EXPERIMENT REPORT - 3

Student Name	Ronik Karki
Project Name	Assignment 1 week 3
Date	01/09/2023
Deliverables	Notebook Name: karki_ronik-24886412-week3-xgb.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. 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. 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	Based on the outcome of the previous experiment (i.e. experiment week 2), the hypothesis of this experiment is that adding feature engineering techniques like one hot encoding might improve the model scores. Furthermore, testing on other models might also improve the score. Imputing the mean value for missing data decreased the scores previously. Other models might perform well if we impute missing data as 0.

1.c. Experiment Objective	The expectation from this experiment is that using one hot encoder will increase the number of features and might increase the chances of overfitting. Using other models than the Logistic Regression model or Gradient boosting model might achieve better scores than the previous experiment results.

---

2. EXPERIMENT DETAILS	
2.a. Data Preparation	For the data preparation part, most of the cleanings were imported from loading the cleaned data file from the previous experiment. Instead of imputing with the mean values, all missing data were imputed by 0. The team column was removed as there were unseen labels in the test set.
2.b. Feature Engineering	For this experiment, two types of feature engineering techniques were used: 1. Data imputation: a. All numerical features that had missing values were replaced with 0 2. One hot encoding: a. All of the remaining categorical features were converted to integers by applying one hot encoding technique.
2.c. Modelling	Feature selection techniques were used for selecting the highly correlated and significant features. Chi2 test for the feature that had both input and output as a categorical feature. If the p-value was less than or equal to 0.05, then the feature was selected. 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, the gradient boosting model, and the extreme gradient boosting model (XGBOOST) to observe the difference between models and their performances.

---

3. EXPERIMENT RESULTS	
3.a. Technical Performance	Three models (i.e. logistic regression model, gradient boosting model, and xgboost) were run for this experiment. Stratified k folds were used to measure scores across different folds and their fluctuation.

1. Logistic Regression Model:

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

Splits	**Training**		**Testing**	
Model	**Scores**	**Fluctuations**	**Score**	**Fluctuations**
Precision	0.65626	0.0090	0.65084	0.08516
Recall	0.48693	0.01763	0.47211	0.03850
F1	0.55894	0.01370	0.54267	0.02244
AUC	0.99536	0.0001	0.99505	0.00087

2. Gradient Boosting Model:

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

Splits	**Training**		**Testing**	
Model	**Scores**	**Fluctuations**	**Score**	**Fluctuations**
Precision	0.91595	0.00549	0.78323	0.08602
Recall	0.84202	0.01437	0.71264	0.05073
F1	0.87736	0.00745	0.74474	0.05758
AUC	0.99945	0.00004	0.99815	0.0005

3. XGBOOST Model:

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

Splits	**Training**		**Testing**	
Model	**Scores**	**Fluctuations**	**Score**	**Fluctuations**
Precision	1.0	0.0	0.75694	0.07813
Recall	1.0	0.0	0.70887	0.05751
F1	1.0	0.0	0.72966	0.04963
AUC	1.0	0.0	0.99799	0.00045

From the above results, we can see that the XGBOOST model outperformed both the

	models. Even though the scores are higher in XGBOOST, the model seems to be overfitting in terms of precision and recall. However, the score remains constant across 10 different folds and has fewer fluctuations in its performance. In terms of ROC score, the model doesn't seem to overfit and hence it can be used for the Kaggle competition to achieve a score near 0.997. On the other side, the logistic regression model doesn't overfit and handles the data very well.
3.b. Business Impact	With a precision score of 75% and recall of 71%, the model is decent enough for the business to get a highly accurate model for player prediction. The fluctuations around the 10 folds are less. This model has a very good ability to distinguish between the positive and negative classes. However, the model is still learning some noise and has overfitted and can be improved before deployment.
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 one-hot encoder. Using train data for encoding, it didn't have sufficient information about new teams and couldn't encode. It can be focused on the last experiment to overcome the issue.

---

4. FUTURE EXPERIMENT	
4.a. Key Learning	One hot encoding feature increased the complexity of the model but it didn't improve the model performance. Data imputation plays a crucial role as imputing with 0 instead of mean values turned out to be very significant for a feature like 'pick' as we have many values missing. XGBoost model performed better in terms of ROC score but it is still overfitting in terms of precision and recall. From this, we can learn that even if the model isn't good enough for the prediction, it is still better at distinguishing between the classes. It might be because of the highly imbalanced dataset.
4.b. Suggestions / Recommendations	The XGBoost model improved the ROC score to 0.99883 but it is still overfitting. There are many suggestions for the upcoming experiment. Therefore, the following efforts can be applied in an attempt to reduce overfitting and improve the ROC score to achieve the best among other competitors for the upcoming final experiment: 1. Impute missing data with mean, mode, or different techniques according to features and look after the correlations to select the best informative imputations. 2. Hyperparameter tuning best two models by focusing on improving ROC score and reducing overfit. 3. Apply imbalanced data handling techniques like oversampling, undersampling, and combined sampling techniques. 4. Adding features that have very high importance scores to the model for reducing higher complexity or using different feature selection techniques.