Football Match Probability Prediction

Ian Burns UG, Gabriel Carson UG, Ethan Head UG

Predicting a football match's outcome can be a difficult task. This is because of the complexity and the number of factors that influence the result. In this project, we focus on utilizing machine learning techniques to predict the probability of home wins, away wins or draws using a Kaggle dataset of more than 150,000 football matches. To accomplish this, dimensionality reduction techniques such as PCA and FLD were applied to preprocess the data, retaining critical features while reducing computational complexity. Classification models used were Bayesian, Non-Bayesian, Supervised, Un-supervised, Parametric, and Un-parametric classification models. These included MPP models, Neural Networks, Random forests, and Hierarchical Agglomerative Clustering evaluated with log-loss. Additionally, the BKS (Behavior-Knowledge Space) fusion technique was used to combine predictions from the two MPP models enhancing overall performance. The results show that fusion significantly improves accuracy and that Neural Networks achieve the best results. This project highlights the efficacy of machine learning in predicting sports and analytics.

1. Introduction

This project is an attempt to predict outcomes of football matches, a sport at the heart of data science for years, given a wide range of data across countless professional leagues. The Kaggle competition "Football Match Probability Prediction" provides data points on 150,000 historical world football matches from 2019 to 2021, with more than 860 leagues and 9500 teams. This comes out to 302.14 MB across 189 columns. The ability to predict the results of sporting events is a major goal of the odds-making and betting industry. Although somewhat controversial, sports betting is an extremely popular and still booming hobby worldwide. Since legalization in 2018, it has grown into a $10 billion market in the United States alone, and every projection only expects further expansion. Developing algorithms to better understand the odds behind outcomes would allow users to capitalize on such a large industry. The current best attempts at this, as presented on Kaggle, typically employ Random Forests or LSTM in their attempt to gain an edge. This project intends to improve that edge.

1.1 Task Allocation

Group Member	Presentation Contribution	Report Contribution	Experiment Contribution
Ian Burns	Dataset description, feature selection, and feature reduction	Methods, 4.1 Dataset, 4.3 Feature Reduction	Pre-Processing, Classification/Regression
Gabriel Carson	Explain the Models used. (MPP, Hierarchical Agglomerative, Random Forest, Neural Network)	Abstract, 1.2 Contribution, 4.2 Metrics, 4.4 Classification	Classification/Regression, Performance Evaluation
Ethan Head	Performance Evaluation and Model Comparisons.	Introduction, Related Works, Performance Evaluation, Work Cited	Research, Performance Evaluation

1.2 Contribution

Our project pushed to work in a new direction in probabilistic classification by implementing a BKS (Behavior Knowledge Space) framework that combines the power of two MPP (Maximum Posterior Probability) Bayesian classifiers, specifically the Euclidean and Mahalanobis distance classifiers. This allowed us to handle differently modeled decision boundaries and include probabilistic reasoning in our classification decision. Constructing the BKS table we were able to solve disagreements between classifiers and provide flexible predictions through mapping joint prediction patterns to the most probable output given the training data.

Integrating BKS in the project addressed important problems in multi-class classification. Specifically, what we designed was an implementation of BKS that extended further than majority voting by leveraging training data to assign probabilities to each combination of classifier output. This ensured that the pipeline could handle uncertain outcomes, delivering consistent results even when classifiers showed varying levels of probability. This implementation of BKS improved the classification accuracy of the MPP classifiers and reduced misclassification rates through data segments with various feature distributions.

Also, this project evaluated the random forest classifiers incorporating domain-specific preprocessing and a feature weighting mechanism for the dataset's structure. On top of this, unlike normal random forest implementation, this shows the integration of the fit_predict_random_forest method which demonstrates a modular design allowing for testing alternate configurations and models in the pipeline. Moreover, the validation of the model on a separate validation dataset ensures consistency and provides a measure of real-world performance.

