Fantasy Football Final Assignment Write Up

Because of the nature of the four different positions, I split the data into four different models: QB, TE, WR, and RB. The first thing I did was build ideas about what data to include and what not to include. For all four models, the 2021 data was all filtered out of the dataframe because they directly cause the points scored in 2021. I also removed all rookies that entered the league in 2021, as they had no data. To further understand the data, I created a correlation data, covariance matrix, and VIF table to see which variables correlated the most with each other and identify any multicollinearity within them.

For imputation, I used K-nearest neighbor imputation because a lot of the players who were either injured or were rookies in 2020 heavily influenced the dataset. I also removed outliers using the 1.5 IQR method. This process is shown in the code below.

```
#IMPUTING WITH KNN
# Specify the number of neighbors
k_neighbors = 2

# Create KNN imputer
knn_imputer = KNNImputer(n_neighbors=k_neighbors)

# Perform imputation
df_knn = knn_imputer.fit_transform(df)

# Convert the NumPy array back to a DataFrame
df = pd.DataFrame(df_knn, columns=df.columns)

# Function to identify outliers using IQR
def find_outliers_iqr(column):
    q1 = column.quantile(0.25)
    q3 = column.quantile(0.75)
    iqr = q3 - q1

    # Define lower and upper bounds for outliers
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr

    # Identify outliers
    outliers = (column < lower_bound) | (column > upper_bound)
    return outliers

# Identify outliers for each column
outliers_dict = {}
outliers_dependent_var = find_outliers_iqr(df[dependent_variable])
```

Now, I was able to separate the data by position and do a model on each.

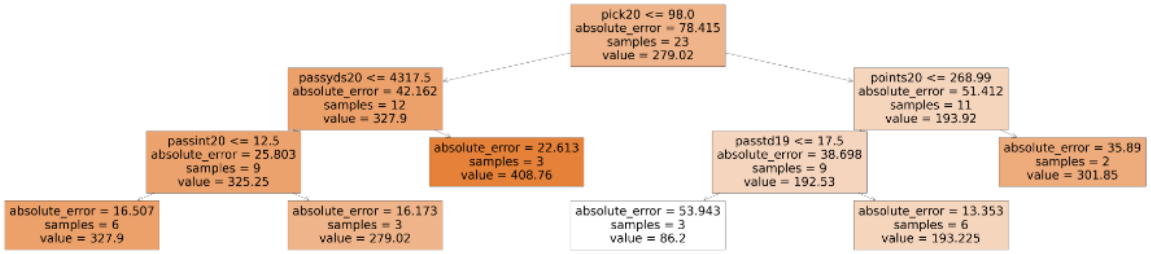
For each position, I tried three separate models: a decision tree, a Lasso regression with all the PCA variables passed in, and a model with handpicked PCA variables and other important variables.

QB Data

With QB's a substantial amount of their points comes from passing yards and rushing yards, while receptions are irrelevant, so I removed those corresponding columns from the QB data. From the correlation matrix and plots, I saw that the variables that most correlated with points21 was points20, salary20, pick20, the passing variables, and the rushing variables.

Decision Tree Model

For my decision tree model, I did grid search, which fine tunes all of the hyperparameters of the decision tree to optimize the error.




```
Best Parameters: {'ccp_alpha': 0.2, 'criterion': 'absolute_error', 'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split':
2}
Best Mean Squared Error: 5489.362198620701
Mean Squared Error on Training Data: 2418.9927597493083
Mean Squared Error on Test Data: 10587.013226266738
R-squared: -1.6436344408548553
```

