Our objective for this project is to help out a local Casino in some bad Blackjack losses. In response to the casino losing over 1.5 million dollars this past month due to an increased loss percentage, specifically in Blackjack; my prioritized task is to figure out a good estimated percentage of blackjack(21) hits for the dealer, compared to the amount of blackjack hits for the player, per game.

We're going to use a dataset titled 'Blackjack Dataset' built with the information we should need to get the job done as well as a smaller CSV with the first 500,000 rows of data we created to save time, given such a quick time frame. Our goal is to achieve a consistent 17% win increase every month for each blackjack dealer over the next 6 months. With this goal being achieved, we predict the casino to gain approximately a 2.7 million dollar monthly revenue increase.

We intend to achieve this goal by increasing the amount of card decks used per game to 10, rather than 8. We also suggest upgrading the headset equipment for the blackjack dealers, as well as, the software used by the support team so that they can both communicate instantaneously mid-game.

A majority of the constraints are likely to be heard from: 1. The players in blackjack, regarding the increase in losses and 2. The casino owner, due to the 1.2 million dollar cost for the headset equipment and software needed to obtain this goal.

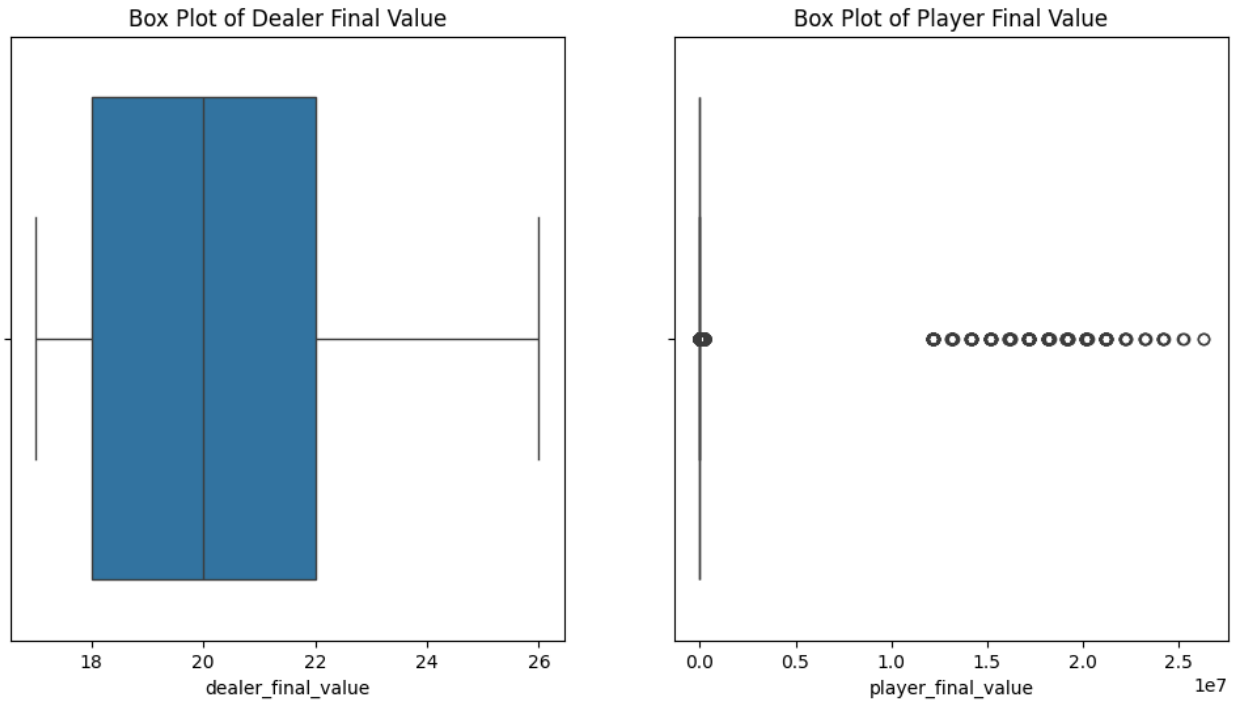
Our main stakeholders involved are: the blackjack dealers- James Johnson, Jim Adams, Kelly Rice, Stacy Williams, John Kline, and William Sanders; the Casino Owner- Ralph Jones, and the Support team supervisor- Debra Jenkins.

