EXPERIMENT REPORT

Student Name	Mark Brackenrig (12964298)
Project Name	Assignment 1C
Date	21/02/2021
Deliverables	notebooks/brackenrig_mark-12964298-week3_SVM.ipynb models/brackenrig_mark_12964298_week3_SVM_voting.sav github: https://github.com/roger-yu-ds/assignment_1

1. EXPERIMENT BACKGROUND	
Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.	
1.a. Business Objective	The goal of this project is to predict whether an NBA 'rookie' will have a career that spans at least 5 years. A rookie is an NBA player who is in the first year of their career. This model has numerous applications – for example, a collectibles investor could use this prediction to inform purchases of 'rookie' basketball cards while they are still cheap.
1.b. Hypothesis	The previous two experiments highlighted that using an ensemble model of two different classifiers and using SMOTE synthetic oversampling improved results. Based on these experiments, and experiments conducted by other team members, incorporating a new classifier into the voting classifier. Since there is currently a model based on decision trees (random forest) and a logistic regression model, a model that constructs predictions without requiring linearity and that is not based on decision trees could improve the model. This experiment will look at including a Support Vector Machine classifier as one of the underlying models as it performed considerably well in comparison to the other base classifiers in the voting classifier.

1.c. Experiment Objective	The objective is to test whether the inclusion of an SVM model within my ensemble model will improve upon previous results. I am aiming to test whether the inclusion of a non-linear model that is not based on decision trees will be able to identify new patterns.

2. EXPERIMENT DETAILS

Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.

2.a. Data Preparation	A shared function `download_data` was used to retrieve data from the Kaggle API, unzips the files and saves them into the raw data directory. This data contains two CSV files – 'train' and 'test'. The training dataset is cleaned by removing the ID columns 'Id' and 'Id_old'. The test dataset is split into 'X_test' (without the id column) and 'test_id' – a pandas series of the Ids. Next, the training set is split into the independent and dependent variables using the shared function `separate_target`. After this the data is transformed using the StandardScaler method used previously for just the linear regression model. Lastly, the training data is then transformed using SMOTE (Synthetic Minority Oversampling Technique). SMOTE randomly selects one of the K nearest neighbours to synthetically generate a data point between the two data points in the feature space. SMOTE will create new data points until the classes are represented evenly in the sample. There was no notable data cleaning performed on the dataset as it was a generally clean dataset with no missing values, or obvious erroneous entries. I decided to revert back to using a validation set (despite also using cross-validation) in this experiment to ensure that I could easily compare results across different models.

2.b. Feature Engineering	The baseline model was used to conduct the test – which was a VotingClassifier model consisting of a linear regression model and a random forest model trained on the SMOTE dataset. The logistic regression model applied Principal Components Analysis to create orthogonal independent variables. This was used due to the high levels of collinearity identified in the dataset (see appendix for correlation matrix). No feature engineering was used for the random forest model or SVM.
2.c. Modelling	The model uses a voting classifier with a random forest classifier, an SVM and a logistic regression classifier as inputs. The voting classifier uses the 'soft' voting method which classifies the result based on the highest average probability of the underlying classifiers. The Hyperparameter tuning was focused on tuning the SVM model, keeping the same grid for the other models the same.

---

3. EXPERIMENT RESULTS

Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

3.a. Technical Performance	BASE MODEL AUC (Training): 0.723 AUC (Test – Kaggle): 0.712 (On raw training data) Precision (Negative Class / Positive Class): 0.3/0.9 Recall (Negative Class / Positive Class): 0.59/0.72 F1 (Negative Class / Positive Class): 0.4/0.80

	WITH SVM AUC (Training): 0.723 AUC (Test – Kaggle): 0.713 (On raw training data) Precision (Negative Class / Positive Class): 0.29/0.9 Recall (Negative Class / Positive Class): 0.60/0.71 F1 (Negative Class / Positive Class): 0.4/0.79
3.b. Business Impact	The model including the SVM base classifier performed marginally better on the training and test set on Kaggle. The problems with predicting the minority class were not improved by using the SVM model as part of the base classifier.
3.c. Encountered Issues	Unfortunately, since this model did not materially improve the results of the model, the previous issues of predicting the minority class still remain. With the addition of the SVM model, the training of the model because significantly more time consuming. Utilising calibration therefore became difficult as it would have extended the training time of the model.