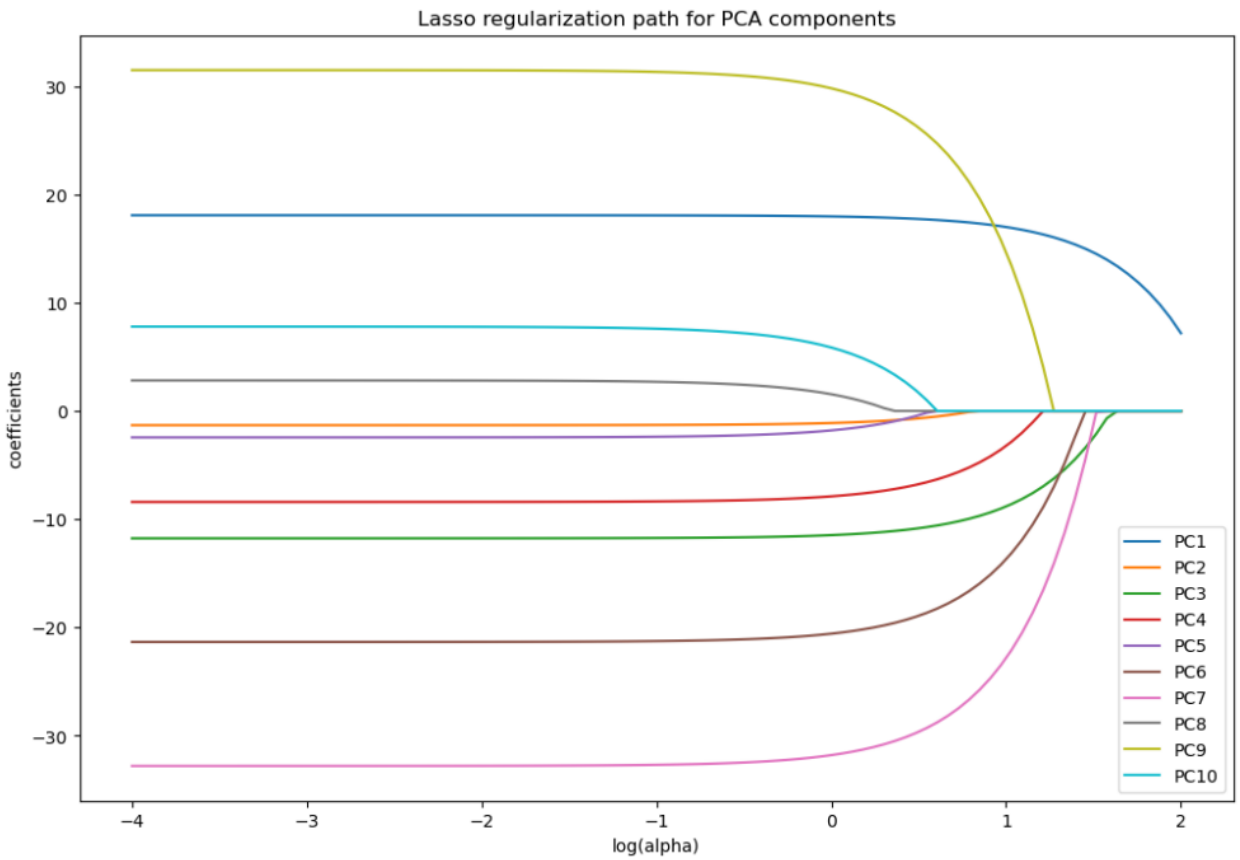
The variables used in this decision tree seem to make sense; the important variables are used to split the data. However, pick20 does not split the data with much impurity, which makes me wonder why it was chosen as the first split. Also, the mean squared error and the r-squared show that this model was incredibly inaccurate.

Lasso + LOO with PCA variables Regression Model

For the next model, I threw in all the variables and made it do a PCA, effectively reducing the number of independent variables from 30 to 10. Although I acknowledge the later latent variables had an eigenvalue of less than 1, meaning they explained less than 1 variable of variance, I chose 10 as the final number of dimensions because the threshold of explained variance I wanted to achieve was 90%, and as shown in the image below, the first 10 latent variables explained about 90% of variance.

```
Variance explained by each latent variable in PCA:  [0.33996621 0.18464473 0.12573501 0.07147708 0.05702594 0.04844739
 0.03715216 0.0290202  0.02200131 0.01916461]
   Explained Variance  Cumulative Explained Variance
0            9.546251                       9.546251
1            5.184824                      14.731075
2            3.530639                      18.261714
3            2.007077                      20.268791
4            1.601288                      21.870079
5            1.360403                      23.230482
6            1.043233                      24.273715
7            0.814887                      25.088602
8            0.617797                      25.706399
9            0.538142                      26.244541
```

I then passed the latent variables dataframe into the Lasso model and got a lasso regularization path for the PCA components.