2. Related Works

Numerous studies have explored predicting sports outcomes, especially in football, employing a wide range of methods and features. Constantinou et al. developed the pi-football model, which integrated both objective and subjective features into a Bayesian network. Similarly, Tax and Joustra employed a variety of dimensionality reduction and classification methods, including principal component analysis, Naïve Bayes, Neural Networks, Random Forests, and Genetic Programming, while using grouped feature sets closely related to those used in pi-football. Ren and Susnjak expanded on this approach by experimenting with simple and advanced machine learning techniques such as Linear Regression, K-Nearest Neighbors, Decision Trees, Support Vector Machines, and Neural Networks. Beyond football, the NCAA March Madness tournaments, with their competitions on Kaggle, have emerged as a benchmark due to abundant historical data and a predictive environment. The accomplishments of Kaggle Grandmaster Darius Barušauskas, who topped the charts by employing methods like XGBoost and Generalized Additive Model, offer a blueprint for success in sports prediction tasks. Together, these works highlight the breadth of established methodologies and their potential to inspire new models.

3. Methods

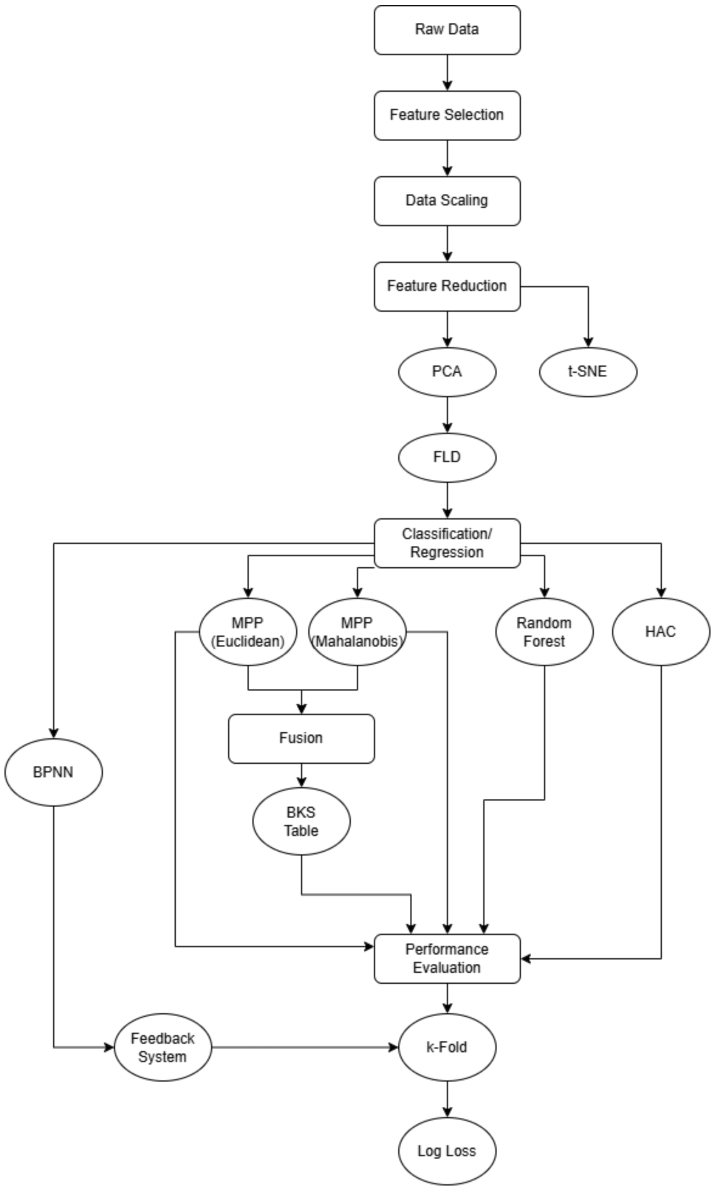

Figure 1: Workspace Flow Chart

Our workflow for this project follows the flow chart in Figure 1, beginning with selection of features from the original dataset and scaling our chosen features. With the scaled data, apply PCA first to begin reducing the number of features, and then apply FLD to bring the data down to 2D. We also applied t-SNE to the scaled data, but got poor results. We then applied five classifiers to our PCA + FLD reduced data: BPNN, MPP with Euclidean distance, MPP with Mahalanobis distance, Random Forest, and Hierarchical Agglomerative clustering. We also used BKS to apply fusion to our Mahalanobis and Euclidean classifiers, with the hope of improving performance. We then evaluated the performance of the five original classifiers, as well as the result of the BKS-based fusion, using k-Fold cross validated log-loss. The back-propogation of our BPNN serves as our feedback system.

