**Predictive Model for Shaping MMA Outcomes: A Case Study**

Chase Stalcup

University of West Florida

Department of Computer Science

Email: [cts36@students.uwf.edu](mailto:cts36@students.uwf.edu)

November 30th, 2024

**Abstract**

Determining the potential outcomes of Mixed Martial Arts (MMA) bouts within the Ultimate Fighting Championship (UFC) could have a substantial impact on official sports betting, and poses significant interest to those that are fans of the sport. This case study explores the development of a predictive model for determining the outcomes of MMA fights in the UFC using a dataset of over 6,000 entries comprising detailed fight statistics. The study aims to analyze key fight metrics and leverage advanced machine learning techniques to predict whether a fighter wins or loses. The modeling process begins with preprocessing and exploratory data analysis to understand the dataset's structure and identify relevant features. Imbalances in the outcome are addressed using Synthetic Minority Over-sampling Technique (SMOTE) to improve model performance. Logistic regression serves as the initial baseline model for prediction, providing interpretable insights into the statistical contribution of individual features. Performance evaluation includes confusion matrices and classification reports to measure metrics like precision, recall, F1-score, and accuracy. To enhance predictive performance, Random Forest classification is implemented, providing robustness to overfitting and feature importance insights. Hyperparameter tuning is conducted using GridSearchCV, optimizing the model over a range of parameters to achieve the best performance. Cross-validation ensures the reliability and generalizability of the results across different subsets of the data. This study not only demonstrates the effectiveness of these machine learning techniques in predicting UFC fight outcomes but also highlights the importance of combining traditional statistical models with ensemble learning methods for comprehensive analysis. Insights gained from this predictive modeling can aid analysts, bettors, and fight strategists in making informed decisions based on historical fighting data.

**Introduction**

While not a major problem for people across the world, determining the outcome of sporting events is considered to be a desirable achievement for fans of sports betting. Specifically looking at a sport as complex as mixed martial arts there are numerous ways for the action to unfold. Incorporating techniques from various martial arts disciplines, including striking, grappling, and ground fighting there are several features to consider when constructing a predictive model. Fighters use methods from boxing, wrestling, Brazilian jiu-jitsu, Muay Thai, karate, judo, and other combat styles.

Machine Learning (ML) is a subset of artificial intelligence that enables systems to learn patterns and make decisions or predictions based on data, without being explicitly programmed. In the context of sports like UFC, ML can be used to build predictive models that forecast fight outcomes based on historical data. This study uses UFC-Fight historical data from 1993 to 2021. The dataset includes 6012 records, and it is qualified by 131 features, however when considering a predictive model, not all of these features are going to be implemented. Focusing on attributes like significant strike percentage, average takedown percentage, and average control time, combined with physical aspects of each fighter like height, reach, and age we can apply various algorithms to build a predictive model. To solve this problem, various powerful ML algorithms (Logistic Regression, Random Forest, SMOTE, and GridSearchCV) are applied for processing the dataset. The developed fight outcome prediction model can assist sports analysts, journalists, and sports bettors to understand the grounds in which they make decisions on a deeper level.

The rest of this paper is organized in sections. Section 2 highlights Problem Description and Data Analysis will clearly define the problem to be solved and perform an initial analysis of the available data to understand its structure and relevance. Dataset Collection, how the necessary data was gathered from reliable sources, ensuring it is comprehensive and relevant to the problem. Data Preparation: cleaning and preprocessing the data by handling missing values, normalizing features, and ensuring consistency for analysis. Feature Analysis by exploring and evaluating the importance of different features to identify the most impactful ones for the predictive model. Section 3 encompasses multiple topics in the general design through Prediction Model Design and Implementation: Design the structure of the predictive model and implement it using appropriate tools and frameworks. Algorithms for Prediction: Select and apply machine learning algorithms best suited for solving the prediction problem. Prediction Model Performance Measures: Evaluate the model’s performance using metrics such as accuracy, precision, recall, or F1-score. Finally Section 4 will show the Model Prediction Empirical Results presenting the outcomes of the prediction model, including test results and insights from the predictions and thus drawing conclusions by summarizing the findings, and discussing the effectiveness of the approach, and propose potential future improvements for the final section 5.

**2 Problem Description and Data Analysis**

This project focuses on historic data from the Ultimate Fighting Championship to identify any key factors that may contribute to a fighters success in MMA, and predict the expected outcome of the matches based on key attributes. This analysis will employ data mining techniques to identify trends in the data and build a predictive model for the outcomes. By uncovering trends and potential predictors, this analysis aims to further understand the science of the sport.