This shows that PC1, PC3, PC6, and PC9 are the most important latent variables in the regression, in that order, because they require the highest penalty term in order for the coefficient of the variable to go to 0. This is illustrated in the resulting model:

```
Optimal Alpha: 6.2464
Mean Squared Error (Lasso + LOOCV): 7048.197322181384
R-Squared: 0.2038461453890399
Equation: y = 276.7405 + 17.6079 * PC1 + -0.0000 * PC2 + -10.2192 * PC3 + -4.6855 * PC4 + -0.0000 * PC5 + -13.8783 * PC6 + -29.
1070 * PC7 + 0.0000 * PC8 + 22.2538 * PC9 + 0.0000 * PC10
```

In this model, I did Leave-One-Out cross validation because there were not enough data points to do a simple train-test split. As shown in the model, the latent variables that “died out” first in the lasso regularization path graph, such as PC2, PC5, PC8, and PC10, had coefficients very close to 0, meaning they were not very important. This model isn’t really believable because some of the variables that happened to be least important according to the lasso regularization path explained a large amount of the variance, such as PC2. Also, this model is not interpretable - if I told someone I could model fantasy football points with ten latent variables they won’t understand what really

influences fantasy football points. Finally, the r-squared and mean squared error show that the latent variables aren't able to predict with very much accuracy.

PCA on Rushing and Passing Variables Model

For the next model, I did two separate PCAs: one that lumped the six rushing variables together and one for the passing variables. A QB's passer rating should be a good indicator of his passing yards, passing touchdowns and other passing metrics. Same goes for the rushing rating. Thus, it makes sense to reduces these variables into latent spaces.
This is the passing PCA variables:

```
passing_columns = ['passyds20', 'passtd20', 'passint20', 'passyds19', 'passtd19', 'passint19']
```

```
Variance explained by each latent variable in PCA:  [0.39652022 0.31703741 0.1533203 ]
   Explained Variance  Cumulative Explained Variance
0            2.474286                      2.474286
1            1.978313                      4.452600
2            0.956719                      5.409318
```

I decided to keep the first three dimensions because together, they explained 85% of the variance, which is pretty good. I did the same for the rushing variables:

```
rushing_columns = ['rushyds20', 'rushtd20', 'rush1st20', 'rushyds19', 'rushtd19', 'rush1st19']
```

```
Variance explained by each latent variable in PCA:  [0.77693219 0.10829867]
   Explained Variance  Cumulative Explained Variance
0            4.848057                      4.848057
1            0.675784                      5.523841
```

This time, I only used the top two PCA variables because they explained to about 88% of the variance.

Next, I did a linear regression model on these PCA variables plus 'points20' and 'points19' because I thought they would help the model given how correlated they are.

```
handpicked_df = df[['passingPC1', 'passingPC2', 'passingPC3', 'rushingPC1', 'rushingPC2', 'points20', 'points19']]
# Split the data into training and testing sets
```

```
Training Mean Squared Error: 3264.637789297113
Training R-squared: 0.6736940847616122
Test Mean Squared Error: 14607.7650
Test R-Squared: -2.2659
Coefficients: [ 2.71375864e+01 -1.05167186e+02 -7.81644587e+01  3.44232935e+01
  4.27803031e+01  1.04316369e-01 -1.66798259e+00]
Intercept: 698.9861588334984
```

Clearly, this model was completely overfitting based on the massive difference between the training MSE and test MSE. Also, it isn't really believable because the according to the coefficients, some of the latent passing variables had a negative correlation and points19 had a negative correlation with the dependent variable. Also, the intercept is incredibly wrong as it states that if a player gets no passing yards, rushing yards, or points in 2020 or 2019, they automatically get 700 points, which is far beyond any player's actual fantasy points.

Lasso and Leave One Out Cross validation on these variables helped the model significantly, but it just stated that all of these latent variables are useless. Here are the results from that:

```
Optimal Alpha: 311.7297
Mean Squared Error (Lasso + LOOCV): 7655.9128
R-Squared: 0.1352
Equation: y = 85.6761 + 0.0000 * passingPC1 + -0.0000 * passingPC2 + -0.0000 * passingPC3 + -0.0000 * rushingPC1 + 0.0000 * rus
hingPC2 + 0.3806 * points20 + 0.3451 * points19
```

