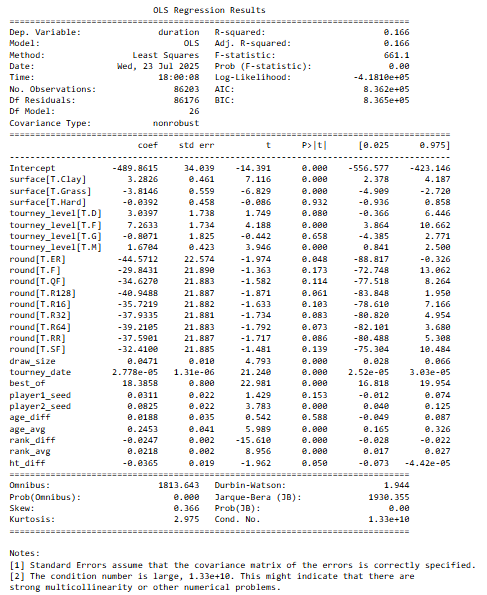
**Performing Model Selection**

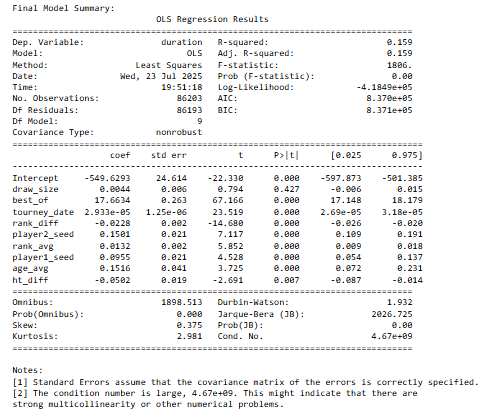
For reference, we will run the original model.





1. ***Forward Selection***

We will first try using forward selection to improve our model. The code starts off with only the intercept and begins to add variables one by one to find the best model to predict **duration**. It chooses the variable with the lowest p-value (less than 0.05) each time. The process will stop when no more useful variables are found. After which, we can observe the output shown below



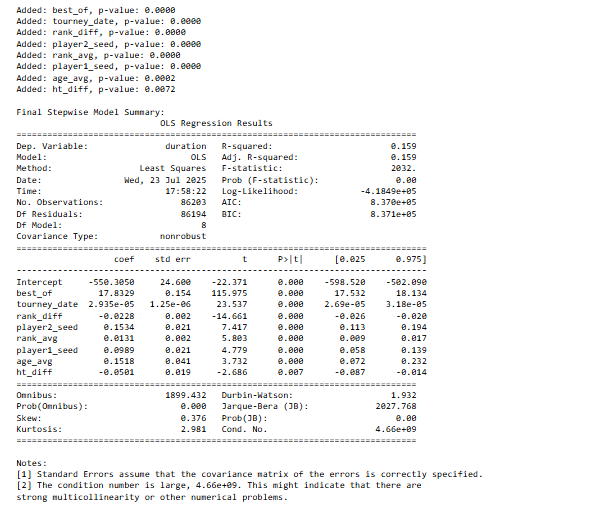
However, after forward selection, we still can observe that the p-value of draw-size is still 0.427.

Though this technique selects most variables that are statistically significant, some variables kept in the model after all variables are selected are still statistically insignificant.

1. ***Step-wise Selection***

Hence, we will be using **stepwise selection**, which differs from forward selection by not only adding variables that are statistically significant, but also checking and removing previously added variables if they become insignificant. We will set the p-value threshold to 0.05. This ensures that only predictors that remain statistically significant in the presence of others remain in the model.

After the selection process, we observe the output as such.



Now we can observe that only statistically significant predictors have been chosen. While the R-squared value has decreased, the overall model has improved.

***How Has the Model Improved?***

1. Model Simplicity

The model done using step-wise selection only has 8 predictors, compared to the original model containing 26 variables. This is a narrowed down model that only has statistically significant variables, making it easier to interpret and less likely to overfit. Additionally, the AIC and BIC still decreased slightly, indicating the model fits the data well while using fewer variables.

