Rohan Chaturvedi

MMA Fight Outcome Prediction

Introduction

When considering the wealth of statistical and technical analyses present in the world of sport, its absence in one of the fastest growing sports in the world [1], Mixed Martial Arts (MMA), seems ironic. Therefore, this project aims to develop a binary classification model to predict the outcome of an MMA fight (win or loss for a given fighter) based on a set of features extracted from fight data. Two approaches to modelling are considered; the first being a hybrid of logistical regression using LU factorisation and a feed-forward neural network, and the second being an SVM classifier which leverages QR decomposition. These two models are then compared via various statistics and visualisations to assess their viability and usability in reliably predicting the outcome of a fight. The use cases for such a model are endless and range from informing bookmakers to a tool for fight promotions such as the UFC to use internally or in a consumer facing setting.

All aspects of the project, from developing the models through to the evaluation and testing, was conducted in Python in a Jupyter Notebook environment. Various libraries such as scipy, numpy, sklean, tensorflow, and others were used to facilitate work with linear algebra, machine learning, and visual evaluations.

Method

I. Database Identification and Data Pre-processing

An appropriate dataset was first found via Kaggle [2], and is a continuously updated dataset which contains 513 different properties of every UFC fight from March 1994 until the present day. Following this, the data has to be pre-processed. This consisted of first removing entries which have missing data in certain columns, then splitting the data into training, validation, and test sets. This allows us to have three distinct data sets so that we can accurately train and test our respective models on. The split used was 60% used for training, and 20% for validation and testing respectively. Once this is completed, the data can be normalised by adjusting it to a normal distribution centred at 0.

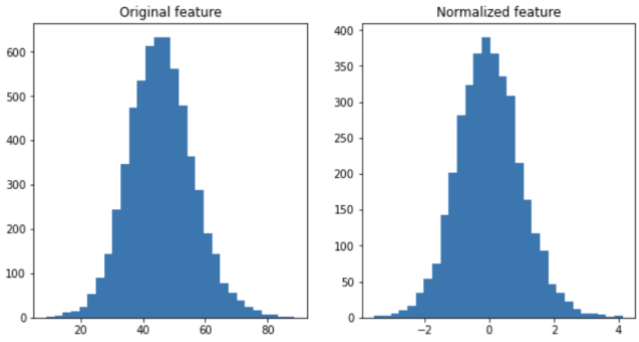

Figure 1: Visualisation of data normalisation

II. Model 1: Logistic Regression and Neural Network

The method used to develop this model was first used to identify essential markers in lung cancer data, and to facilitate prediction of the probability that one develops lung cancer [3]. The method is shown in Figure 2.

The first step would be to complete the least squares approximation using LU factorisation for the dataset. This is done using the scipy.linalg library in Python, with the solve function used to complete the forward and backward substitution.

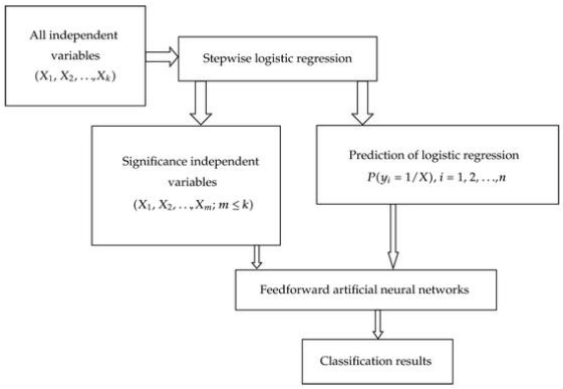

Figure 2: Model 1 methodology [3]

Considering that a logistic regression of all 513 features would result in an inaccurate model, regularisation must be performed to reduce overfitting and improve the generalisation of the model – resulting in a more refined and predictive model. In this instance, elastic net regularisation at a ratio of 0.5 is used to leverage both L1 and L2 regularisation techniques. L1 allows the parameters which are irrelevant have their coefficients reduced, and L2 allows variables with high correlations (such as submission attempts and takedowns landed) to be accounted for. Figures 3 and 4 show how these are applied to the cost function.

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Figure 3: Cost function for L1 regularisation [4]

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Figure 4: Cost function for L2 regularisation [4]

The significant independent variables and predictions from this logistic regression are then fed into a feedforward neural network as per Figure 2 and is trained on the training dataset through 50 Epochs. This Epoch value was selected as it is the point at which our evaluation statistics began to plateau, thus we avoid overfitting to the training dataset. The sigmoid function is used in this neural network as we are creating a binary classification method, and we use the Adam algorithm for optimisation due to its high computational efficiency and its ability to leverage other optimisation algorithms such as AdaGrad and RMSProp. The choice of the loss function is cross-entropy loss as we are working with a binary classification model, and this function is designed to measure the distance between predicted probability distribution and true distributions.

Following the construction of this model, the evaluation steps outlined in (IV) are then followed to assess the quality of the developed model.