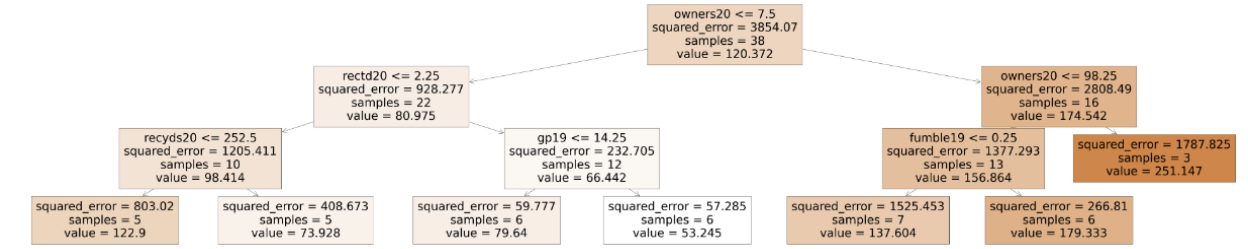
As shown, all of the coefficients for the latent variables are 0, while points20 and points19 seem to have the greatest affect. This makes sense, because a good proxy for how good many points they score in 2021 is how many points they scored in previous years. However, at this point, modeling points21 based on just the average of points20 and points19 would become a better model. At .13, the r-squared is still pretty bad as well, showing that a model with just points in 2020 and 2019 cannot accurately predict points in 2021.

For all of the other positions, I did much of the same thing.

TE Data

Contrary to QB's, tight ends have a much bigger emphasis on receptions and don't get passing or rushing yards, so I deleted those variables from the data.

Decision Tree Model

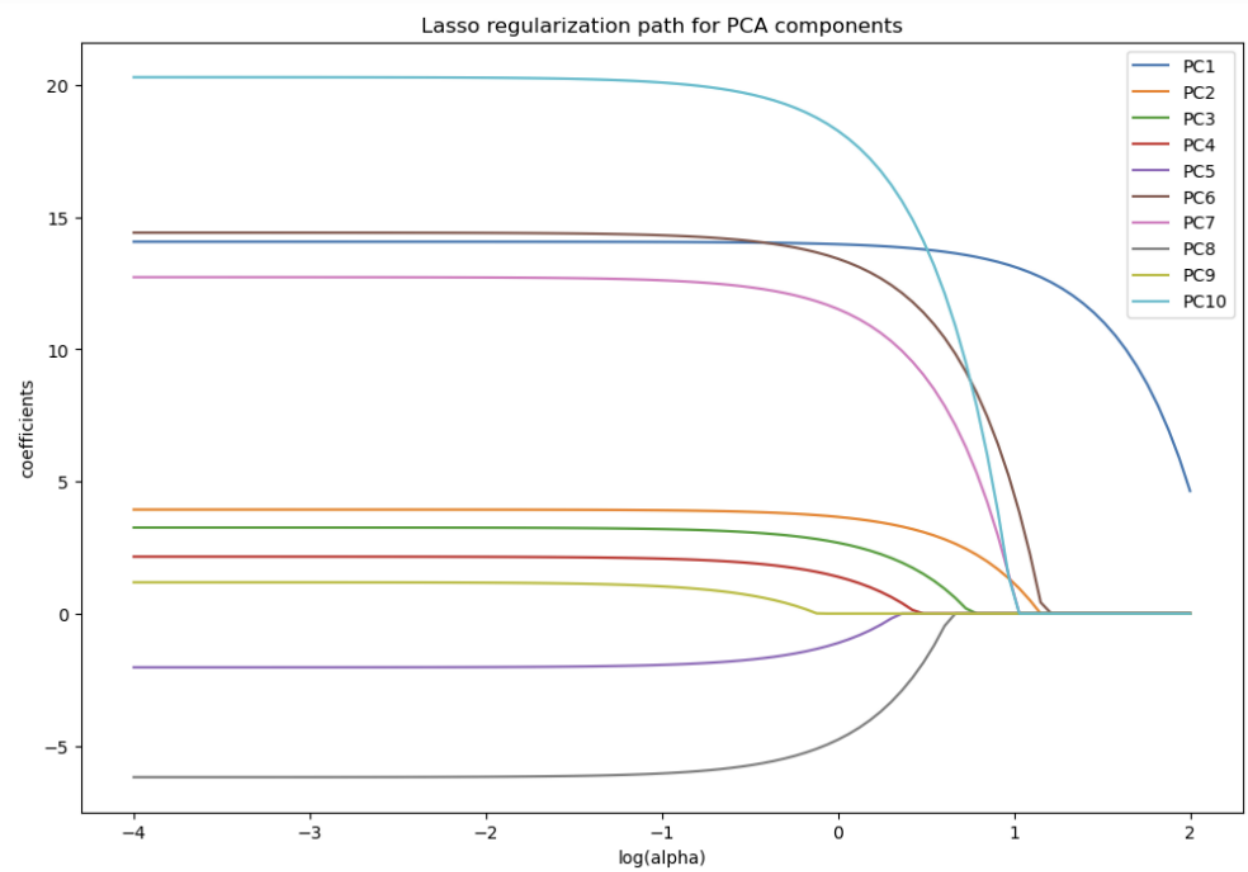



```
Best Parameters: {'ccp_alpha': 0.5, 'criterion': 'squared_error', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 1
0}
Best Mean Squared Error: 3124.157670486343
Mean Squared Error on Training Data: 838.5978506550096
Mean Squared Error on Test Data: 2331.2716924326223
R-squared: 0.3992454791175589
```