1. Lower Condition Number (Reduced Multicollinearity)

The lower condition number (4.66 + 09) indicates that the model has reduced in multicollinearity. This means the predictors selected are not strongly correlated with one another, making a more stable model.

***R-squared Value***

The R-squared value dropped, from 0.166 to 0.159. This is expected, as the significant drop in the number of variables will cause the R-square value to drop. However, the model is still improved due to the simplicity and reduced multicollinearity.

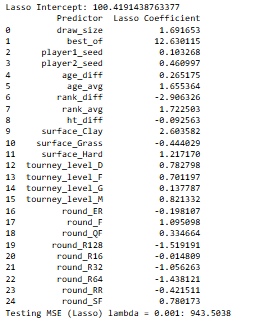
***3) Lasso Regression***

For lasso regression, we will start by selecting the same variables used in the original model, and convert the categorical ones into numeric format using one-hot encoding.

Next, we will split the data into training and testing sets, and standardize the predictors. We will then train the Lasso model using an estimated tuning parameter (lambda = 0.001) to penalize less important features. After fitting the model, we can inspect the resulting coefficients to identify which predictors are retained. Lastly, we will evaluate the model's performance on the test data by calculating the Mean Squared Error (MSE).

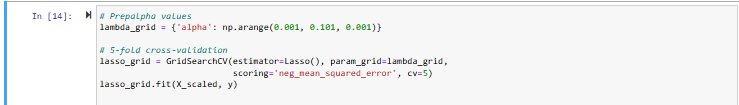


From the output below, we can see that all predictors were retained, meaning none were removed by the model. Variables like best\_of, rank\_diff, and surface\_Clay showed strong influence on duration.

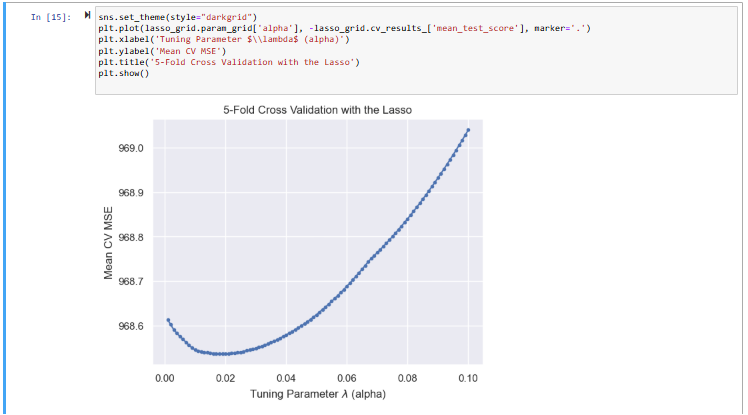


***Selecting the Best Tuning Parameter***

Next, to identify the best tuning parameter, we can perform 5-fold cross-validation to find the best tuning parameter for the Lasso regression model. This is done by testing multiple alpha values from 0.001 to 0.1 to minimize the mean squared error.

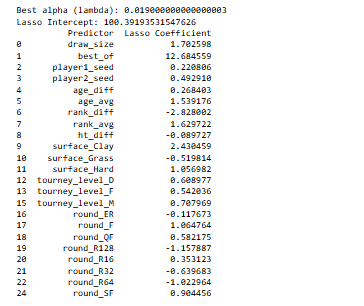


We will plot the mean cross-validated MSE against different values of the tuning parameter. This will help us identify the alpha value that gives the lowest MSE. In the graph shown below, we look for the lowest point on the curve, where the model performs best.



From the output below, compared to the initial Lasso model using lambda = 0.001, this updated model uses a tuned lambda = 0.019, from 5-fold cross-validation.

Despite a slightly higher regularization strength, all predictors still remain in the model, indicating they are still useful. Some coefficients have slightly shrinked. This suggests the model has become a bit more conservative while retaining similar predictive structure.



Compared to the baseline, the Lasso model is improved from the original OLS model because it applies a penalty that shrinks less important coefficients, which helps to reduce overfitting. This helps simplify the model while maintaining predictive performance. As a result, the model generalizes better to unseen data.