**2.1 Dataset Collection**

The dataset that was acquired for this problem analysis came from Kaggle and it was gathered by Rajeev Warrier. “This is a list of every UFC fight in the history of the organisation. (as of 2021) Every row contains information about both fighters, fight details and the winner. The data was scraped from ufc stats website” (Warrier). The set had 4 sheets:

* data.csv
* preprocessed\_data.csv
* raw\_fighter\_details.csv
* raw\_total\_fight\_data.csv

The only set that was implemented in the model was the data.csv containing 6012 records, and it is qualified by 131 features.

**2.2 Data Preparation**

The data preparation process began with loading the dataset from a CSV file named fight\_data.csv into a pandas DataFrame, providing a structured format for analysis. The initial focus was on the target variable, the Winner column, which indicated the outcome of the matches. To ensure the data was suitable for binary classification, unique values in this column were examined. Rows with invalid or unexpected entries were filtered out, retaining only those in which the winner was either "Red" or "Blue." This cleaned column was then re-encoded into a binary format using LabelEncoder, where "Red" was represented as 1 and "Blue" as 0. This transformation simplified the target variable into a form suitable for modeling.

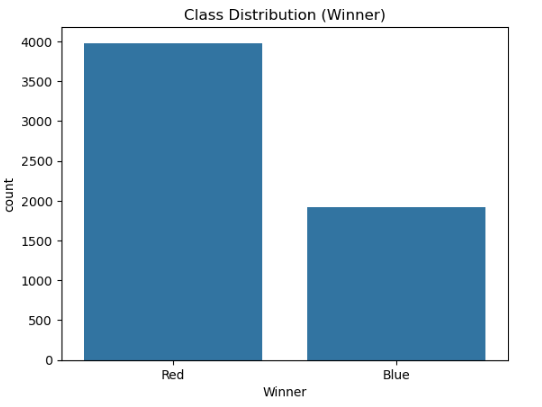
Following the target variable preparation, a set of features relevant to predicting match outcomes was carefully selected. These included statistics such as significant strike percentages, control time, total fight time, and key fighter attributes like height, reach, and age. To avoid inadvertent data modifications and warnings, a copy of these selected features was created. The resulting feature matrix, X, and the binary-encoded target variable, y, formed the basis for subsequent analysis.

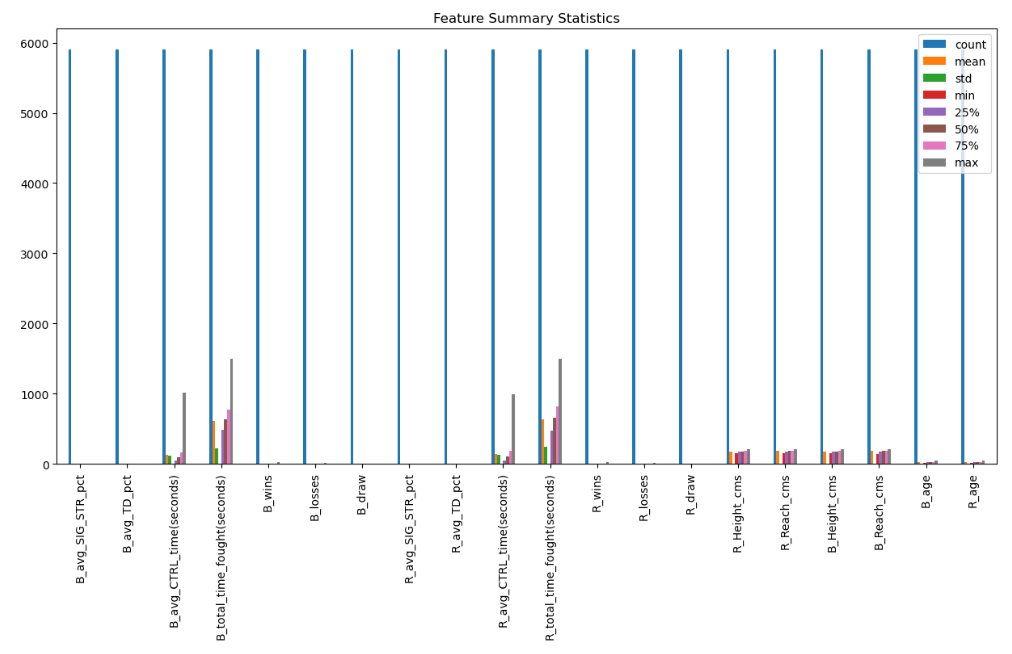
Since missing data could potentially compromise the model's performance, any missing values within the feature matrix were handled by imputing them with the median value of the respective columns. This approach maintained the consistency and integrity of the dataset without introducing bias from extreme values. To ensure the preparation process was successful, the target variable was verified to confirm that it contained only the two expected classes, aligning with the requirements of a binary classification task.

