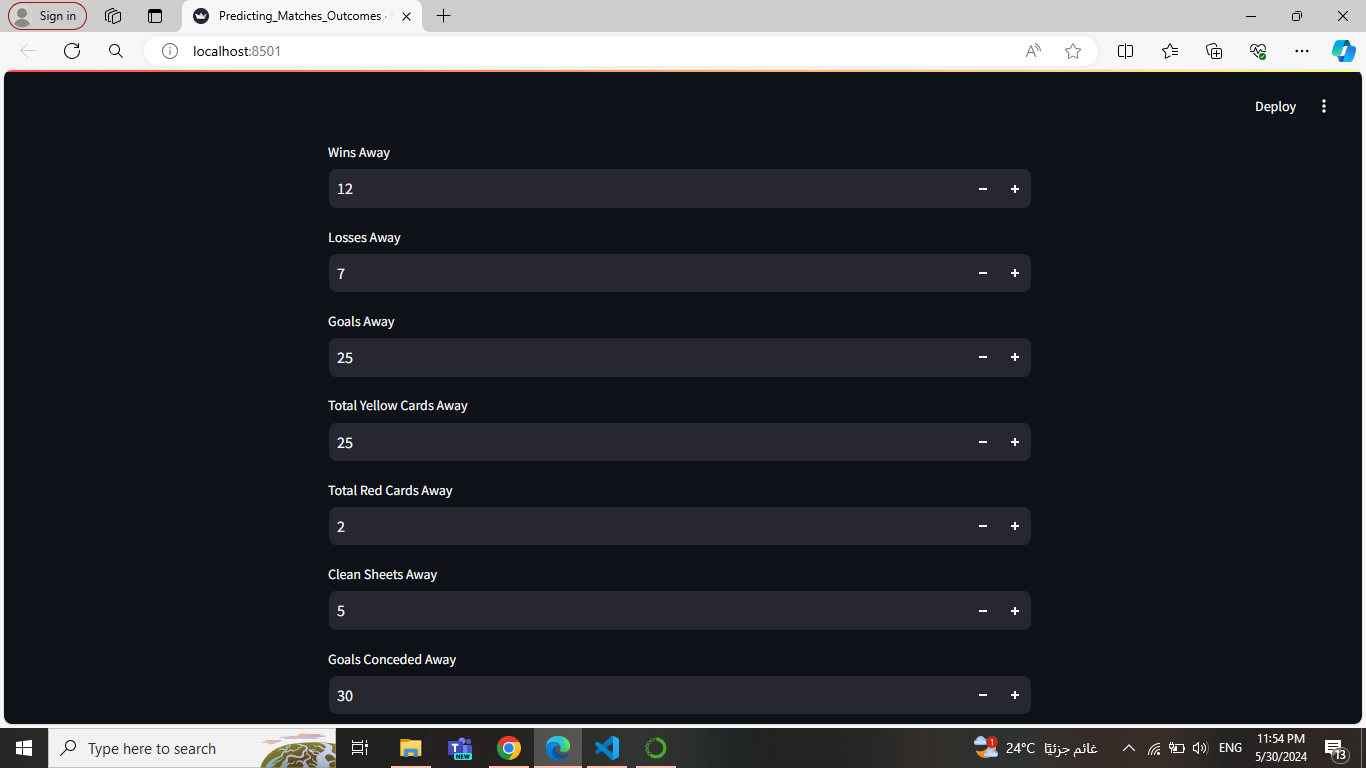


This output shows the best parameters for a model by using the grid search method. The best maximum depth is 20, the minimum sample leaf is 1, the minimum sample split is 10, and n estimators are 150. Regarding the best model, there were 4 good models. So, in every run the output changes to one of them. But this is the output for the Random Forest model.

**13- Model Deployment:**

****

****



A screenshot of a computer

Description automatically generated

This is the last step, the deployment model. These are pictures from the Streamlit web app page. It makes the user enter his numbers and the model predicts who is the winning team or if there is a Draw.

**Conclusion:** In conclusion, the project journeyed through understanding and predicting premier league match results using data analysis and machine learning. It started by cleaning and preparing the data for analysis, then transformed categorical information into numbers to improve accuracy. Nine different models were tested. Overall, the analysis indicates that Random Forest, Gradient Boosting, XGBoost, and Decision Tree are effective models for predicting match result. Due to their high evaluation metrics and accuracies. Random Forest Classifier was chosen for deployment due to its good performance and simplicity. The deployment via the Streamlit web app provides a user-friendly interface for utilizing the model's predictions. The analysis of football match prediction via machine learning offers valuable insights but also has limitations. Diverse datasets could mitigate biases, advanced feature selection techniques and model interpretability methods may enhance transparency, and addressing class imbalance and conducting external validation could bolster generalizability.

Data analysis and machine learning play pivotal roles in real-world applications by extracting actionable insights from vast amounts of data and automating decision-making processes. From healthcare to finance, and transportation, these technologies enable organizations to optimize operations, improve efficiency, and drive innovation. Moreover, in fields like healthcare, machine learning models can assist in early disease detection, personalized treatment plans, and improving patient outcomes, ultimately saving lives, and transforming industries.

**Useful Links:**

1. The YouTube link explains the codes used in Jupyter Notebook to build the models and evaluate them: <https://youtu.be/9EGeQPRVYoI>

1. The YouTube link explains the codes used in Visual Studio Code to build the Streamlit web app and how the user can interact with the deployment model: <https://youtu.be/yI9HOGjK2ls>
2. Github Link: <https://github.com/faresmansourr>
3. Project Repository Github Link: <https://github.com/faresmansourr/Match-Result-Prediction>