For tightends, I expected receptions to have a huge influence on how many fantasy points they earn, which is why I was surprised owners20 was the first split. Nevertheless, the decision tree had a solid r-squared of .4, although it seemed to be heavily overfitting as the test MSE was so much greater than the training MSE.

Lasso + LOO with PCA variables Regression Model

Similar to the QB data, I threw all of the variables into a PCA and chose the top 10 latent variables.



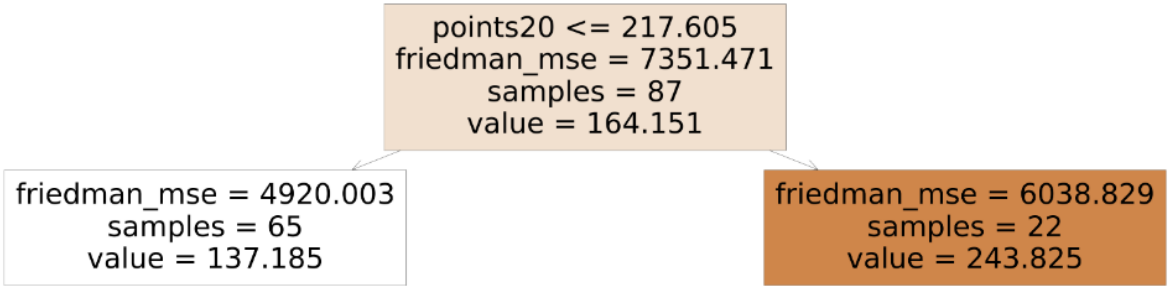

```
Optimal Alpha: 5.6182
Mean Squared Error (Lasso + LOOCV): 1971.6572870381883
R-Squared: 0.489059256550885
Equation: y = 120.6112 + 13.5187 * PC1 + 1.6795 * PC2 + 0.0000 * PC3 + 0.0000 * PC4 + -0.0000 * PC5 + 8.2819 * PC6 + 5.8855 * P
C7 + -0.0000 * PC8 + 0.0000 * PC9 + 8.5602 * PC10
```

This model seems to be pretty good - there is a .5 r-squared with only 5 latent variables that seem to actually matter. This may be because tight ends are more predictable than quarterbacks - they have less metrics to track while quarterbacks have so much variability because they're performance is also dependent on factors that are out of their control such as his O-line and how good his receivers are at running routes.

WR Data

Similar to tight ends, wide receivers should have much less variablility because they aren't as directly dependent on how well his offense is playing. They also don't do any passing or rushing, so I deleted those variables from the dataset.

