12/11/2024

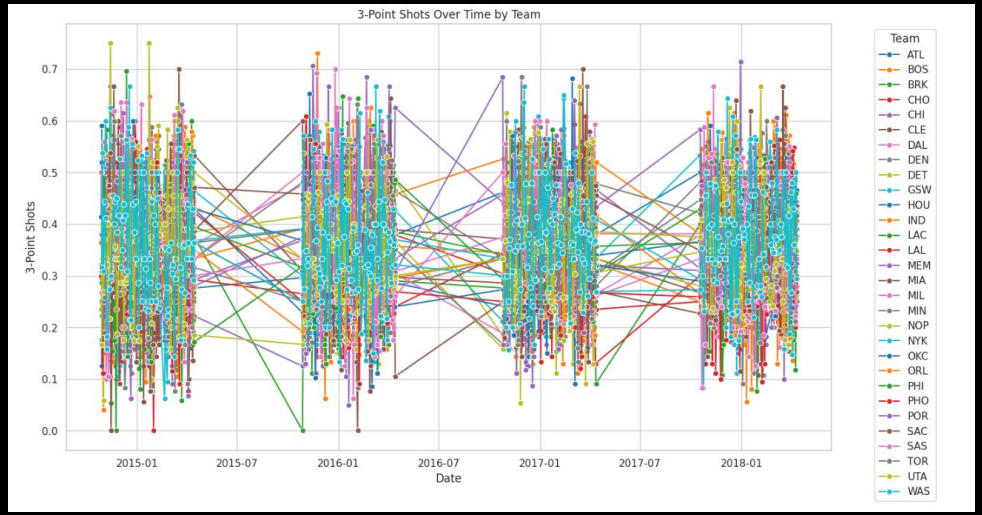
BYU

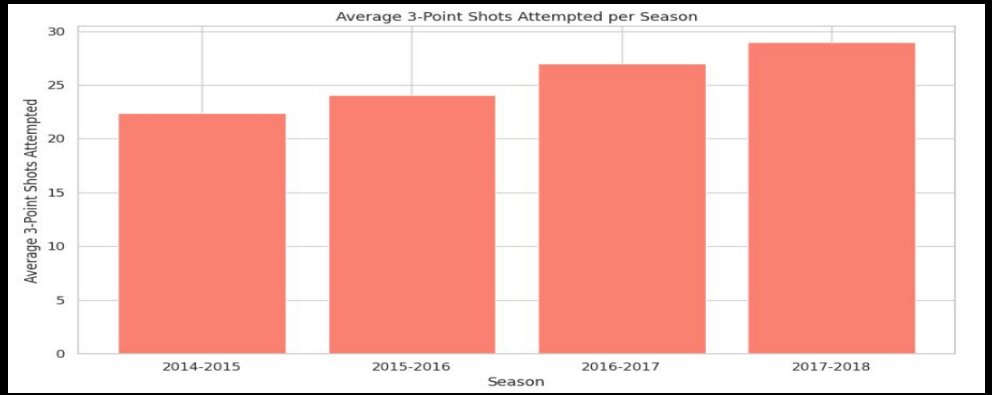
Deep Dive Into NBA Data

Ryan Corry

Introduction

- I love sports and I believe that statistics and sports were a dream marriage and it makes watching/following the sport so much more of an enjoyable experience for me.
- There are lots of trends that you can look at, and tons of different data points that you can use to make predictions
- The question that everyone wants to know, what is the best indicator of a winning basketball team? And how do we predict if a team won or lost?





IS THE NBA ALL ABOUT THE 3?