These steps collectively ensured that the dataset was clean, well-organized, and ready for use in predictive modeling. The attention to detail in cleaning, selecting, and preprocessing the data provided a strong foundation for the subsequent stages of analysis and model training.

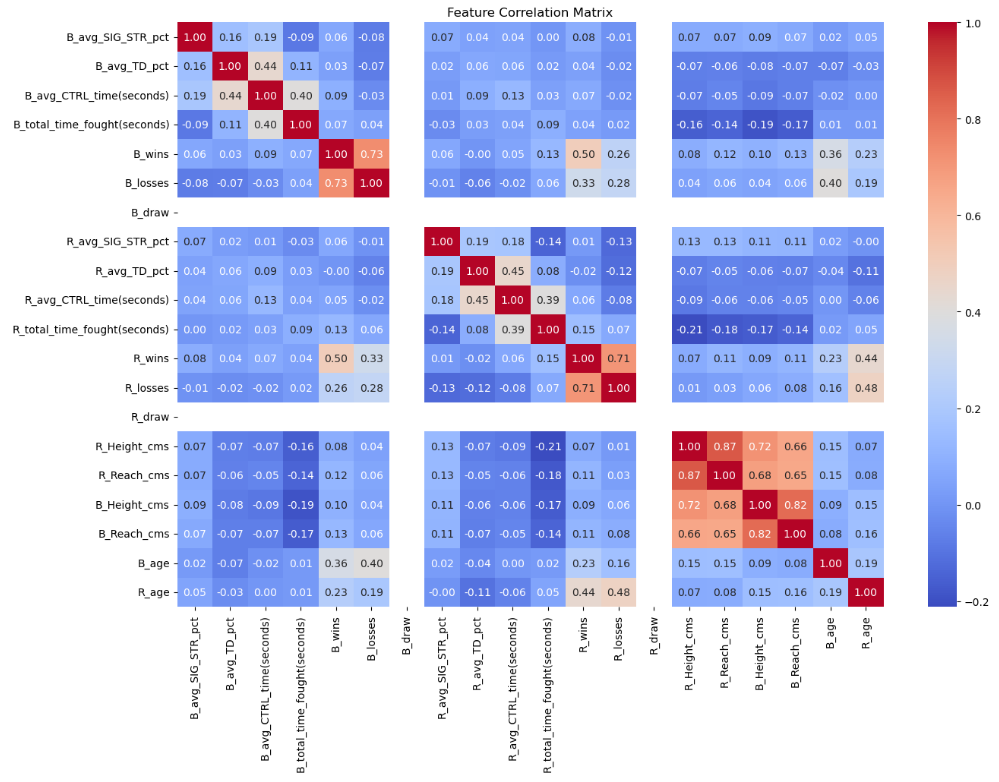
**2.3 Feature Analysis**

The feature analysis process began with an exploration of the class distribution for the target variable, Winner. Using a count plot, the distribution of "Red" and "Blue" outcomes was visualized, providing an initial understanding of class balance within the dataset. This step was essential for evaluating whether the dataset was imbalanced, which could influence the choice of model or evaluation metrics.