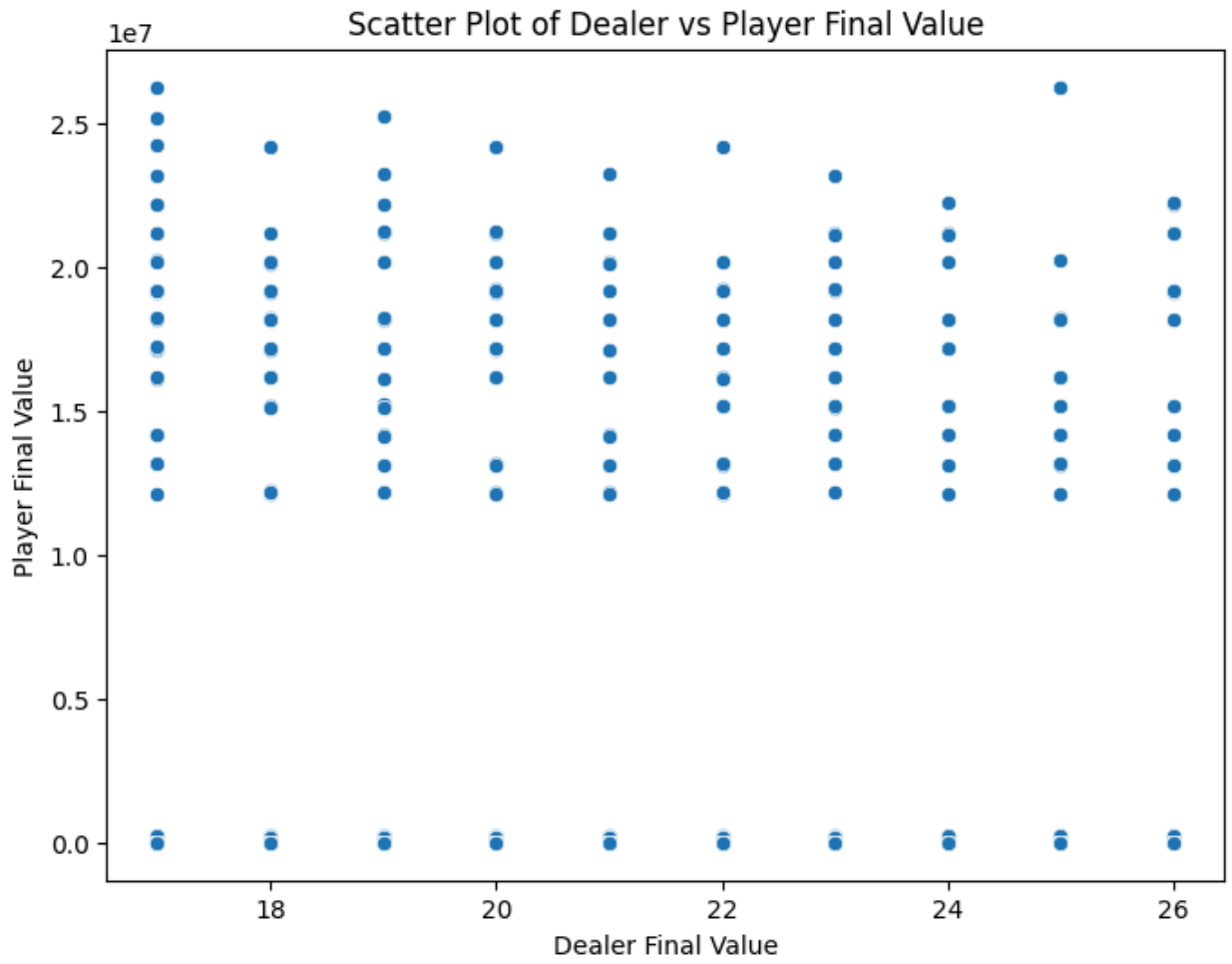
This is where we collect and preprocess the data to have it ready for the following steps. We remove any leading/trailing whitespace to have cleaner data to work with. Then we replace any non-numeric characters (J, Q, K, A) with empty strings and convert to numeric.

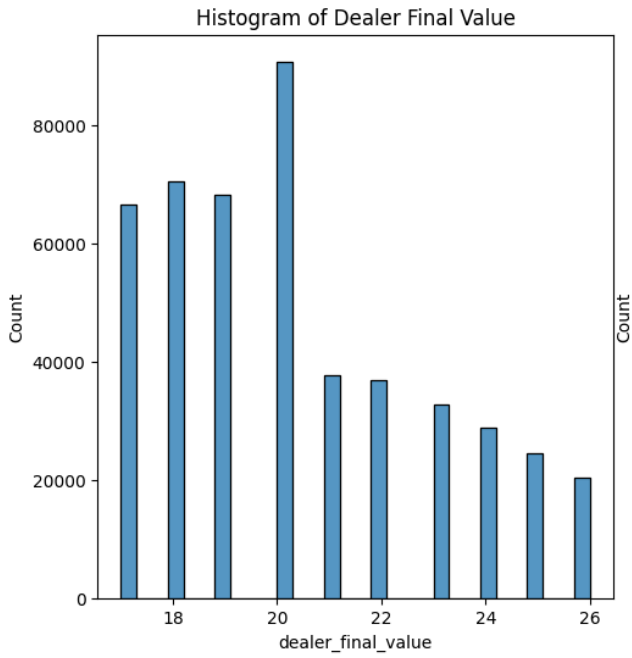
After cleaning the data and removing rows with missing values in both columns as well as removing duplicate rows based on all the columns, we're now working with 476,524 rows of data instead of 500,000.

