**Data**

A close up of a map

Description automatically generatedThe data set consists of all games played between the 2010-11 season up to the 2018-19 season. This included data for roughly 11,400 games played during this time. This data is publicly available from the NHL.

**Preprocessing**

To ensure the data is functional for the model, cleaning of the data is necessary to remove null values, and some normalization of data will provide proper analysis. This data set was processed previously, containing no null values; the features, however, did need to be normalized as their ranges were spread out. While preparing the data, it was clear that two features, power-play opportunities, and hits, were not significant to determine if a team will win the game. These features were then removed to help the overall capability of the model. After removal of these insignificant features, seven features remained to train the model.

**Linear Models**

For this data, a multiple linear regression model is required. This formula is as follows:

Y = β0 + β1 X1 + β2 X2 + … + βp Xp + ε

After preprocessing, the data was split into sets for training the model and outcome data for our tests. The models used during this cross-validation were Logistic Regression and a Linear SVC model. Obtaining all the scores, the accuracy rate at the predictive behavior of the model, the LR model was considered to be the most accurate of the three. A ROC curve was drawn to depict the accuracy of the model and to illustrate the overall sensitivity vs. specificity the model portrays. The area under the curve was calculated at roughly .86, proving the model is a reasonable representation of predicting team wins given certain features. A confusion matrix was then implemented to observe the test model. The test set was introduced to the model to test the accuracy, which portrayed the following results:

|  |  |
| --- | --- |
| True Positive: 2230 | False Positive: 605 |
| False Negative: 651 | True Negative: 2231 |

The model predicted the result roughly 80% accurately, proving the model can estimate a correct result 80% of the time.

**Non-Linear Models**

For my non-linear models, I used four separate non-linear classification models to analyze the best outcome. SVC, Decision Tree Classifier, Random Forest Classifier and Bagging Classifier were used to evaluate the data and compared to each other to find the best result. For these fits, the data was separated into two sets, a test set and a training set, in order to maintain accuracy with all four models. The training set took on 70% of the original data set in order to train the models with a sufficient amount of data points. The test set took on the remaining 30% of the original data set.

**Non-Linear Model Fitting**

A close up of text on a white background

Description automatically generatedIn order to fit the models to their best, a training method was used to fit the data using multiple given parameters and to determine which parameters gave the best results. For example, the Random Forest Classifier model was given four different parameters for the ‘n\_estimators’ parameter and two different parameters for the ‘criterion’ parameter. The method then created models for all different parameter options, eight separate models in this case, and cross-validated them individually three times. This gave this specific model 24 different models to compare and choose the best parameters for precision and recall. If these models were given different parameters, the separate models were then fit and checked for accuracy. The most accurate model was then used to produce a confusion matrix and a ROC curve to analyze the best model of the four different models trained.

**Non-Linear Model Results**

First, I started with an SVC model to analyze. The SVC model was allowed multiple types of parameters, so the optimization process resulted in fitting 72 different models three times. This allowed for the fine-tuning of the SVC. The optimization produced the same parameters for both precision and recall, and a 77.79% accuracy was calculated. The confusion matrix is as follows:

|  |  |
| --- | --- |
| True Positive: 2680 | False Positive: 733 |
| False Negative: 791 | True Negative: 2657 |

A screenshot of a cell phone

Description automatically generatedA ROC curve was fitted using this model, producing an area under the curve of roughly 0.86. Next, I looked into the Decision Tree Classifier model. This was put through the same optimization process and resulted in separate sets of parameters for precision and recall. The two models were then fit and compared by their accuracy and AUC of the ROC. In this scenario, the recall optimization model had a higher accuracy of roughly 0.01, with roughly 0.74 accuracy. A confusion matrix was produced as follows:

|  |  |
| --- | --- |
| True Positive: 2657 | False Positive: 756 |
| False Negative: 1020 | True Negative: 2428 |