---

4. FUTURE EXPERIMENT	
Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.	
4.a. Key Learning	The incorporation of the SVM model as part of the base classifier has only marginally improved results. As I attempt to extend the model, the model appears to be overfitting to a larger degree. One way around this is to attempt to remove outliers in the larger training dataset. Theoretically this would allow the model to learn more important patterns rather than learn patterns that are not generalisable to the test set. I previously hypothesised that there is a level of inherit randomness in the dataset.

	The entire class has performing models within <0.01 AUC of each other on the test set. At best, there is only marginal improvements that are likely to be made on this dataset without overfitting.
4.b. Suggestions / Recommendations	Attempting to remove outliers in the training set could improve results. Using an autoencoder, or another anomaly detection algorithm, on the test set and then removing stated outliers in the training set could improve the results of the model by removing unnecessary data points. Further to this, I have had success in combining other approaches that were differently from my own. Other teams may have taken very different approaches to my team, which if combined with our approach, could improve the final score.

Appendix

Correlation Matrix

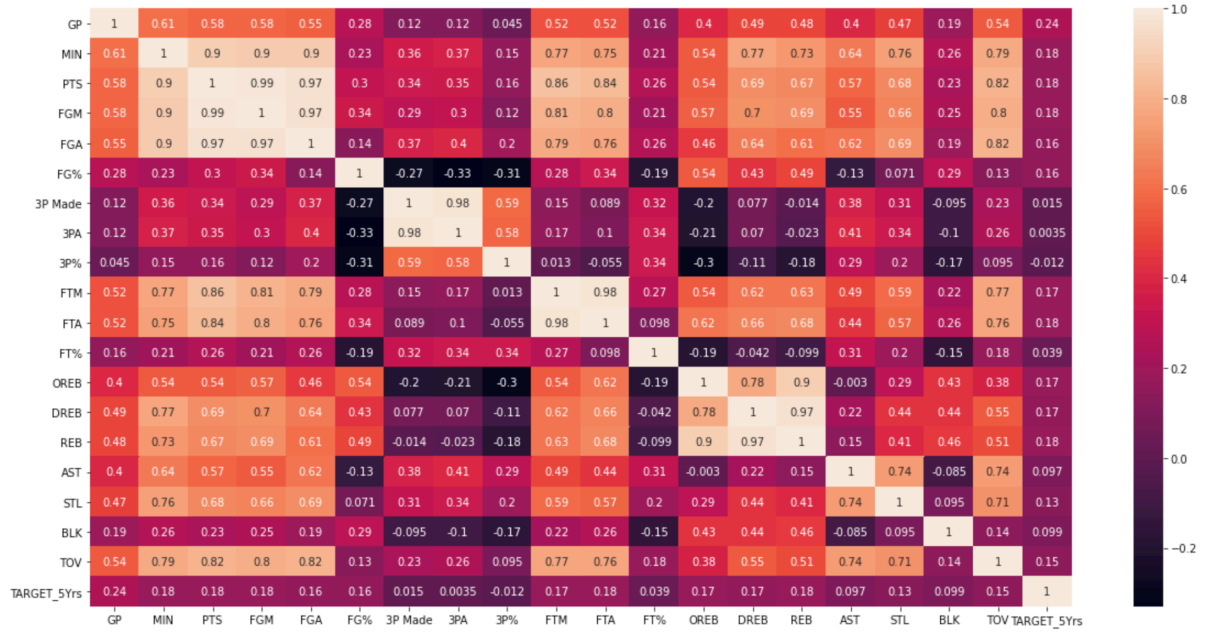

Figure 1: Correlation Matrix Heatmap