Statistical Inferences Report: Correlation and Regression

Minerva University

CS51: Formal Analyses

Prof. Ribeiro

January 31, 2022

Introduction

eBay is a muti-billion-dollar company whose primary business is facilitating bidding of items on its platform for a fee. The dataset is from the auction of a game for a console on eBay. In this analysis, we shall attempt to answer the research question 'Is there a linear relationship between number of bids and the total price of a game auctioned on eBay?'. The analysis will give insights into the gaming market by reviewing the secondary gaming market found on auctioning platforms such as eBay.

During the analysis, we shall calculate the Pearson's r, check the conditions for and generate a line of best fit using the least squares method and calculate the R^2 value of the simple linear model. We shall further attempt to create a multi-linear regression model using all the applicable datapoints. We shall use various python packages such as NumPy, Matplotlib and SciPy in our analysis. We shall also generate custom Python functions and other function's generated in the Minerva CS Classroom to complement the analysis done.

Dataset

The dataset we shall be working with was retrieved from the OpenIntro sample datasets. The dataset is for the auction data for the game Mario Kart for the Nintendo Wii. The auction is done on eBay and was collected in early October 2009.

The dataset shall be used as a sample for the video game population. We are interested in finding out whether there is a linear relationship between the number of bids and the total price of the game at the end of the auction.

There are two variables we shall be using in our analysis: number of bids and total price. The number of bids is a quantitative variable since there is a numerical measurement to the data. The measurements are discrete because of the nature of how the bidding system work, where the click of the 'bid' button is counted as an auction. The second variable is the total price. This is a quantitative variable for the same reasons. It is a discrete variable as well because of the eBay auction system only allows for up to 2 decimal places when setting the bidding price, setting a clear distinction between one price and the next. The number of bids shall be our predictor variables and total price be the response variable when applying regression.[1]

The data was pre-processed to remove any rows containing empty values for the columns `Number of bids` and `Total Price`. Furthermore, in order to carry out multiple regression, all rows containing an empty column is removed from the dataset.

[1] #variables: I classify and describe the two variables we shall focus on in our analysis. The delineation of the variables is necessary in aspects such as regression, in which we have to define a predictor and success variables.

Methods

Data visualization

In order to carry out any analysis, we shall first generate the descriptive statistics of the two variables and generate the appropriate graphs to represent the data. The descriptive stats are summarized in table 1 and the code used is in Appendix A. The histograms of the two variables as well as a scatter plot showing the two variables is in figures 1,2 and 3 respectively.

Table1: Summary statistics for the sample Number of bids and Total Selling price of the Mario Kart Game for the Nintendo Wii for the month of October 2009

	Number of bids	Total Price
Count	n = 141	n = 141
Mean	$\bar{x} \approx 13.38$	$\bar{y} \approx 47.43$
Mode	16	46
Median	14	46.03
Range	28	46.02
Standard Deviation	$s_x \approx 5.74$	$s_y \approx 9.08$

