EXPERIMENT REPORT - 4

Student Name	Ronik Karki
Project Name	Assignment 1 week 4
Date	08/09/2023
Deliverables	Notebook Name: karki_ronik-24886412-week4_linear_discriminant_analysis.ipynb

---

1. EXPERIMENT BACKGROUND	
1.a. Business Objective	The main goal of this project is to predict if a college basketball player will be drafted to join the NBA league based on his statistics for the current season. Since there are only two possible outputs, the result will be used to classify whether the player will be drafted or not. As the data is highly imbalanced (i.e. class-0 has 99.04% more target values compared to class-1), using accuracy scores would be useless as the model which predicts all output as class-0 would have an accuracy of ~99.04%. Therefore the good metrics for this classification would be a high precision score, recall score, roc score, and f1 score for class 1 as we are more interested in finding how better the model performs in classifying and predicting the positive class. If the model misclassifies then the business will face two types of loss. 1. False Positive Error: In this, the model predicts that a member will be drafted but in reality, the member won't be selected. This error would increase the expectations of the viewers and hurt them as the player won't be drafted. 2. False Negative Error: In this error, the model predicts that a player will not be drafted. However, this would be more surprising to the spectators as the players who are not predicted to be selected will be drafted. Therefore, a good model should be able to minimize both errors in creating a reliable model.
1.b. Hypothesis	The experiment is based on an attempt to improve the scores obtained from the previous experiment. The hypothesis for this experiment is that the model will improve scores if we impute values according to the feature types. Furthermore, using different models and hyperparameter tuning the models might also improve the model performance. In addition, removing the multicollinearity of the features might help the model decrease confusion.

1.c. Experiment Objective	The expectation from this experiment is that we might see an increase in the correlation strength of the features once the imputation of the feature is done according to the data type of the features. More feature engineering and selection of the features might improve the scores. Furthermore, we might see an improvement in the scores by using different models and hyperparameter tuning.

---

2. EXPERIMENT DETAILS	
2.a. Data Preparation	For the data preparation part, the cleaned file was imported from the last experiment. It was imported and the imputation was applied differently as explained in the feature engineering section.
2.b. Feature Engineering	For this experiment, three types of feature engineering techniques were used: 1. Data imputation: a. All of the categorical features that had missing values were imputed by mode values. b. Numerical features were further analyzed and they were replaced with either 0 or with mean. For example, if the height column has a missing value and is being replaced with 0 then it makes no sense. Therefore, features like this were imputed with mean rather than 0. 2. Label encoding: a. All of the categorical features were converted to integers by applying label encoding techniques. 3. Feature transformation: a. Features that were combined like 'dunksmiss_dunksmade', 'rimmade_rimmiss', and 'midmade_midmiss' were used for gaining the values of a miss for the respective fields and they were transformed into the new feature. b. Highly missing variable like 'pick' was converted to categorical feature (1 if there is a value and 0 if the value is missing)) as imputing any values would make the model biased according to the value imputed.
2.c. Modelling	Feature selection techniques were used to remove the features that were highly correlated with each other using the Pearson correlation matrix. If the features were correlated and had a strength of greater than 95% then they were removed. This removes the model complexity and also one of the features would be enough for giving the information captured to the model. For the robustness of the model, stratified cross-validation was used to split the train and validation data for 10 splits. Stratified K-fold cross-validation was used as the dataset was highly imbalanced. It was tested in the logistic regression model first and

then the gradient boosting model was used to observe the difference between models and their performances.

The features were then fitted in several machine learning models like:

1. Discriminant Analysis models: Linear discriminant Analysis model and Quadratic Discriminant analysis model
2. Linear model: logistic regression model
3. Ensemble model: XGB, LightGBM, Gradient boosting model, and Ada boost model
4. Naive Bayes model: Gaussian naive Bayes model

---

3. EXPERIMENT RESULTS

3.a. Technical Performance

There were many results and testing done throughout this experiment. The two most significant models were the Linear Discriminant Analysis model and XGBoost

1. Linear Discriminant Model:

   The following outputs were obtained on a 10-fold cross-validation technique.

Splits	**Training**		**Testing**	
Model	**Scores**	**Fluctuations**	**Score**	**Fluctuations**
Precision	0.38674	0.00316	0.38832	0.030004
Recall	1.0	0.0	1.0	0
F1	0.55776	0.00329	0.55881	0.03053
AUC	0.99726	0.00006	0.99723	0.00075

2. XGBoost Model:

   The following outputs were obtained on a 5-fold cross-validation technique.

Splits	**Training**		**Testing**	
Model	**Scores**	**Fluctuations**	**Score**	**Fluctuations**
Precision	1	0	0.75934	0.037269
Recall	1	0	0.71086	0.03743
F1	1	0	0.73326	0.019890
AUC	1	0	0.99804	0.00019

3. Tuned XGBoost Model:

   The following outputs were obtained on a 5-fold cross-validation technique.

Splits	Training		Testing	
Model	**Scores**	**Fluctuations**	**Score**	**Fluctuations**
Precision	0.93727	0.00787	0.67335	0.08743
Recall	0.57234	0.00982	0.24839	0.059716
F1	0.71064	0.00775	0.36006	0.06965
AUC	0.99810	0.00007	0.98613	0.00305

XGBoost model turned out to have the best score. However, it is overfitting. Therefore, the model was tuned according to the roc score. The result reduced overfitting due to changed parameters but it didn't lead to improvement in ROC scores. The Linear Discriminant Analysis(LDA) model doesn't overfit and also provides a score consistently with very minimal fluctuations.

3.b. Business Impact

To get the prediction of the model, in terms of business perspective, the xgboost model has a higher prediction ability in terms of precisely predicting than the LDA model. LDA model has perfect recall. This means that the LDA model will not miss players who will be drafted for the prediction. But, it also predicts other members who will not be drafted for the upcoming season.

This makes spectators and players feel sad as the players might not be drafted. In this case, the XGboost model can be used as it makes predictions correctly for 75% of the time. But it slightly misses members who will be drafted next season. However, scout team who want to look after players who get drafted can use the LDA model as they won't miss players who will be drafted.

In terms of Kaggle prediction, both the LDA model and Xgboost model are very useful as it has a robust prediction for ROC score and is higher with very little fluctuation.

3.c. Encountered Issues

There were some issues while converting the team feature in the data set. The train data didn't have information or unseen values presented in the test dataset which raised an issue while converting to the labels using a label encoder. Using train data for encoding, it didn't have sufficient information about new teams and couldn't encode. While training it was encoded but it didn't show any significant changes in the model. Hence, it was dropped.

4. FUTURE EXPERIMENT	
4.a. Key Learning	Feature engineering and handling missing values are very important and we can see in the observation 'pick' value had a significant increase in the correlation strength with the target variable as it was differently imputed and feature engineering was applied. Hyperparameter tuning doesn't necessarily improve the model performance. Feature selection and reducing model complexity improved the model score with improved training time. Also, it made the model less dependent on other features.
4.b. Suggestions / Recommendations	For this competition, most of the feature engineering techniques, feature selection, and model selection were used to follow the complete machine learning pipeline. The ROC score of 0.99922 was achieved. However, more techniques like KNN imputer for imputation or predicting missing values with other features would have made the imputation more sensible to create more real values and predictions.