We chose a number of supervised machine learning models to predict the number of field goals made. We ran, evaluated, and compared the performance of the following models:

* Decision Tree
* Logistic Regression
* GBM
* GBM with PCA

Before running the models, we converted the predictors and response variable to their correct data type. For example, numeric factors such as SHOT\_DIST were explicitly converted into numeric, while categorical factors such as FGM were explicitly converted into categorical factors.

Because the binary response variables of, 0 and 1, showcased a relatively balanced dataset, with the split being roughly 55%/45%, we did not need to undersample or oversample the data.

We also split the dataset into a training, and testing set. The split was set at 70% training, and 30% training, ensuring an equal

**Decision Tree**

The Decision tree algorithm is an algorithm that uses a tree-like data structure to make either predictions for regression, or classification problems. Given the business problem and context, a categorical variable decision tree was chosen as we wanted to classify, given the available data, whether or not an attempted shot made became a FGM (Field Goal Made); in this particular situation, there would be two categories for the response variable: either 0, denoting an attempted shot that missed, or 1, denoting an attempted shot that resulted in a field goal.

A decision tree is suitable supervised machine learning algorithm because it is fairly easy to explain and visualize. For example, the following figure below is the generated decision tree diagram. Shot Distance, titled as SHOT\_DIST, as well as the distance to the closest defender, titled as CLOSE\_DEF\_DIST, were the two most important variables in the decision tree, and of which the decisions are based upon.

[INSERT decision\_tree/regression\_tree.png]

The ROC curve, and AUC of the decision tree algorithm showed similar results to the other models, as seen in the figure below. Nevertheless, it was decided not to use the decision tree algorithm as the model only showed 3 raw probabilities given the different permutations of SHOT\_DIST and CLOSE\_DEF\_DIST. This would not have looked good on the Shiny app, as we wanted to show more probabilities on a more granular level, as there were other predictor variables that were not being used in the final model.

[INSERT decision\_tree/ROC.png]

**Logistic Regression**

Logistic Regression is a parametric model used for binary classification, and it is based off the logit (logistic) function, hence the name. There are a few assumptions that were made with logistic regression:

1. There is a large enough dataset to make accurate predictions
2. Observations are independent of one another
3. Response variable is binary (0 or 1).
4. Predictor variables are related to logit function
5. Minimal multicollinearity among the predictor variables

All assumptions except (4) and (5) are true, and due to the lack of time on our part, we were unable to ascertain whether the last two assumptions were, indeed, true.

The results for the logistic regression showed similar results to other models. Listed below is the ROC curve/AUC results of the logistic regression model.

[INSERT glm/glm\_auc\_roc.png]

In terms of other metrics like accuracy, balanced accuracy, sensitivity, and specificity, the logistic regression model also showed similar results to the other models. Unfortunately, logistic regression took the longest amount of time to train compared to all the other models, at 2.5 hours. Listed below are the other metrics, as well as the runtime of the logistic regression model.

[INSERT glm/glm\_output.txt]

**Stochastic Gradient Boosting (GBM)**

Stochastic Gradient Boosting is a relatively complex supervised learning algorithm. The algorithm continuously iterates through several trees, at one at a time, so that it can boost the performance of its weakest learners.

When initially training the GBM model, we first decided to keep CLOSEST\_DEFENDER\_PLAYER\_ID, and player\_name, when training the GBM model, but the model itself was fairly big, and in terms of variable importance, both features were not that important unless if the players had a large enough data sample to draw meaningful conclusions from. For example, looking at the figure of output variable importance, one could see the top CLOSEST\_DEFENDER and player\_name names were highly regarded players from the 2015 season, or at least players who had a lot of playtime.

[INSERT gbm/output\_variable\_importance.txt]

Also, including the features CLOSEST\_DEFENDER\_PLYAER\_ID and player\_name made the model size unnecessarily big for little to no improvement in the prediction of FGM. In relation to the size of the models, it also took a lot longer to train the model as well, at about 4 hours instead of the usual 1 hour.

For hyper parameter optimization, we found that GBM produced the best results at with an interaction depth of 5, 40 trees, minimum of 10 observations in each node, and a shrinkage of 0.1. Using such hyper parameters offered a relatively fast training time of around 10 minutes.

Listed below is the ROC curve, and AUC value for GBM without CLOSEST\_DEFENDER\_PLAYER\_ID, and player\_name predictor variables. Also listed below is the console output of training the GBM model, showing some of the metrics from the test dataset. Overall, the GBM model did the best compared to the other models, although not by a huge margin. It had an accuracy of 62.01%, a sensitivity of 0.8661 which were the highest of all the models. Unfortunately, its specificity was at the lower end compared to other models at 0.3221.

[INSERT gbm/roc\_auc\_gbm.png]

[INSERT gbm/output\_without\_variable\_importance.txt]

**Stochastic Gradient Boosting (GBM) with PCA**

We also chose to perform GBM with the results generated from PCA. We used all 8 of the components for the GBM model. Listed below was the variable importance of each component:

[INSERT gbm\_pca/relative\_importance.jpg]

Listed below are the metrics of the GBM with PCA. Overall the GBM with PCA model was fairly competitive with the previous GBM model, as it scored 61.58% accuracy, and a 0.8035 sensitivity. Its specificity was a bit better than the previous GBM model at 0.3884. In terms of runtime both GBM models were fairly similar with a training duration of around 10 minutes.

[INSERT gbm\_pca/roc\_auc.jpg]

[INSERT gbm\_pca/gbm\_pca\_output.txt]

**Evaluation**

All models performed fairly similarly in terms of metrics, such as ROC, AUC, Accuracy, Balanced Accuracy, Sensitivity, and Specificity. In terms of overall performance, the models were not spectacular: they showed an accuracy, balanced accuracy, and AUC of around 0.60, sensitivity of around 70% to 80%, and a specificity of 30% to 40%. Nevertheless, there were some positives to be noted: although the model was only 10% better than chance, the sensitivity of all models was quite high at 70% to 80%. This meant that the model was fairly good at correctly predicting the percentage of actual positive cases (i.e FGM); on the flip side, all models had fairly poor specificity, meaning the models were not so good at correctly predicting the percentage of actual negative cases (i.e. shots that missed).