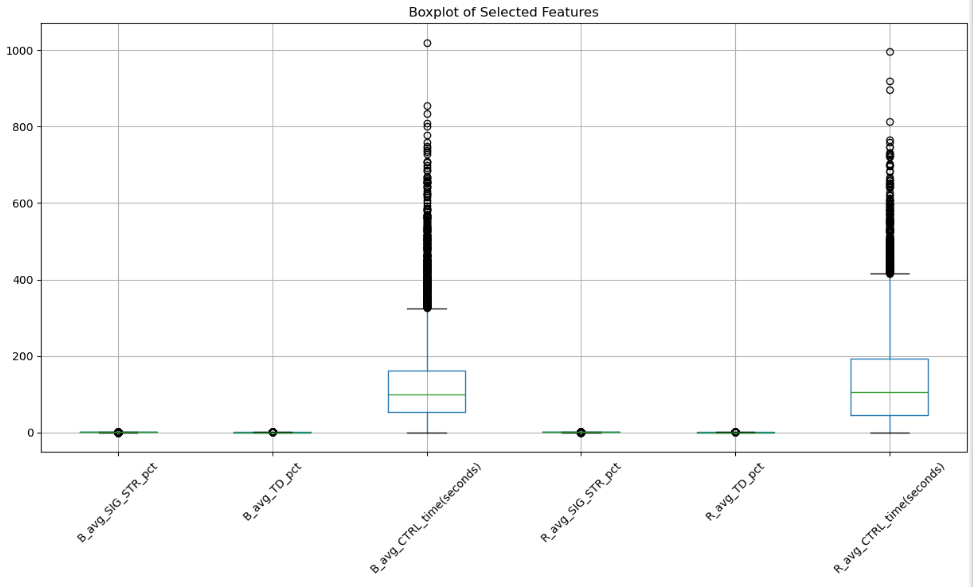
****

Next, a summary of the feature distributions was generated by calculating descriptive statistics such as mean, median, and standard deviation for each feature. These statistics were visualized in a bar chart, offering a comprehensive overview of the dataset's numerical properties. This step helped to identify potential discrepancies or irregularities in the data, such as features with minimal variance or extreme values.****

To assess relationships among features, a pairwise correlation analysis was performed. The correlation matrix was visualized as a heatmap, with annotations indicating the strength and direction of the relationships between features. This visualization provided insights into multicollinearity and potential redundancy, which could influence feature selection and model interpretation.

****

Lastly, outliers within selected numerical features were examined using boxplots. The analysis focused on key statistics such as significant strike percentages, takedown percentages, and control time for both fighters. By plotting these features, the presence of extreme values was highlighted, enabling further consideration of their impact on the modeling process.



Overall, the feature analysis provided valuable insights into the dataset’s structure, relationships, and potential challenges. These findings informed subsequent steps in the modeling pipeline, ensuring a more robust and reliable analysis.

**3 Prediction Model Design and Implementation**

The prediction model design and implementation in this code revolves around training a Logistic Regression model to predict the winner of a fight based on various fighter attributes, with additional application of Random Forest and SMOTE. First, the dataset is loaded and preprocessed by filtering the 'Winner' column to ensure that only valid categories (Red and Blue) are considered. The winner labels are then encoded into binary values, where "Red" is mapped to 1 and "Blue" to 0. The relevant features, including metrics such as significant strike percentage, takedown percentage, and fight statistics for both the blue and red fighters, are selected. The data is split into training and test sets, and missing values are handled by filling them with the median value of each feature. Before model training, the features are scaled using StandardScaler to standardize them. The Logistic Regression model is then trained using the One-vs-Rest strategy for multi-class classification. The model's performance is evaluated through various metrics, including the confusion matrix, classification report (which provides precision, recall, and F1-score), and the ROC-AUC score. The results indicate the model's ability to classify the winner of the fight with reasonable accuracy and are visualized using a heatmap of the confusion matrix and performance metrics.

**3.1 Algorithms for Prediction**

*Random Forest*

The predictive modeling process utilized a Random Forest classifier enhanced by SMOTE for handling imbalanced classes. Random Forest is an ensemble learning algorithm that builds multiple decision trees during training. Each tree operates independently and outputs a prediction, and the final classification is determined by aggregating these predictions (e.g., majority voting for classification tasks). The algorithm inherently reduces overfitting through the averaging process and by leveraging random sampling of data and features for tree construction.

*SMOTE (Synthetic Minority Oversampling Technique)*

To address class imbalance, SMOTE (Synthetic Minority Oversampling Technique) was incorporated into the workflow. SMOTE generates synthetic examples for the minority class by interpolating between existing data points. This ensures a more balanced dataset, helping the Random Forest model learn effectively without being biased toward the majority class.

*Pipeline*

The integration of SMOTE and Random Forest was achieved using a pipeline, ensuring seamless data preprocessing and model training. Hyperparameter tuning for the Random Forest model was conducted through grid search with 5-fold cross-validation. This approach systematically evaluated combinations of parameters such as the number of trees (n\_estimators), tree depth (max\_depth), and minimum samples required to split a node or form a leaf (min\_samples\_split and min\_samples\_leaf). Feature selection strategies (max\_features) were also optimized to balance model complexity and performance.