Logistic Regression

- The first model that I tried was a Logistic Regression Model
- It ran fine, but it was clear that this model had some errors built into it as it was not giving great results.

```
Accuracy: 1.00
Confusion Matrix:
[[1009    0]
 [   0  959]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1009
           1       1.00      1.00      1.00       959

    accuracy                           1.00      1968
   macro avg       1.00      1.00      1.00      1968
weighted avg       1.00      1.00      1.00      1968
```

Extremely Randomized Trees - Test Accuracy: 0.9162

Histogram Gradient Boosting - Test Accuracy: 0.9533

Gradient Boosting - Test Accuracy: 0.9395

AdaBoost - Test Accuracy: 0.9411

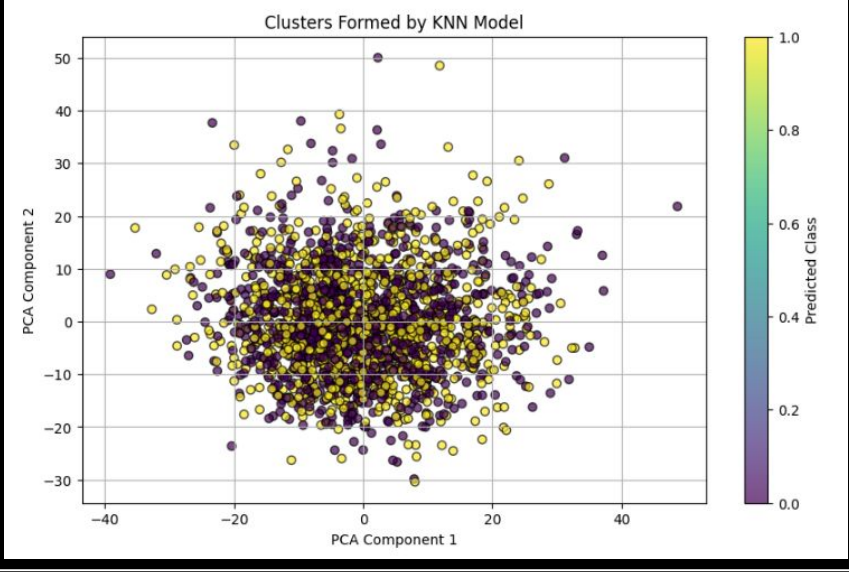
Support Vector Classifier Test Accuracy: 0.5467

KNN Model Accuracy: 0.84

Random Forest Model Accuracy: 0.91

Decision Tree Model Accuracy: 0.82

Finding A New Model



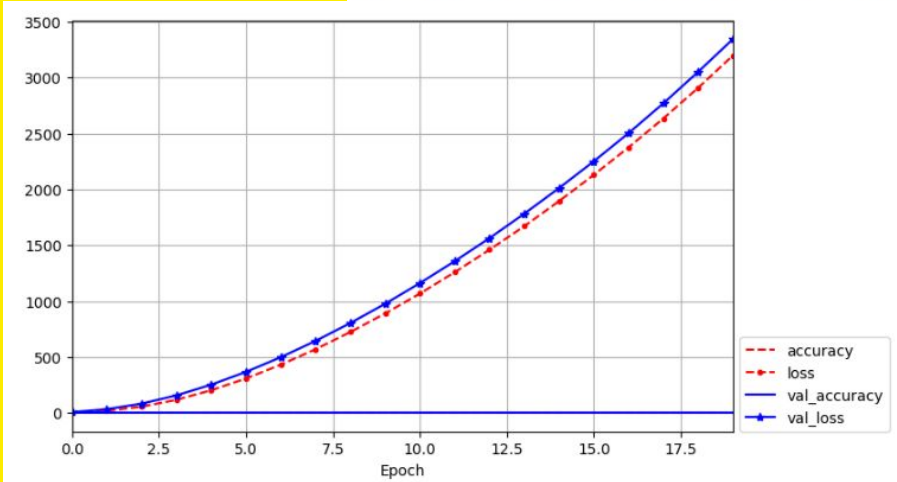
Deep Learning Model

- Once I tried multiple predictive models, I wanted to train a deep learning model to see if it would be able to accurately predict wins and losses
- I used keras to create multiple different models, the one I landed on was a 2 layer relu model with a sigmoid layer at the end.