4.1 Dataset

The dataset contains over 150,000 historical football matches from 2019 to 2021, covering 860+ leagues and 9,500 teams. It includes train.csv (with target labels: home, away, draw) and test.csv (without target labels). The train and test files both have 187 unique features, but the test file does not have a 'target' column that shows the correct match result. The goal is to predict probabilities for match outcomes using train.csv and make predictions for test.csv for submission.

4.2 Metrics

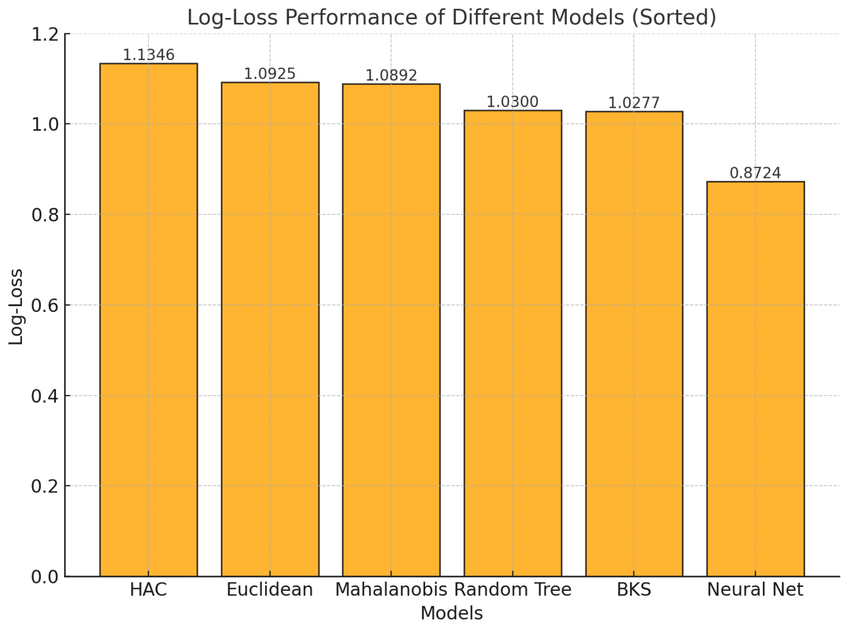

Figure 2: Log-Loss Performance of Different Models

Overall, the performance of these models varied. The MPP classifiers had log losses that were similar, Euclidean being 1.0925 and Mahalanobis being 1.0892 indicating moderate prediction accuracy. The HAC classifier performed worse than the MPP models with a log-loss of 1.0974. The Random Forest classifier showed good performance achieving a log loss of 1.0300, closely getting beat by the log loss of BKS fusing the two MPP classifiers resulting in a log loss of 1.0277. Coming far in the lead, even with 10 epochs, the Neural network outperformed the other models with a log loss of .8724 showing its capability to capture intricate relationships in the data by correcting its mistakes with back propagation.