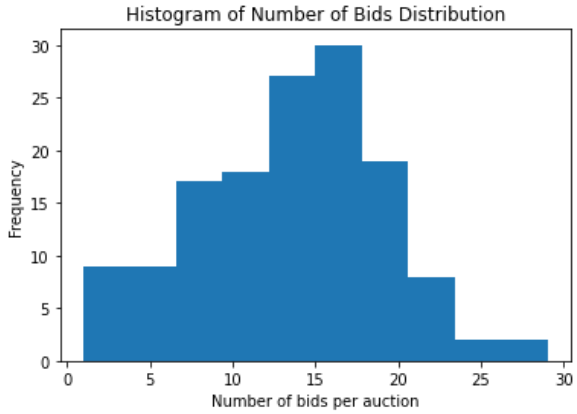

Figure 1. Histogram for Number of bids per auction

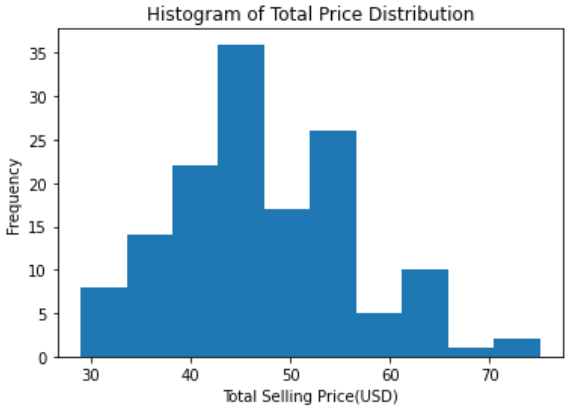

Figure 2. Histogram for Total Selling Price

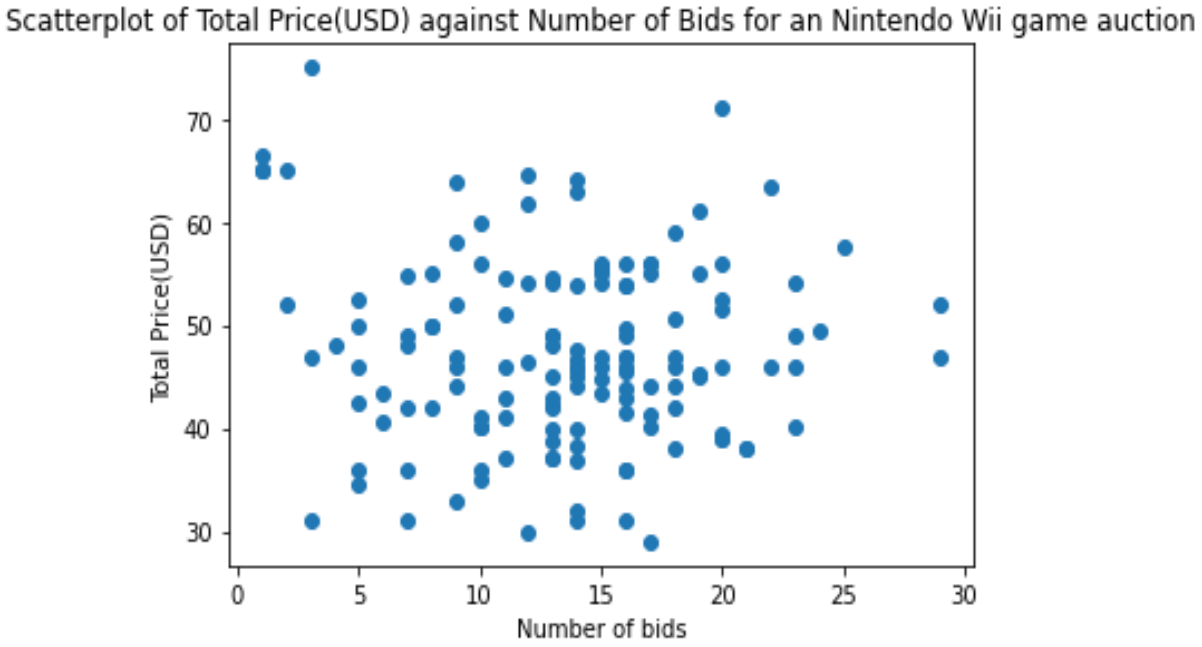

Figure 3. A scatterplot of the total price of a game in USD against the number of bids per auction.[2]

Linear correlation

The first analysis we shall conduct is quantifying the linear correlation between the two variables. We shall use the Pearson's r and its formula $r = \frac{1}{n-1}\sum_{i=1}^{n}(\frac{x_i-\bar{x}}{s_x})(\frac{y_i-\bar{y}}{s_y})$ to quantify the correlation for the sample data. We shall first check if the conditions for linear correlation are met. The conditions are:

- **Heteroscedastic**: This condition is met by checking figure 3, where the variability in Total Price across different slices of x (Number of Bids) is close
- **Linearity**: This condition is not met by checking the scatter plot of the two variables in figure 3, where no trend is discernable and the distribution of the datapoints does not resemble the shape of a football.
- **Presence of outliers**: This condition is not met due to existence of several outliers in the top right of the scatter plot in Figure 2. These outliers do not accurately represent the correlation between the variables due to their distance from the other variables. Furthermore, two outliers were removed because of their strong skew on the data.