*Logistic Regression*

Logistic Regression was used as a predictive algorithm in a separate modeling process. This algorithm is a linear model that predicts probabilities for binary classification problems, making it particularly suitable for tasks like predicting match outcomes. The core of Logistic Regression lies in its ability to model the relationship between the input features and the probability of a specific outcome using a logistic function, also known as the sigmoid function. This function ensures that the predicted probabilities are constrained between 0 and 1.

*GridSearchCV*

GridSearchCV is a hyperparameter tuning technique in machine learning, used to find the best combination of hyperparameters for a given model. It systematically works through multiple combinations of hyperparameters to determine the optimal set that maximizes the model’s performance. This process helps in improving model accuracy and generalization by selecting the best parameter values.

Each iteration of the code as you can see from the .ipynb file added a new method or algorithm to achieve better results. If the accuracy was too low it was to be improved upon and after adding Random Forest alone made the accuracy perfect which is unrealistic. The best possible outcome came from adding GridSearch cross validation for the final run taking about 5 minutes to process after running the code.

Once the best hyperparameters were identified, the model was evaluated on the dataset. Metrics such as the confusion matrix, classification report, and ROC-AUC score were used to assess performance. The confusion matrix provided a detailed breakdown of correct and incorrect predictions, while the classification report summarized precision, recall, and F1 scores for each class. The ROC-AUC score offered insight into the model's ability to distinguish between the classes.

This methodology demonstrated the Random Forest algorithm's robustness in handling complex datasets, particularly when paired with oversampling techniques like SMOTE to address imbalances. The grid search ensured that the final model was fine-tuned for optimal accuracy, providing a strong foundation for reliable predictions.

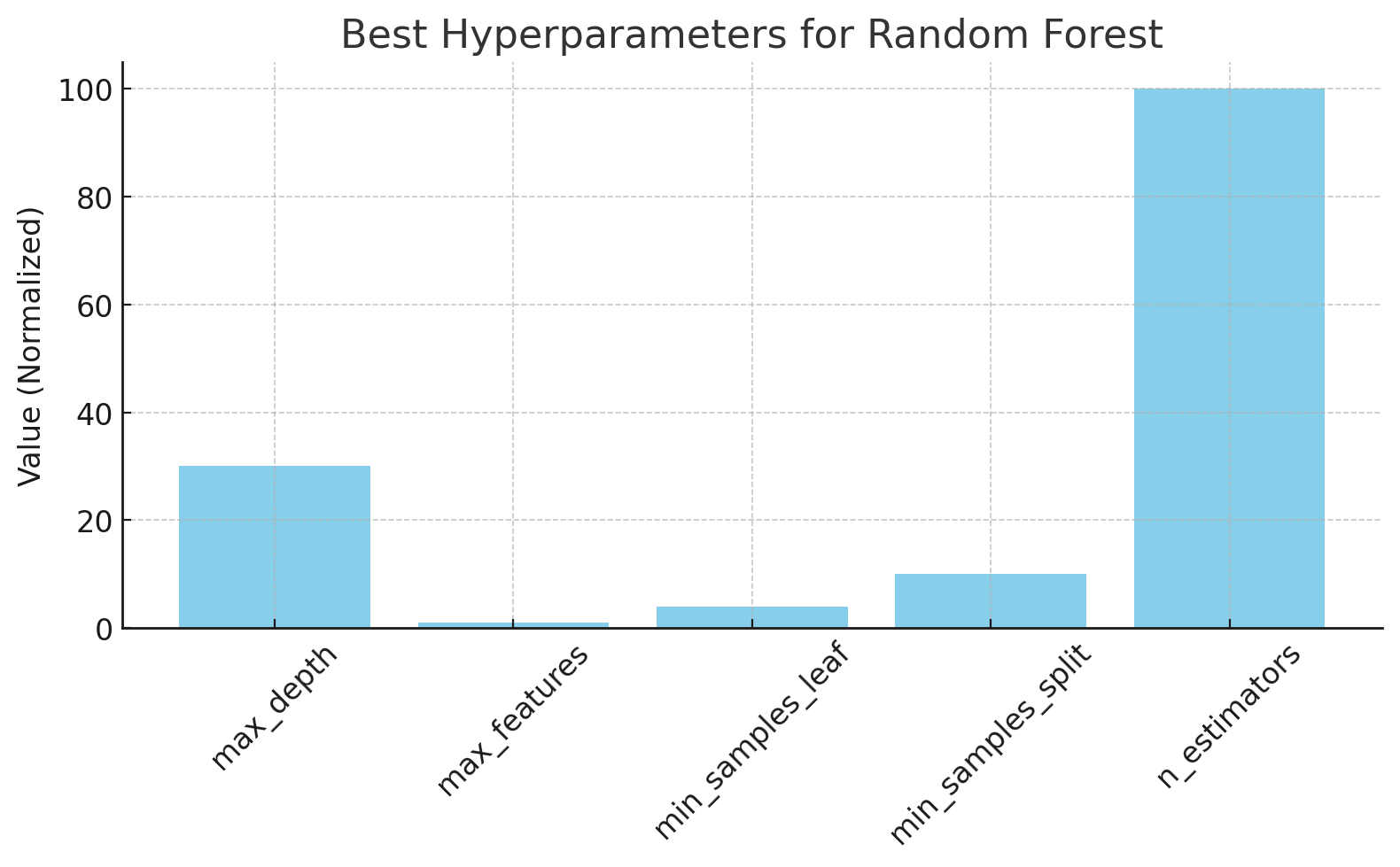
**3.2 Prediction Model Performance Measures**