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.InputLayer(input_shape=(32,)))
model.add(tf.keras.layers.Dense(500, activation="relu", kernel_regularizer=l2_reg,
                                kernel_initializer='he_normal'))
model.add(tf.keras.layers.Dense(200, activation="relu", kernel_regularizer=l2_reg,
                                kernel_initializer='he_normal'))
model.add(tf.keras.layers.Dense(1, activation="sigmoid"))
```

```
model.compile(loss="categorical_crossentropy",
              optimizer="adam",
              metrics=["accuracy"])
```

```
Test loss: 3343.5830078125
Test accuracy: 0.5127032399177551
```



Finding Anomalies

- Based on the low accuracy of my model, I wanted to test to see what anomalies existed within my data.
- I used a Gaussian Mixture model with 3 components to help find anomalies.

	Unnamed: 0
230	672
286	415
292	473
307	623
339	124
...	...
8910	55108
9177	76111
9203	20112
9528	17116
9778	21158

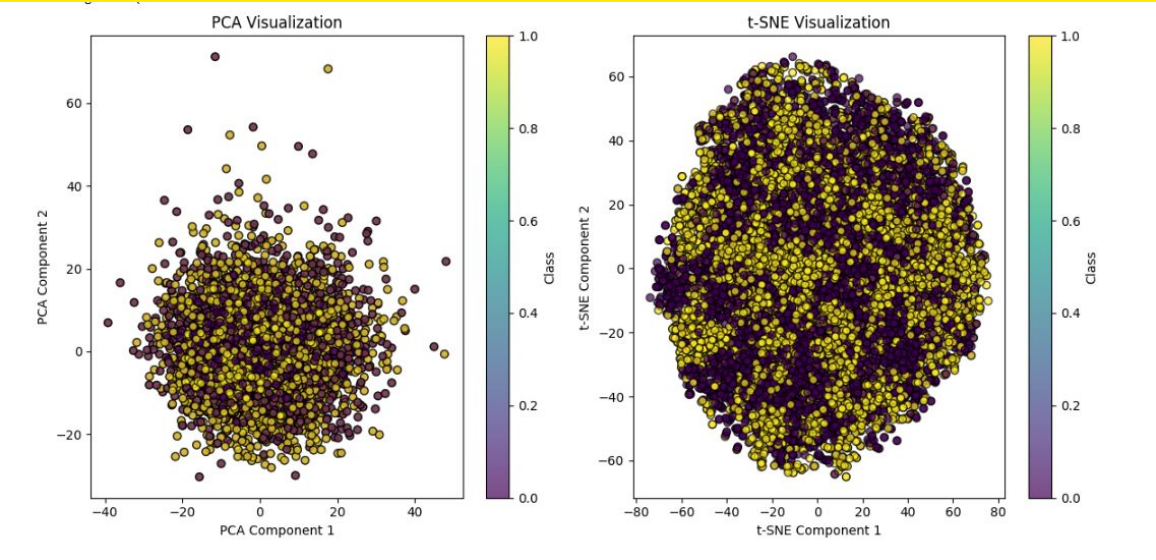
99 rows × 1 columns

```
densities = gmm.score_samples(X_scaled)