Assuming that all the conditions are met, we proceed with calculating the Pearson's r. The calculations done and the relevant code can be found in Appendix B.

[2] #dataviz: I use a scatterplot to visualize the bivariate data. I further interpret the scatterplot when checking various conditions at further parts of the analysis.

Regression

The next analysis is regression, where we shall generate a linear regression model and compute the r value of the linear model. The conditions for using the least squares line regression have to be met in order to carry out the analysis. The conditions are:

- **Linearity**: This condition is not met by checking the scatterplot of the two variables in Figure 3. The same reasons can be found when checking for the same conditions for linear correlation
- **Near normalcy of residuals**. This condition is not met by checking the residual plot in Figure 4, where the residuals are not evenly distributed about the 0-horizontal line. Drawing a vertical line about the middle of the plot, the two halves created do not have the same distribution, with the left half having more residuals above the best fit line and the right half having more of its residuals below the best fit line.
- **Independence**: this condition is not met by checking the nature of the gaming auction community and society as a whole. The same people can bid on more than one item, thereby eliminating independence between individual game auctions.
- **Variability**: This condition is not met. By checking the scatterplot in Figure 5, the variability of the datapoints is not consistent, with mote variability on the left end compared to the right.

The linear regression model, which shall take the form $y = \beta_0 + \beta_1 \mathrm{x}$. For our model, the formula shall be in the form $T = \beta_0 + \beta_1 N$,where N is the response variable (number of bids) and t is the predictor variable (total price). The code for generating the residual plot, and the generation of the model can be found in Appendix C.[3]

[3] #regression: I construct a simple linear regression model by construing a bet fit line using the least squares method. I further interpret the intercept and coefficient of the response variables in the context of the research question.

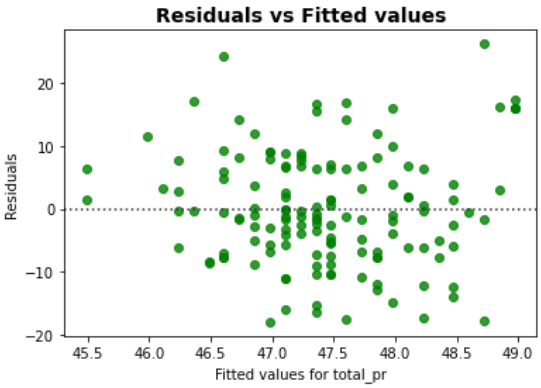

Figure 4. A residual plot for the proposed line of best fit for the two variables. total_pr represent the total price variable.

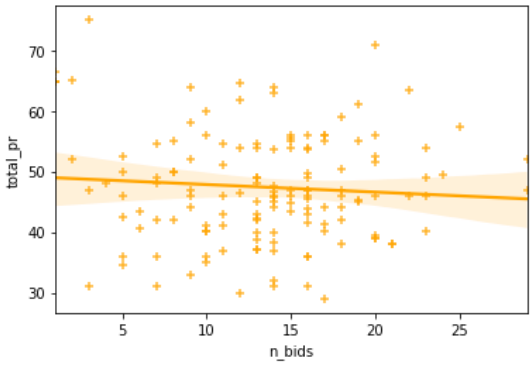

Figure 5. A scatterplot of total price vs number of bids. The scatterplot includes a best fit line using the least squares regression method. n_bids and total_pr represent number of bids and total price (is USD) respectively.

Significance

After deriving our regression model, we shall conduct a hypothesis test to determine the significance of the observed regression on the regression model. To carry out a significance test, she shall first define our hypothesis, which are:

> H_0 (Null hypothesis). There is no linear relationship between the number of bids and total price.
>
> H_A (Alternative hypothesis): There is some linear relationship between number of bids and total price.