The prediction model performance is assessed using several key metrics, including the confusion matrix, classification report, ROC-AUC score, and visualizations. The confusion matrix reveals the breakdown of true positives, true negatives, false positives, and false negatives, providing insight into the types of errors the model makes. The classification report includes precision, recall, F1-score, and accuracy: precision measures the accuracy of positive predictions, recall assesses the model’s ability to identify all relevant instances, F1-score balances precision and recall, and accuracy reflects the overall correctness of predictions. The ROC-AUC score evaluates the model's ability to distinguish between classes, with a higher score indicating better performance. Additionally, the confusion matrix is visualized as a heatmap, making it easier to interpret the model’s performance and identify areas of improvement. Together, these metrics offer a comprehensive view of the model's effectiveness in making accurate predictions.

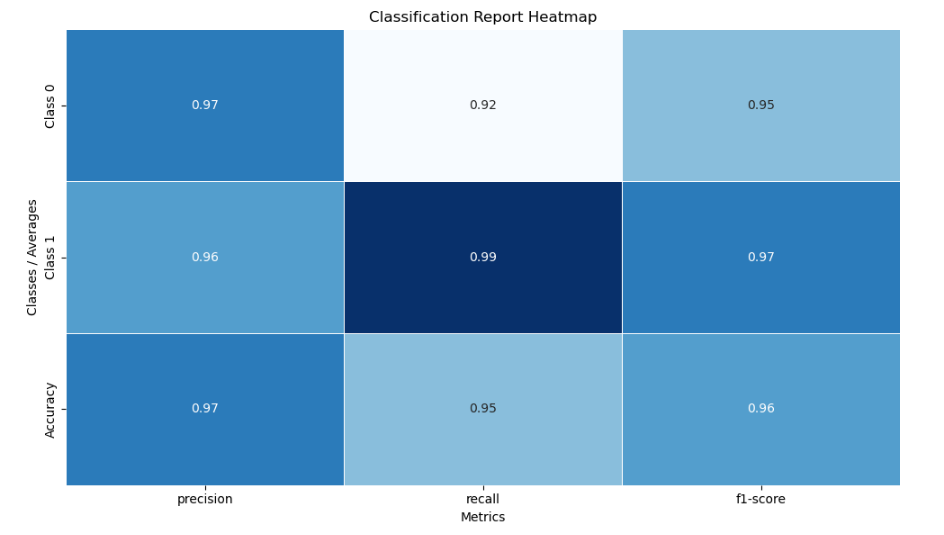
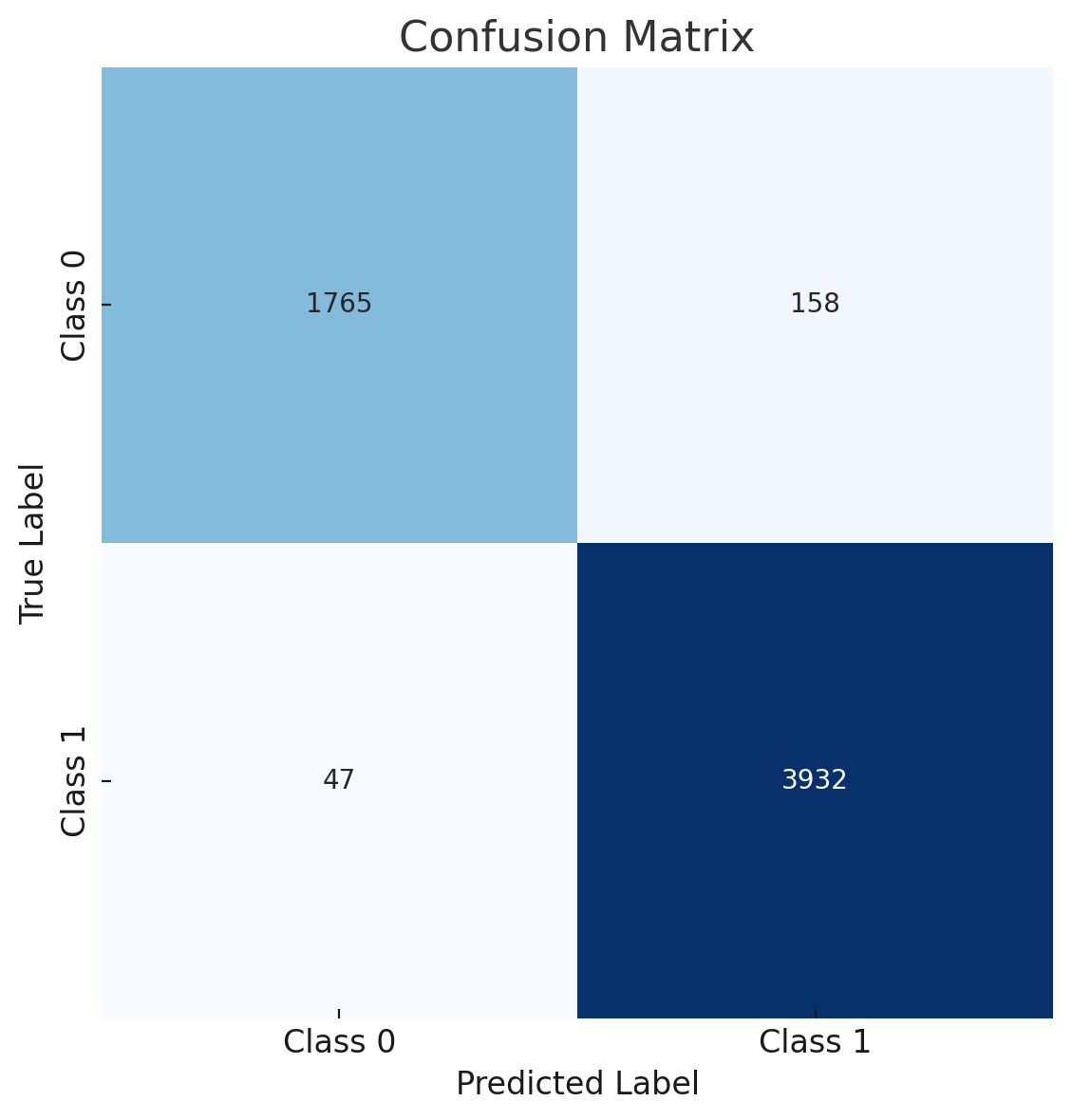
**4 Model Prediction Empirical Results**

The empirical results from the model prediction process highlight the effective performance of the Random Forest classifier, enhanced by the use of SMOTE (Synthetic Minority Over-sampling Technique) for class balancing. The primary objective of this experiment was to identify the optimal model parameters through grid search and assess the model's predictive accuracy and other evaluation metrics.

After setting up the Random Forest model, a series of hyperparameters were tuned using grid search with 5-fold cross-validation. The resulting best hyperparameters for the model were: rf\_\_max\_depth: 30, rf\_\_max\_features: 'sqrt', rf\_\_min\_samples\_leaf: 4, rf\_\_min\_samples\_split: 10, and rf\_\_n\_estimators: 100. This configuration allowed the model to balance predictive power and overfitting, producing the best results during the cross-validation process with a score of 0.6417.