4.3 Feature Reduction

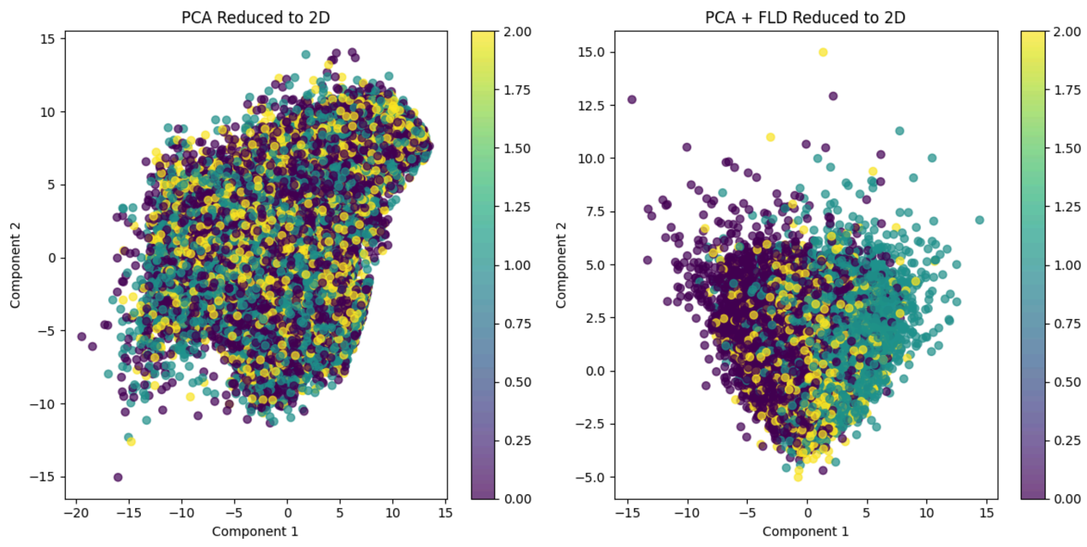

Figure 3: PCA vs PCA + FLD Performance

For our feature reduction, we first implemented PCA, using an error threshold of 0.05, or 5%, which reduced the data to 123 features. We then applied FLD to the resulting data, which reduced the dataset to two dimensions, given that our data has three classes. We alternatively tested reducing the data to two dimensions using just PCA, but Figure 3 shows that combining PCA and FLD had far better results when it comes to separating it.

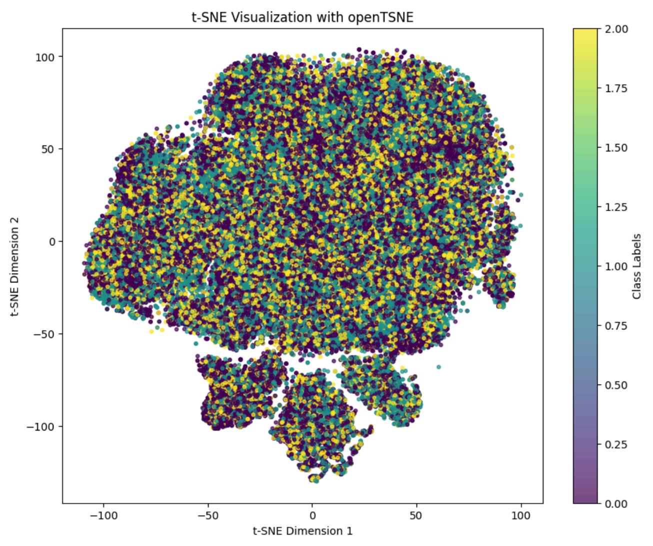

Figure 4: t-SNE Visualization

We implemented t-SNE, but had poor performance due to the nature of the features. As shown in Figure 4, t-SNE had a tough time finding any kind of linear separation of the data. t-SNE commonly has issues reducing datasets with a very high dimensionality, which is likely the reason it could not separate our dataset's 187 features. In future work, applying t-SNE to a slightly reduced dataset could provide better results.

4.4.1 MPP

The MPP (Maximum Posterior Probability) implementation includes Euclidean and Mahalanobis distance metrics and evaluates class probabilities by getting the distance from test points to class means. The Euclidean MPP computes linear distances without taking feature correlation into account. On the other hand, the Mahalanobis MPP uses the covariance structure of each class which gives sensitivity to feature relationships. The Euclidean distance classifier achieved a log loss of 1.0925 indicating a slightly higher uncertainty compared to the Mahalanobis classifier boasting a log loss of 1.0892.

4.4.2 HAC

The HAC (Hierarchical Agglomerative Clustering) classifier groups the data into clusters using a bottom-up approach assigning class probabilities based on cluster-to-class associations. This implementation uses Ward linkage which achieves the best performance on this data. It calculates the class probability for test samples by finding the closest cluster and then using cluster-to-class mappings. The classifier was evaluated with PCA +FLD data and achieved a log loss of 1.1346, showing it performed worse than MPP classifiers while taking more than 5 times longer to run. Why is this? This is likely due to the unsupervised nature of the clustering and the reliance on too few data points for the computation.