To test for the t-value, we shall first generate the t-value of the point estimate suing the formula $t = \frac{estimate - null\ value}{SE}$. The standard error used (SE) shall be calculated using the formula $SE(\beta_1) = \frac{s_y}{s_x}\sqrt{\frac{(1-R^2)}{n-2}}$. The code for the significance test is in appendix D.[4]

Multi-linear regression

Finally, we shall attempt to generate a mule-linear regression model using all the variables provided in the dataset. We shall use the forward selection method of fitting predictor variables. The forward -selection method involves adding variables to the model until the

[4] #significance: I apply a hypothesis test on the coefficient of the predictor variable in the simple linear regression model. I further interpret the results in the Results section of the report.

optimum R^2 value is reached. The code for the forward selection algorithm can be found on appendix E.[5]

Results and Conclusions

With the assumptions that the conditions for linear correlation are met, the results of the linear correlation analysis are a Pearson's value of -.079. This indicates a slight negative linear correlation between the two variables. There is an indirect relation between the number of bids and total price of the game, with an increase in number of bids begetting a decrease in total price. From the analysis, it can be concluded that when there is an increase in number of bids, there is an expected decrease in total selling price. The coefficient value calculated only indicates the magnitude and direction of correlation between the two variables, and does not, for instance, tell whether the conditions for the linear correlation are met. There is a direct causal relationship either between the two variables, because a subsequent bid needs to have a higher price than the last bid, therefore increasing the number of bids causes an increase in total price[6]

The results of the linear regression model are represented by the formula $T = 49.0979 - 0.1245N$, with 49.0979 being the intercept of the model and -0.1245 as the slope. For every single increase in a bid, there is a reduction of about .1245 dollars in the total selling price of the game.

The hypothesis test for the significance of a linear relationship between number of bids and total price returned a p value of .07, which is above the significance level of .05. Based on these results, we fail to reject the null hypothesis since there isn't sufficient evidence in favor of the null hypothesis. In the context of the research question, this means that there isn't a statistically significant linear relationship between the number of bids and the total selling price of the Mario Kart game.

The R^2 value of the linear model is .006, and the adjusted $R^2 = -.001$. This is a relatively weak R^2 value. In context of our research question the R^2 indicates that the variability in total selling price of the game can be attributed by about -.1% to the number of bids the game has. This shows a generally insignificance of the number of bids to the total price of the game

The multi-linear regression model landed on the formula $T = 35.0583 + 7.9884\mathrm{W} + 0.1688\mathrm{SP} + 0.2359\mathrm{N} + (2.349\mathrm{e} - 05)\mathrm{SR} - 0.4887\mathrm{D}$. The symbols are in Table 3 of Appendix E The model was not able to include data containing qualitative data, which requires more advanced models. The multi-linear model has a higher adjusted $R^2 = 0.73$ and much low p-values for most of its coefficients (refer to Appendix E), suggesting that this model would be

[5] #regression: I constructed a multi linear regression model, with the clearly defined response variable of total game price. I further interpret the model in the results portion of the report.

[6] #correlation: I compute the correlation between the two variables using Pearson's r formula. I further interpret the results and tie it back to the research question. I further explore a possible causal relationship between the variables.

better than the simple linear model above at estimating the game total price based of the variables provided.

Assuming the existence of a stronger correlation between the two variables, the model could be used in gauging the eventual selling price of a game based on the number of bids at a particular time. For instance, eBay could trigger the system to lower the seller fees if the number of bids reach a certain number, because the gross revenue generated from the auction would be enough.