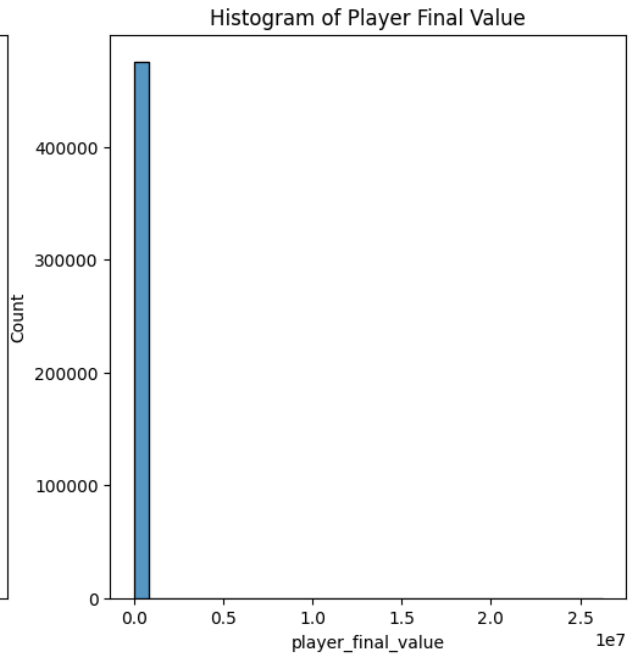
Now we are defining, identifying, and removing any outliers after calculating the IQR for the dealer_final_value and the player_final_value.

After that cleaning it looks like the amount of rows we will be working with is 441,377 and just 12 columns needed because the others don't have any data relevant to our goals for this project. Take a look at some early models:




Scatter Plot of Dealer vs Player Final Value
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Player Final Value
18
20
22
24
26
Dealer Final Value


Histogram of Dealer Final Value
80000
60000
40000
20000
0
Count
18
20
22
24
26
dealer_final_value


Histogram of Player Final Value
400000
300000
200000
100000
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
1e7
player_final_value

Now it is time to use a Standard Scalization method to split the data into training and testing subsets with X defining features and y being the target variable Let's take a look at the data shapes after splitting:

```
Shapes after splitting:
X_train: (353101, 5)
X_test: (88276, 5)
y_train: (353101,)
y_test: (88276,)
```

The y_train and X_train have the same number of rows, which is good for clean testing, however, the X_train has 5 columns where the y_train only has 1. That is because we need more data regarding the dealer in order to make the adjustments we're aiming for to get the dealer back ahead of the game rather than the player.

For our modeling steps we are going to use 3 different model types to see which will be best for our specific investigations. Those 3 model types are the Linear Regression model, the Decision Tree Regressor, and the Random Forest Regressor. We are going to do the first 2 models using the GridSearchCV and the last using the RandomizedSearchCV(for the sake of time).

Linear Regression:

```
param_grid = {
    'fit_intercept': [True, False],
    'copy_X': [True, False],
    'positive': [True, False]
}
```

```
Best parameters: {'copy_X': True, 'fit_intercept': True, 'positive': True}
Best score (negative MSE): -8.592813017987051
```

```
linear regression Mean Squared Error: 8.605462924211428
linear regression rmse: 2.933506932702125
linear regression R-squared: 0.1491550697082501
```

Decision Tree Regressor:

```
param_grid_dt = {
    'max_depth': [3, 5, 7, 10, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2'],
    'random_state': [42]
}
```

```
Best parameters for Decision Tree: {'random_state': 42,
'min_samples_split': 2, 'min_samples_leaf': 2, 'max_features': 'sqrt',
'max_depth': 10}
Best score (negative MSE) for Decision Tree: -5.396477493350279
```

```
Decision Tree Mean Squared Error: 5.643332066437312
Decision Tree rmse: 2.3755698403619525
Decision Tree R-squared: 0.4420287994999349
```

Random Forest Regressor:

```
param_dist_rf = {
    'n_estimators': [100, 200, 500],
    'max_depth': [5, 10, 15, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2'],
    'random_state': [42],
    'n_jobs': [-1]
}
```

```
Best parameters for Random Forest: {'random_state': 42, 'n_jobs': -1,
'n_estimators': 200, 'min_samples_split': 2, 'min_samples_leaf': 4,
'max_features': 'sqrt', 'max_depth': 10}
Best score (negative MSE) for Random Forest: -5.37282833400743
```

```
Random Forest Mean Squared Error: 5.401711237531769
Random Forest rmse: 2.3241581782511638
Random Forest R-squared: 0.4659184913314769
```

I feel that the Random Forest Regressor is the best model to use in this specific modeling test. I do feel that the Decision Tree model is a close runner-up and it was done using GridSearchCV rather than RandomizedSearchCV so it may actually be a more reliable model to use for this assignment.

For future assignments with this dataset I would love to crunch and clean the data a bit more to get a more accurate and specific estimate for getting the dealer back to producing income rather than taking such big losses. But luckily this method and formula has helped the casino out of the hole due to losses and got them back breaking even so I feel it was a pretty beneficial project for all.

Data Source:

https://www.kaggle.com/datasets/fry1999/blackjack-dataset/data