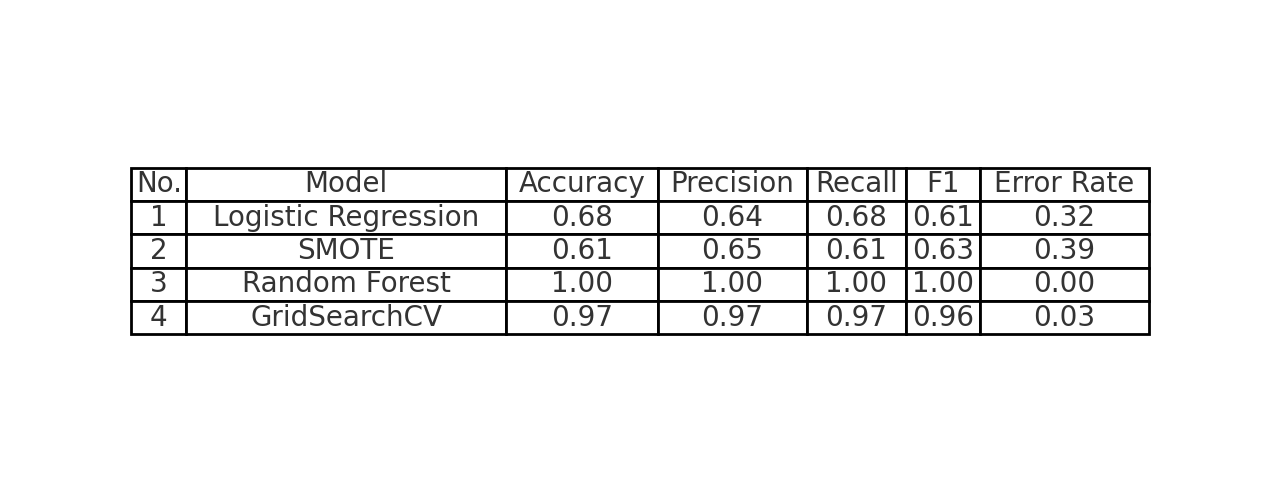
****

When the model was evaluated on the full dataset, the accuracy was found to be 97%. This result was corroborated by a confusion matrix that highlighted the model’s ability to correctly classify instances in both classes. Specifically, the model achieved 1,765 true negatives and 3,932 true positives, with 158 false positives and 47 false negatives. These numbers indicate a strong ability to accurately predict both classes. The classification report further confirmed the model's effectiveness, with precision, recall, and F1 scores of 0.97 for both classes, reflecting a well-balanced model in terms of false positives and false negatives. The macro and weighted averages, which aggregate these metrics across both classes, showed values close to 0.96, further validating the model’s consistent performance across different categories.

****

Additionally, the ROC-AUC score, which measures the area under the receiver operating characteristic curve, was calculated to be 0.9925. This high value indicates that the model has an excellent ability to distinguish between the two classes, with a minimal likelihood of misclassifying observations.

In conclusion, the results of this model prediction experiment show that the Random Forest classifier, combined with the SMOTE technique for class balancing, provides a robust and highly accurate predictive model. The model’s ability to achieve a high ROC-AUC score and impressive precision, recall, and F1 scores further supports its suitability for the classification task at hand. Below is a table showing the differing results as algorithms were adjusted after each run to achieve the best possible results.

****

**5 Conclusions**

In conclusion, the model development process involved careful data preparation, feature selection, and the implementation of predictive algorithms like Logistic Regression and Random Forest. Performance evaluation was carried out using key metrics such as the confusion matrix, classification report, and ROC-AUC score, which demonstrated the model’s ability to distinguish between classes and accurately predict outcomes. The results indicate strong predictive performance, with the Random Forest model, in particular, achieving a high level of accuracy and AUC score. These findings suggest that the implemented models are effective for the task at hand, with potential for further optimization and refinement.

The UFC fight prediction model can offer significant benefits to analysts and sports betting by providing data-driven insights into the likely outcomes of matches. By leveraging historical data and advanced machine learning algorithms, the model can identify key performance indicators that influence a fighter's success, allowing analysts to make more informed predictions. For sports bettors, this predictive tool can enhance decision-making by offering a statistical edge, potentially improving the accuracy of betting strategies and reducing risks associated with unpredictable outcomes.

**References:**

Rajeev W. (2020). UFC dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/rajeevw/ufcdata

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to data mining. Pearson.

Scikit-learn developers. (n.d.). sklearn.model\_selection.GridSearchCV. Scikit-learn. Retrieved November 30, 2024, from https://scikit-learn.org/dev/modules/generated/sklearn.model\_selection.GridSearchCV.html

Machine Learning Approaches to Real Estate Market Prediction Problem: A Case

Study. Shashi Bhushan Jha, Vijay Pandey, Rajesh Kumar Jha, Radu F. Babiceanu

(*For Case Study Style and Template*).