Given the results of this analysis, it is reasonable to assume that the number of bids a game has does not determine its value in the secondary market. While the increase in bids causes an increase in prices by the conditions of the eBay auction system, The subsequent bidders do not have to bid immediately to the true value of the time. Moreover, the time that the owner sets for the bid influences the eventual price, with price jumps occurring more often towards the end of bid time.[7]

Some extraneous variables not accounted for include the total number of Wii games in the market, which would influence the overall supply of the game. Another extraneous variable is period after release, in which the longer the game has been out, the more it is available, until the company ceases production after which supply reduces. Finally, another variable not accounted for is the interest in the game, while this may have been captured by the number of bids, and cannot be measured directly, some metric could be used in place of the interest variable.

Reflection

The formal analyses lessons taught last semester and this semester have emphasizes the importance of providing context to the work done. For instance, while a successful regression model can be created, a well-crafted hypothesis test will determine if the model is statistically significant. The statistical significance can be used in explaining the whole regression model to a person with statistical knowledge but no computer algorithmic knowledge.

[7] #induction: I apply inductive reasoning as to why the predictor variable is not the best at predicting the response variable.

Appendix

Appendix A: Importation, processing and visualization of data

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.api as statsmodels
from scipy import stats

dataset = pd.read_csv("mariokart.csv")

```

```

# Remove the two outliers in total price
dataset = dataset[dataset['total_pr'] < 100]
df_x = dataset['n_bids']
df_y = dataset['total_pr']

plt.hist(df_x)
plt.title("Histogram of Number of Bids Distribution")
plt.xlabel("Number of bids per auction")
plt.ylabel("Frequency")
plt.show()

plt.hist(df_y)
plt.title("Histogram of Total Price Distribution")
plt.xlabel("Total Selling Price(USD)")
plt.ylabel("Frequency")
plt.show()


plt.scatter(df_x, df_y)
plt.title("Scatterplot of Total Price(USD) against Number of Bids for an Nintendo Wii game auction")
plt.xlabel("Number of bids")
plt.ylabel("Total Price(USD)")

plt.show()

```

```python
# descriptivestats

def descriptive_stats(datalist, variable_name=""):
    mean = sum(datalist)/len(datalist)
    sd = np.std(datalist)
    _range = max(datalist)-min(datalist)

    print(f"""The Descriptive statisitcs for the column {variable_name} Are:\n
    \t Count: {len(datalist)}
    \t Mean: {mean}
    \t Mode: {stats.mode(datalist)}
    \t Median: {np.median(datalist)}
    \t Range: {_range}
    \t Standard Deviation: {sd}
    """
        )


descriptive_stats(df_x, "Number of Bids")
descriptive_stats(df_y, " Total Price( in USD)")

```

```
The Descriptive statisitcs for the column Number of Bids is:

        Count: 141
        Mean: 13.382978723404255
        Mode: ModeResult(mode=array([16], dtype=int64), count=array([15]))
        Median: 14.0
        Range: 28
        Standard Deviation: 5.743524969082079

The Descriptive statisitcs for the column  Total Price( in USD) is:

        Count: 141
        Mean: 47.43191489361702
        Mode: ModeResult(mode=array([46.]), count=array([8]))
        Median: 46.03
        Range: 46.019999999999996
        Standard Deviation: 9.081276030788121
```

Appendix B: Correlation and Pearson's R

```
# correlation
r, p = stats.pearsonr(df_x, df_y)


def pearsons_r(x, y):
    # calculate pearson's r using the formula
    if len(x) != len(y):
        raise Exception("Improper dataset")
    n = len(x)
    sd_x = np.std(x)
    sd_y = np.std(y)
    mean_x = sum(x)/n
    mean_y = sum(y)/n

    _sum = 0
    for a, b in zip(x, y):
        numerator = (a-mean_x)*(b-mean_y)
        denominator = (sd_x*sd_y)
        _sum += numerator/denominator

    return _sum/n


print("Pearson's r=", pearsons_r(df_x, df_y))
print(r)

```

```
Pearson's r= -0.07873206017839551
-0.07873206017839551
```

Appendix C: Simple Linear Regression

```
# simple linear regression
def multi_regression(column_x, column_y, data):
    ''' this function uses built in library functions to construct a linear
    regression model with potentially multiple predictor variables. It outputs
    two plots to assess the validity of the model.