threshold = np.percentile(densities,1)
outliers = densities < threshold
NBA.iloc[outliers,0]
```

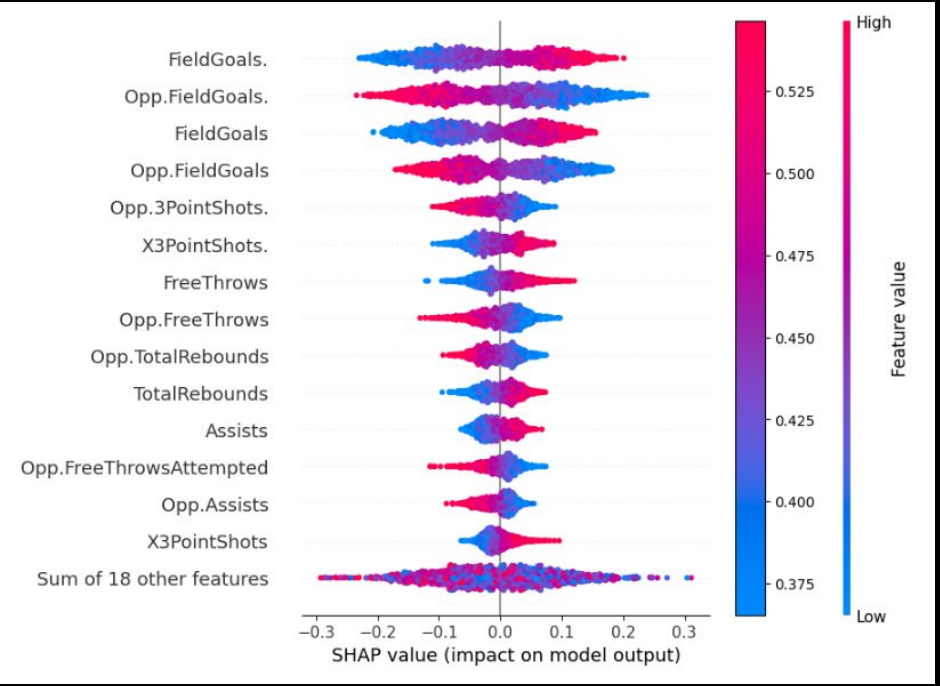
PCA and t-SNE

- In an effort to improve my model's accuracy, I wanted to try different dimensionality reduction techniques to see if they could help
- As you can see in the images, I did not have much luck trying to get the clusters to seperate

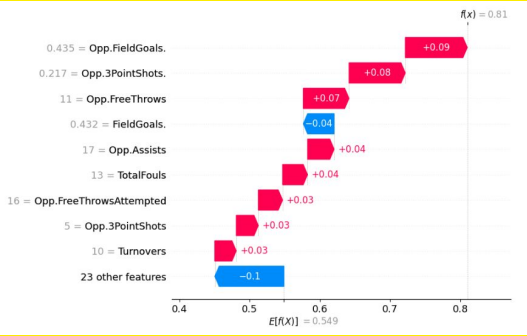


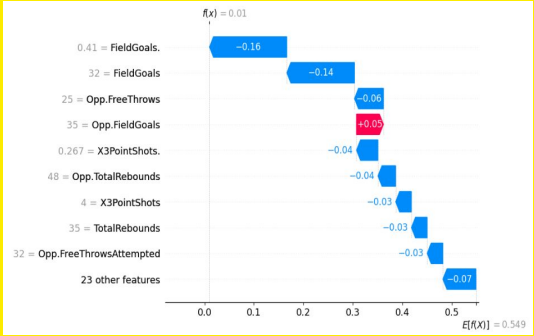
SHAP VALUES

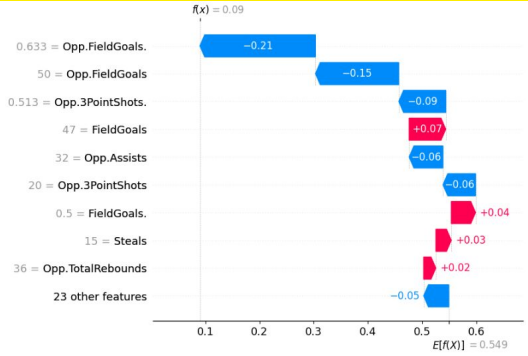
- Finally, I wanted to use SHAP values as a way to show which aspects of a basketball game most impacts the model's prediction of Wins and Losses.
- It was interesting to see that field goals was by far the most dominant statistic when it comes to the model's predictions.

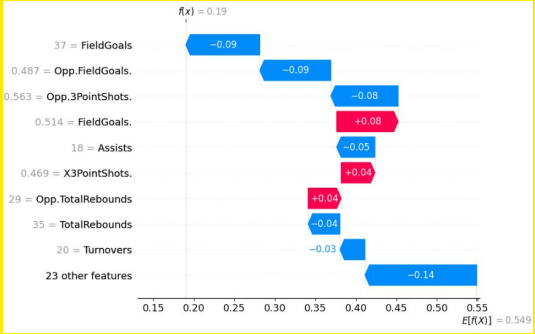


Waterfall Plots









Conclusion

- In conclusion, I am very happy with how my data came out.
- I learned a lot about how to work with sports data and came out of this project with a different way to view the game.
- I believe that the best way to enjoy sports is to understand how the game is played from a strategic level. And the best way to understand the best strategies of the game is to understand what best predicts wins and losses and how your team can best be positioned to win!