III. Model 2: SVM Classifier using QR Factorisation

The method for Model 2 is one which leverages QR factorisation to then develop an SVM classifier using refined parameters. The first step is to perform a QR factorisation on the training set using scipy.linalg library. Then, the validation and test datasets were transformed using the R matrix that was obtained from the training data. This has the effect of reducing the complexity and increasing the efficiency of the SVM training. Furthermore, this method of factorisation is typically more numerically stable than LU factorisation as used in Model 1.

To refine the number of hyperparameters the SVM classifier will work with, Randomised Search Algorithm is used due to its significantly lower processing time as opposed to Grid Search which will iterate all possible combinations of hyperparameters. Furthermore, it has a faster convergence and allows for scalability going forwards. To refine which hyperparameters to use in the SVM training, Recursive Feature Elimination (RFE) is used due to its iterative removal of least importance features. This has the benefit of lowering the chance of overfitting to training data, improving the interpretability due to the lower number of features, as well as reducing the cost of training the SVM classifier. This overall process will cut down the number of features iteratively until the optimal number of features is reached and will choose the most important features. This is combined with 5-fold cross validation to help reduce overfitting and to compare different hyperparameter configurations, aiding in our feature selection. Using a k value of 5 allows for a balance between computational efficiency and model evaluation reliability (as we still have roughly 720 rows per fold).

Following this, the SVM classifier is then constructed with kernel methods to allow for learning of non-linear decision boundaries where data is not linearly separable. This also adds versatility due to the use of various kernel functions (such as sigmoid, RBF, etc). The model is then evaluated as per the methodology outlined in (IV).

IV. Evaluation Methodology

In the evaluation of each model, accuracy, precision, recall and F1 scores are used to analyse the quality of the model. The accuracy score is used to represent the number of correctly classified data instances over the total data instances, whereas the precision tells us the frequency (or infrequency) of false positives. Conversely, recall helps identify the sensitivity of the model, or the true positive rate, and helps identify the frequency of false negatives. The F1 score is the harmonic mean of precision and recall. Furthermore, visualisations are created for the model's performance on the validation set. This includes a confusion matrix to show the number of true positive and negatives, an ROC curve to visualise the ratio of the true positive rate against the false positive rate, and a precision-recall curve to visualise the overall performance of the model. Ideally, the goal is to achieve an area under each curve as close to 1 as possible.

Results, Evaluations and Conclusions

I. Model Results

Model 1 demonstrates a strong performance on the training set, achieving an accuracy of 0.871, precision of 0.867, recall of 0.875, and an F1 score of 0.871. On the validation and test sets, the model maintains good performance, with accuracy scores of 0.816 and 0.817, respectively. The confusion matrix suggests that the model is effective in classifying both positive and negative outcomes. The ROC curve has an AUC of 0.8, indicating reasonable discriminative ability. The precision-recall curve exhibits a moderate average precision of 0.74, which is satisfactory.

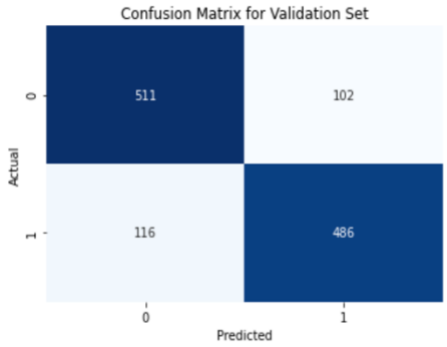

Figure 8: Confusion Matrix for Model 1

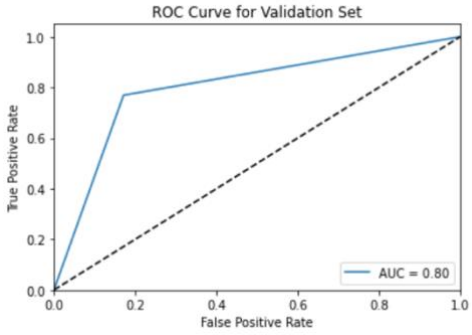

Figure 8: ROC Curve for Model 1

```
Training set evaluation:
Accuracy: 0.871
Precision: 0.867
Recall: 0.875
F1 score: 0.871

Validation set evaluation:
Accuracy: 0.816
Precision: 0.828
Recall: 0.794
F1 score: 0.811

Test set evaluation:
Accuracy: 0.817
Precision: 0.821
Recall: 0.798
F1 score 0.809
```

Figure 8: Evaluation metrics for Model 1

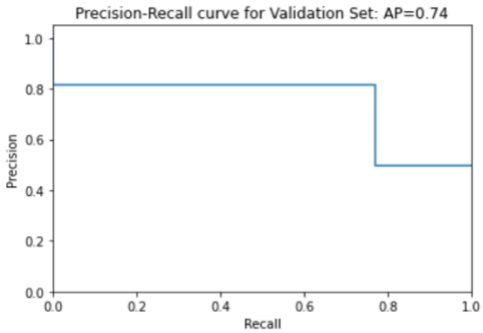

Figure 8: Precision-Recall Curve for Model 1

Model 2, on the other hand, employs random search, RFE, and 5-fold cross-validation to determine the optimal number of features, which is found to be 30. The training set performance is slightly lower compared to Model 1, with an accuracy of 0.7916, precision of 0.8197, recall of 0.7436, and an F1 score of 0.7798. The model performs similarly on the validation and test sets, with accuracy scores of 0.7992 and 0.7969, respectively. The confusion matrix indicates a somewhat weaker performance in classifying positive outcomes compared to Model 1. However, Model 2 shows a higher AUC of 0.88 in its ROC curve, suggesting better discriminative ability. The precision-recall curve displays a higher average precision of 0.87, signifying improved precision and recall trade-off.