    Retrieved from : https://sle-collaboration.minervaproject.com/?url=https%3A//sle-authoring.
    minervaproject.com/api/v1/worksheets/aa2e8df4-77e4-4f65-97e4-8feb4316cda7/&userId=11848&
    name=Kyron+Nyoro&avatar=https%3A//s3.amazonaws.com/picasso.fixtures/
    Lewis_Nyoro_11848_2021-08-17T06%3A59%3A18.237Z&noPresence=1&readOnly=1&isInstructor=0&
    signature=af3757169a9e383e43f4a89eeec4119cf01fa142aa7ce4973cb56412a3209326
    '''

    if len(column_x) == 1:
        plt.figure()
        sns.regplot(x=column_x[0], y=column_y, data=data,
                    marker="+", fit_reg=True, color='orange')

    # define predictors X and response Y:
    X = data[column_x]
    X = statsmodels.add_constant(X)
    Y = data[column_y]

    global regressionmodel
    regressionmodel = statsmodels.OLS(Y, X).fit()

    # residual plot:
    plt.figure()
    residualplot = sns.residplot(
        x=regressionmodel.predict(), y=regressionmodel.resid, color='green')
    residualplot.set(xlabel='Fitted values for '+column_y, ylabel='Residuals')
    residualplot.set_title('Residuals vs Fitted values',
                           fontweight='bold', fontsize=14)

    # QQ plot:
    qqplot = statsmodels.qqplot(regressionmodel.resid, fit=True, line='45')
    qqplot.suptitle("Normal Probability (\"QQ\") Plot for Residuals",
                    fontweight='bold', fontsize=14)


multi_regression(["n_bids", ], 'total_pr', dataset)
regressionmodel.summary()

```

OLS Regression Results

Dep. Variable:	total_pr	R-squared:	0.006
Model:	OLS	Adj. R-squared:	-0.001
Method:	Least Squares	F-statistic:	0.8670
Date:	Tue, 01 Feb 2022	Prob (F-statistic):	0.353
Time:	19:29:31	Log-Likelihood:	-510.71
No. Observations:	141	AIC:	1025.
Df Residuals:	139	BIC:	1031.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	49.0979	1.947	25.217	0.000	45.248	52.948
n_bids	-0.1245	0.134	-0.931	0.353	-0.389	0.140

Omnibus:	2.633	Durbin-Watson:	2.074
Prob(Omnibus):	0.268	Jarque-Bera (JB):	2.612
Skew:	0.326	Prob(JB):	0.271
Kurtosis:	2.859	Cond. No.	37.1

Appendix D: Significance Test

```
1
2  def significance_test(alpha, SE=None, sy=None, sx=None, r=None, b1=None, n=None):
3      if b1 is None:
4          # fill in an expression, in terms of the quantities above, to copmpute the point-estimate
5          # for the slope
6          b1 = r*sy/sx
7      print("b1 =", b1)
8
9      if SE is None:
10         SE = (sy/sx) * ((1 - r**2)/(n-2))**0.5
11     print("SE =", SE)
12
13     t = (b1 - 0)/SE
14     print("t =", t)
15
16     p = (1-stats.t.cdf(t, n-2))*2
17     print("p =", p)
18
19
20 params = dict(regressionmodel.params)
21 significance_test(.05, b1=params['n_bids'],
22                    sy=np.std(df_y), sx=np.std(df_x), r=r, n=len(df_y))
23
```

```
b1 = -0.12448584707152377
SE = 0.133693636097995
t = -0.9311276939186388
p = 1.6465980960548086
```

Appendix E: Multi-linear Regression

```python
# forward selection
def forward_selector_regression(dataset, response):
    # Find the multiple linear regeession model using the forward selection method
    # 1. Calculate the r^2 of a simple linear model of each variable
    # 2. Pick the variable with the highest r^2 value
    # 3. Repeat steps 4-7, :
    #   4. Create new models by each adding one of the remaining variables to the preceding model
    #   5. Calculate the r^2 of the new models and compare the r^2 value to that of the previous
    #       model
    #   6. Proceed with the model amongst the generated ones with the highest r^2.
    #       7. If none of the new models have a higher  r^2 value than the , stop and return the
    previous model.