4.4.3 Random Forest

The Random Forest classifier uses a unit of decision trees to predict probabilities of classes by averaging the output of individual trees. Our implementation used 100 trees with no max depth restriction, making sure the model could capture intricate correlations in the data. It achieved a log loss of 1.0300, outperforming both MPP and HAC classifiers. This shows the random forest's ability to handle non-linear patterns and combines multiple trees making it more effective for the given task supporting the placement of the top contenders on the Kaggle competition.

4.4.4 BKS

The BKS (Behavior-Knowledge Space) implementation combines the predictions from the Euclidean and Mahalanobis MPP classifiers to improve overall performance. It uses a probabilistic BKS table to map prediction pairs from both classifiers to class probabilities based on training data. For unseen prediction pairs, uniform probabilities are assigned as fallback. This fusion approach leverages the strengths of both distance metrics, so when evaluated on the validation data, it achieves a log loss of 1.0277. This result shows that combining the two classifier's predictions through BKS can outperform each classifier individually, showing the value of ensemble methods in improving accuracy while also keeping runtime below 5 minutes.

4.4.5 Neural Network

Our Neural Network implementation uses a feedforward architecture with one hidden layer of 64 neurons and a softmax output layer for classification. We trained the model for 10 epochs with mini-batches of size 32 and a learning rate of 0.01, the network generally would run for approximately 15 minutes. Despite the long training time in comparison to the other models, its log loss was even lower than the winners on the Kaggle leaderboards sitting below 0.9 at 0.8724, outperforming all of our other implemented models. This highlights the strength of the neural network's ability to capture complex, non-linear patterns in the data by utilizing backpropagation to correct its past mistakes, which makes it the best-performing model in our experiments.

4.5 Performance Evaluation

Our performance evaluation approach centered on the use of Log-Loss as a probabilistic scoring metric, as it penalizes overconfidence in incorrect outcomes. It is also the most common form of evaluation amongst sports predicting competitions for said reason. We complemented this metric with a K-fold validation scheme to guard against overfitting. In each fold, we applied hierarchical agglomerative clustering on different data subsets to further validate model stability. Our findings showed that the Neural Network model performed best, followed by a BKS approach and a Random Tree model. Distance-based methods such as Euclidean and Mahalanobis clustering demonstrated comparatively lower predictive accuracy, notably behind the BKS table that fused them. Our results from the Neural Network specifically drastically improved upon the Kaggle competition's winner, although we had the luxury of a feedback system while they did not.

5. Discussion

Looking ahead, we believe that sourcing more comprehensive, high-quality datasets could enhance model accuracy and that scaling neural networks on more powerful hardware would allow for deeper architectures and more thorough hyperparameter tuning. The lessons learned from this project show the importance of data quality and proper feature reduction. These insights serve as a solid foundation for future work aimed at building even more robust predictive models in sports analytics.

6. References

Barušauskas, Darius. "1st Place Solution: Women's Machine Learning Competition 2018." Kaggle Discussion Post, April 2, 2018. https://www.kaggle.com/competitions/womens-machine-learning-competition-2018/discussion/53597.

Constantinou, Anthony C., Norman E. Fenton, and Martin Neil. "pi-Football: A Bayesian Network Model for Forecasting Association Football Match Outcomes." Knowledge-Based Systems 36 (December 2012): 322–339. https://doi.org/10.1016/j.knosys.2012.07.008.

"Football Match Probability Prediction." Kaggle Competition. Accessed November 26, 2024. https://www.kaggle.com/competitions/football-match-probability-prediction.

Ren, Yiming, and Teo Susnjak. "Predicting Football Match Outcomes with Explainable Machine Learning and the Kelly Index." arXiv preprint arXiv:2211.15734, November 28, 2022. https://doi.org/10.48550/arXiv.2211.15734.

Tax, Niek, and Yme Joustra. "Predicting the Dutch Football Competition Using Public Data: A Machine Learning Approach." September 2015. https://doi.org/10.13140/RG.2.1.1383.4729.

"Why U.S. Sports Betting Could Become a $45 Billion Market." Goldman Sachs Insights article. Accessed December 12, 2024. https://www.goldmansachs.com/insights/articles/why-us-sports-betting-could-become-a-45-billion.