Decision Tree Model

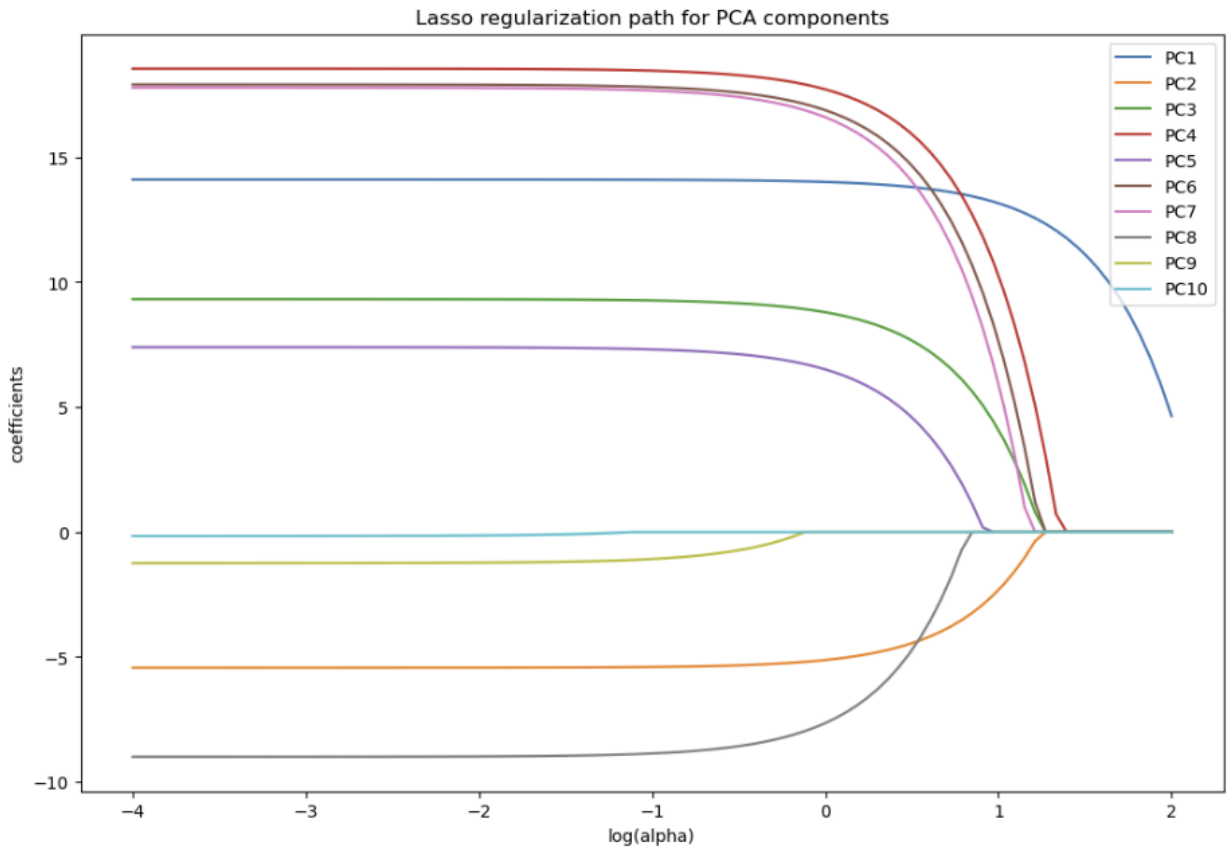



```
Best Parameters: {'ccp_alpha': 0.0, 'criterion': 'friedman_mse', 'max_depth': 1, 'min_samples_leaf': 2, 'min_samples_split': 5}
Best Mean Squared Error: 6202.7636727947065
Mean Squared Error on Training Data: 5054.385412864688
Mean Squared Error on Test Data: 3762.0931376497865
R-squared: 0.5372763855724187
```

Looking at the model, this is way underfit - we are only predicting points21 from one split: points20. However, I think this makes sense because the best proxy for points 21 is points20. Since wide receivers don't have as much variability, other factors should not matter as much. Although the r-squared is high, I don't think this is a good model because there should be other factors that come into play.

Lasso + LOO with PCA variables Regression Model

```
Variance explained by each latent variable in PCA:  [0.45963377 0.1397351  0.08309271 0.05253903 0.04894552 0.04238047
 0.03668679 0.03220114 0.0266922  0.02044099]
   Explained Variance  Cumulative Explained Variance
0           10.681697                     10.681697
1            3.247386                     13.929083
2            1.931040                     15.860123
3            1.220985                     17.081108
4            1.137473                     18.218581
5            0.984904                     19.203486
6            0.852586                     20.056072
7            0.748341                     20.804413
8            0.620316                     21.424728
9            0.475040                     21.899768
```