    # Modified from: https://sle-collaboration.minervaproject.com/?url=https%3A//sle-authoring.
    minervaproject.com/api/v1/worksheets/aa2e8df4-77e4-4f65-97e4-8feb4316cda7/&userId=11848&
    name=Kyron+Nyoro&avatar=https%3A//s3.amazonaws.com/picasso.fixtures/
    Lewis_Nyoro_11848_2021-08-17T06%3A59%3A18.237Z&noPresence=1&readOnly=1&isInstructor=0&
    signature=af3757169a9e383e43f4a89eeec4119cf01fa142aa7ce4973cb56412a3209326

    final_columns = []
    optimum = False
    current_r_2 = 0
    predictor_columns = list(dataset.columns)
    predictor_columns.remove(response)
    while optimum is False:
        largest_column = None
        largest_column_r_2 = None
        for column in predictor_columns:
            skip = False
            column_x = final_columns+[column]
            X = dataset[column_x]
            X = statsmodels.add_constant(X)
            Y = dataset[response]
            # construct model:
            global regressionmodel
            try:
                regressionmodel = statsmodels.OLS(Y, X).fit()
            except:
                # The column is not quantitative, therefore cannot be used in our  linear model
                skip = True
            if skip is False:
                if largest_column_r_2 is None or regressionmodel.rsquared > largest_column_r_2:
                    largest_column = column
                    largest_column_r_2 = regressionmodel.rsquared_adj
        if largest_column_r_2 > current_r_2:
            current_r_2 = largest_column_r_2
            final_columns.append(largest_column)
            predictor_columns.remove(largest_column)
        else:
            optimum = True
    print("The final coumns used are ", final_columns)
    X = dataset[final_columns]
    X = statsmodels.add_constant(X)
    Y = dataset[response]
    regressionmodel = statsmodels.OLS(Y, X).fit()


forward_selector_regression(dataset, "total_pr")
regressionmodel.summary()

```

[8]

```
The final coumns used are  ['wheels', 'seller_rate', 'start_pr', 'n_bids', 'duration']
```

OLS Regression Results

Dep. Variable:	total_pr	R-squared:	0.740
Model:	OLS	Adj. R-squared:	0.730
Method:	Least Squares	F-statistic:	76.82
Date:	Tue, 01 Feb 2022	Prob (F-statistic):	9.06e-38
Time:	19:36:39	Log-Likelihood:	-416.20
No. Observations:	141	AIC:	844.4
Df Residuals:	135	BIC:	862.1
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	35.0583	1.780	19.695	0.000	31.538	38.579
wheels	7.9884	0.539	14.830	0.000	6.923	9.054
seller_rate	2.349e-05	8.61e-06	2.728	0.007	6.46e-06	4.05e-05
start_pr	0.1688	0.038	4.468	0.000	0.094	0.244
n_bids	0.2359	0.091	2.579	0.011	0.055	0.417
duration	-0.4887	0.174	-2.801	0.006	-0.834	-0.144

Omnibus:	7.899	Durbin-Watson:	1.889
Prob(Omnibus):	0.019	Jarque-Bera (JB):	7.795
Skew:	0.479	Prob(JB):	0.0203
Kurtosis:	3.641	Cond. No.	2.46e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.46e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

Table 2. The symbol notation of various variables in the multi-linear regression model

Symbol	**Measurement**
W	Number of wheels included
N	Number of bids
S	Starting price
D	Duration of bid
SR	Seller Rating
SP	Start Price

[8] #algorithms: I implement the forward selection multi-linear regression strategy using clear, concise steps. The code is well documented showing all the steps of the algorithm