A screenshot of a cell phone

Description automatically generatedThe ROC was then created for both models and compared, proving once again the recall model was more accurate and represented a better model with an AUC of roughly 0.81. Even though the recall model was more accurate in this scenario, it still does not stand up against the SVC model. Next, the Random Forest Classifier was analyzed. The same optimization process was completed resulting in the same parameters for both precision and recall. The confusion matrix and ROC are as follows:

|  |  |
| --- | --- |
| True Positive: 2614 | False Positive: 799 |
| False Negative: 813 | True Negative: 2635 |

A screenshot of a cell phone

Description automatically generated

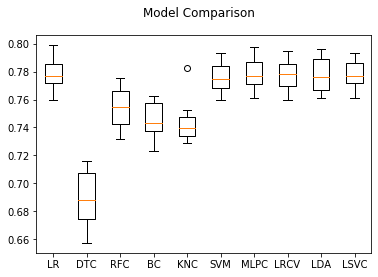
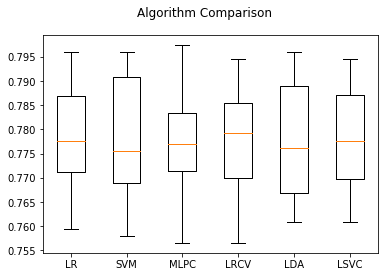
This accuracy was calculated at roughly 0.77, with an AUC of 0.85. This model is close in comparison with the SVC model; however, the SVC model still stands as the better. Lastly, the Bagging Classifier was put to the test, coming up with differing parameters for precision and recall. The accuracy score was close for both models; however, the precision model edged out the recall model by roughly 0.002 at 0.751. Considering the model’s accuracy was close, both ROC models were used to compare the two showing another close fight. The precision model edged out the recall model by 0.003 at 0.837. The confusion matrix for the recall model was then produced and resulted in the following:

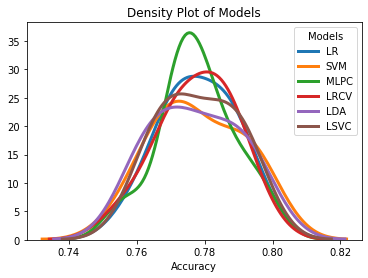
|  |  |
| --- | --- |
| True Positive: 2596 | False Positive: 817 |
| False Negative: 871 | True Negative: 2577 |

Given all four models, the data tends to produce similar models, but the SVC model stands out with an accuracy of 77.79% and an AUC of 0.86.

**Model Comparisons**

|  |  |  |
| --- | --- | --- |
| Model | Accuracy Mean | Standard Deviation |
| LR | 0.778311 | 0.010344 |
| SVM | 0.778166 | 0.012244 |
| MLCP | 0.777728 | 0.010759 |
| LRCV | 0.777728 | 0.010557 |
| LDA | 0.777436 | 0.012019 |
| LSVC | 0.778165 | 0.010813 |

In order to compare the models, I took a long approach to make sure our models used the same training sets and had a proper evaluation. To start, I fit all the models included in the initial evaluations with some additional classification models to compare the non-optimized models. This step was to determine which models fit the data best and which were appropriate to optimize. A box plot was created with cross-validation of ten folds and a focus on the accuracy of each model. As seen from the results, four models that we used produced results significantly worse than others, so the Decision Tree Classifier, Random Forest Classifier, Bagging Classifier, and K-Nearest-Neighbor models were dropped. This left three linear models, Logistic Regression, Linear Discriminant Analysis, and Linear SVC, a non-linear model, SVC, and a neural net model, MLP Classifier, to optimize their fit. From here, an optimization was done similar to the non-linear models, using a grid search with multiple parameters to determine the best parameters for each model. This cross-validation resulted in the best parameters for precision, recall, and f-1 score. If the best parameters for each score were different, each model was fit and compared using their accuracy. The highest accuracy was then used to fit the model. A final box plot was created with the optimized models to compare their results. The box plot shows

that the models are fairly even, with all accuracy scores being within roughly 0.001 of each other. A density plot of the models was created to compare the models. These two plots show the Multi-layer Perceptron classifier and the Logistic Regression CV classifier show to less deviation from the mean and prove to be able to predict the outcome of future games more accurately. With these results, it is apparent, even with the similar accuracy scores, that the Multi-layer Perceptron classifier and the Logistic Regression CV classifier are the best models to use moving forward.