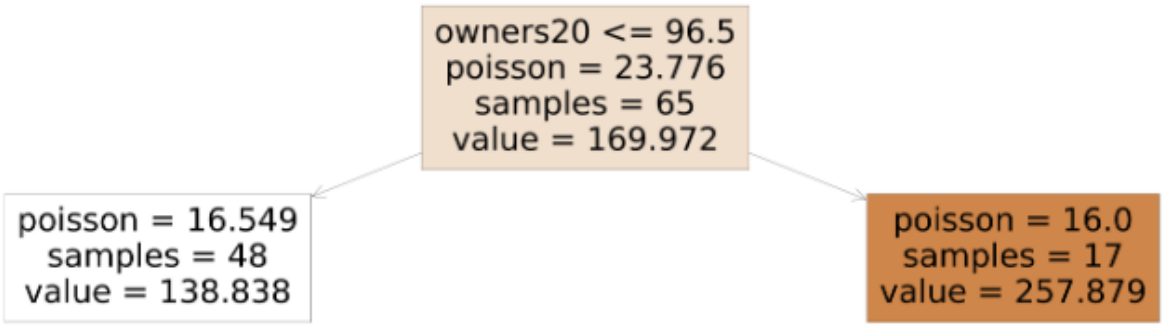

```
Optimal Alpha: 1.8383
Mean Squared Error (Lasso + LOOCV): 5020.229402085309
R-Squared: 0.3285161311710262
Equation: y = 166.8778 + 13.7571 * PC1 + -4.9885 * PC2 + 8.2240 * PC3 + 16.9319 * PC4 + 5.5394 * PC5 + 15.9937 * PC6 + 15.5230
* PC7 + -6.8263 * PC8 + -0.0000 * PC9 + 0.0000 * PC10
```

Similar to the other models, I chose 10 PCA variables because it explained about 90% of the variance. This model has a relatively low optimal alpha penalty term, which may mean that the model is pretty well-behaved and does not overfit, even when adding this many latent spaces. This is not a bad model with an r-squared of 0.33, but I think it is hard to tell a story when eight latent spaces are in the equation.

RB Data

Running backs had the biggest emphasis on rushing yards and receptions, so I deleted the passing variable. Unfortunately, like quarterbacks, running backs are also heavily influence by how well their O-line does, as they have to create gaps for running backs to run. This introduces more variance into the dataset, which lowers my expectations for how well the model is.

Decision Tree Model

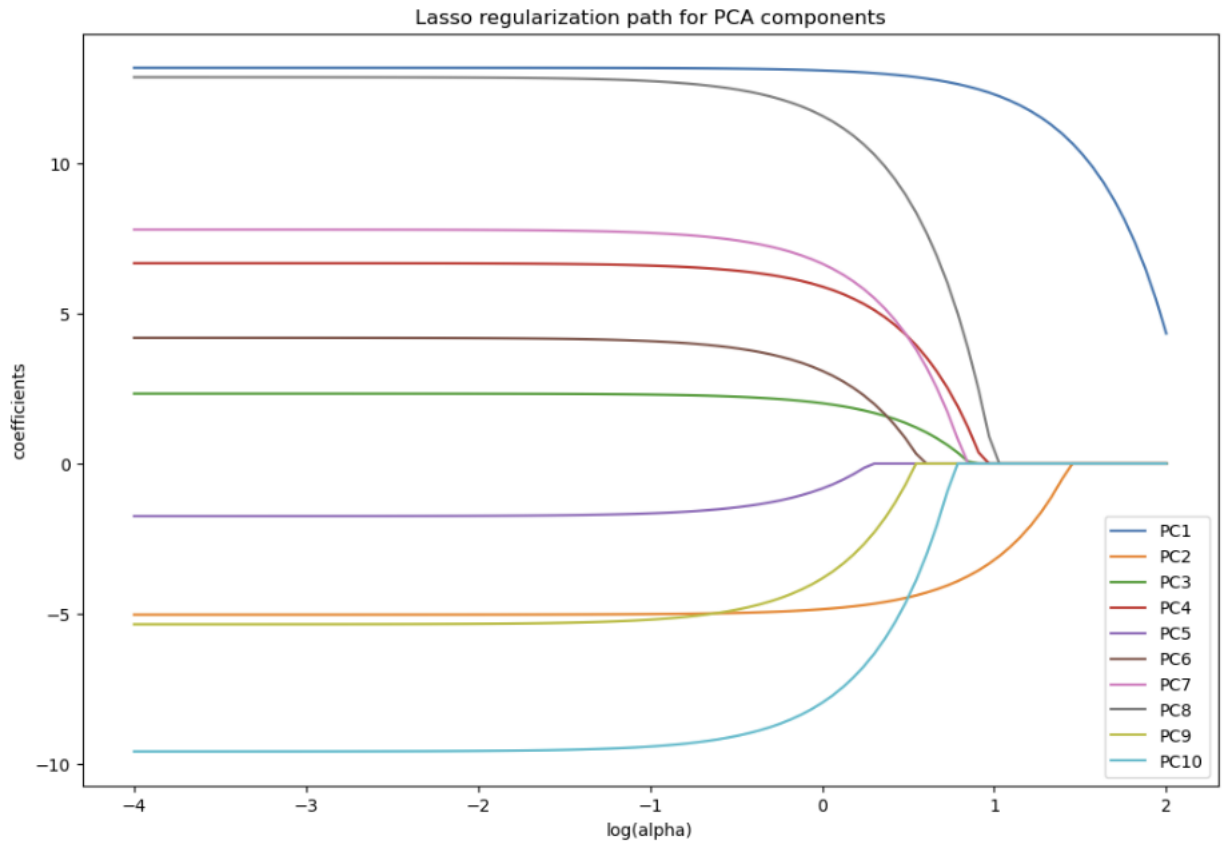



```
Best Parameters: {'ccp_alpha': 2.0, 'criterion': 'poisson', 'max_depth': 2, 'min_samples_leaf': 2, 'min_samples_split': 5}
Best Mean Squared Error: 8876.600432693967
Mean Squared Error on Training Data: 5658.644046387741
Mean Squared Error on Test Data: 8245.351665229555
R-squared: -1.9423641334639825
```