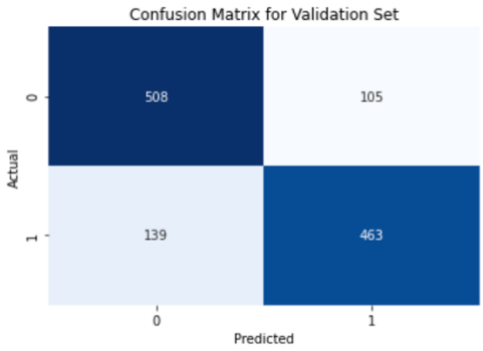

Figure 12: Confusion Matrix for Model 2

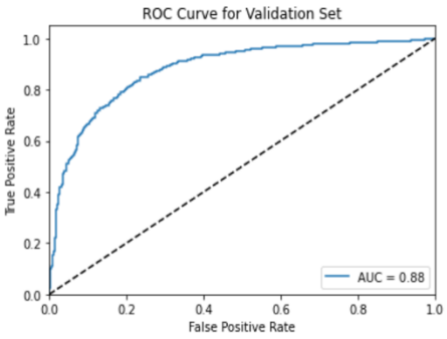

Figure 12: ROC Curve for Model 2

```
Training - Accuracy: 0.7916
Training - Precision: 0.8197
Training - Recall: 0.7436
Training - F1 Score: 0.7798
Validation - Accuracy: 0.7992
Validation - Precision: 0.8151
Validation - Recall: 0.7691
Validation - F1 Score: 0.7915
Test - Accuracy: 0.7969
Test - Precision: 0.8134
Test - Recall: 0.7572
Test - F1 Score: 0.7843
```

Figure 12: Evaluation metrics for Model 2

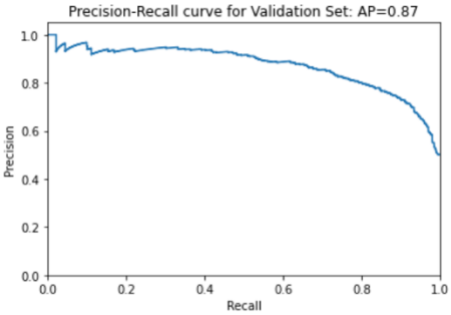

Figure 12: Precision-Recall Curve for Model 2

II. Evaluation, Conclusion and Further Applications

Both models have their respective strengths and weaknesses. Model 1 demonstrates overall higher accuracy, while Model 2 excels in discriminative ability and precision-recall trade-off [5]. When evaluating the flexibility for future development and scalability, Model 2 seems to be the more suitable choice.

Model 2's superior discriminative ability and precision-recall performance indicate its potential for handling complex or diverse datasets [6]. Its use of RFE for feature selection and hyperparameter tuning can adapt more easily to new data, making it suitable for integrating live fight data and additional inputs [7].

In contrast, Model 1 could be more suitable in scenarios where high accuracy is a priority and data inputs are relatively stable, as it demonstrates superior performance on this metric.

The SVM classifier used in Model 2 is a robust choice for handling various types of data [8]. The adaptability and performance of Model 2 provide more room for future improvements and make it a promising choice for MMA fight outcome prediction, especially when considering the integration of live fight data and the potential for increased data inputs in the future.

Works Cited

[1] Sports Brief, "Top 10 fastest-growing sports in the world right now," 14 March 2023. [Online]. Available: https://sportsbrief.com/other-sports/35933-top-10-fastest-growing-sports-world-fastest-growing-sport/. [Accessed 27 April 2023].

[2] D. McInerney, "UFC/MMA Biggest Dataset With Differentials," Kaggle, 17 April 2023. [Online]. Available: https://www.kaggle.com/datasets/danmcinerney/mma-differentials-and-elo?select=masterMLpublic.csv. [Accessed 27 April 2023].

[3] T. Tunc, "A New Hybrid Method Logistic Regression and Feedforward Neural Network for Lung Cancer Data," 5 December 2012. [Online]. Available: https://doi.org/10.1155/2012/241690. [Accessed 2023 April 27].

[4] A. Nagpal, "L1 and L2 Regularization Methods," 13 October 2017. [Online]. Available: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c. [Accessed 27 April 2023].

[5] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *Proceedings of the 23rd International Conference on Machine Learning,* no. ICML 2006, pp. 233-240, 2006.

[6] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[7] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines," in *Machine Learning*, 2002, pp. 46(1-3), 389-422..

[8] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 20(3), 273-297.