Interestingly, the running back's only split on the decision tree model is owners20, which is a solid proxy for how well a running back does overall. Given the aforementioned variance that running backs have, it makes sense that the model only has one split; it underfits so that the accuracy isn't as affected by the variance. However, the MSE shows that this model doesn't predict well at all.

Lasso + LOO with PCA variables Regression Model

```
Variance explained by each latent variable in PCA:  [0.38979604 0.18937782 0.10755533 0.04440916 0.03753015 0.03139189
 0.03031762 0.02688043 0.02259264 0.02125765]
   Explained Variance  Cumulative Explained Variance
0           11.461086                     11.461086
1            5.568234                     17.029320
2            3.162425                     20.191746
3            1.305753                     21.497498
4            1.103491                     22.600989
5            0.923009                     23.523998
6            0.891422                     24.415420
7            0.790359                     25.205779
8            0.664286                     25.870065
9            0.625034                     26.495099
```




```
Optimal Alpha: 13.8940
Mean Squared Error (Lasso + LOOCV): 5923.7125218460515
R-Squared: 0.21320915292277276
Equation: y = 172.1663 + 11.9301 * PC1 + -2.7821 * PC2 + 0.0000 * PC3 + 0.0000 * PC4 + -0.0000 * PC5 + 0.0000 * PC6 + 0.0000 *
PC7 + 0.0000 * PC8 + -0.0000 * PC9 + -0.0000 * PC10
```

With a high penalty term, it seems like only the first two latent spaces are used; again, I think the models are trying to underfit more to the data because of the high variability of a running back. However, it may be too underfit because it's r-squared is too low to predict anything well.

Conclusion

Looking at the results from all the of the models of every position, it seems like there is no clear answer for how to model points21. As expected, the quarterback and running back positions didn't produce any good models because those positions play with the most factors out of their own control. I think the results are null for all positions except for the tight-end.

The best models seemed to be from the tight end; both the decision tree and lasso results were solid. There seemed to be a stronger correlation between variables like owners20 and the receptions variables and the dependent variable for tight end as compared to other positions. In turn, these variables could be condensed into latent variables that could explain a larger proportion of the variance. This explains why in the lasso model, despite only few PCA variables were involved in the equation, the r-squared came out to be a very high .49. Although there are flaws with this lasso regression like later PCA variables are prioritized over earlier PCA variables that explain more variance, I think one of the only good models that explains fantasy football points in 2021.