[Year]

Assignment 4

DATA, INFERENCE & APPLIED MACHINE LEARNING (COURSE 18-785)

YASSER CORZO

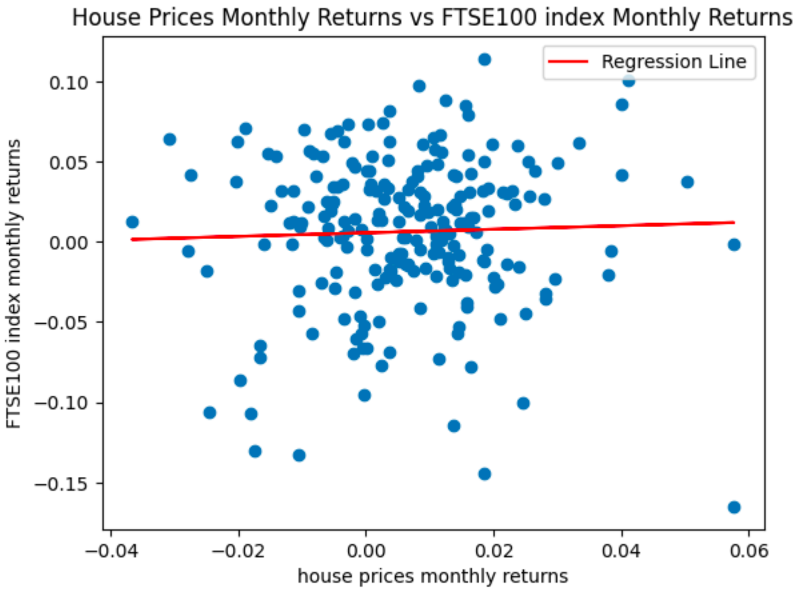
CARNEGIE MELLON UNIVERSITY | 5000 Forbes Ave, Pittsburgh, PA 15213

LIBRARIES USED

- Numpy
- Pandas
- Scipy
- Sklearn
- Statsmodels
- Stepwise_Regression
- Tabulate

REPORT

Q1



a.

```
Correlation Coefficient: 0.034976120776073316
```

Above is the regression model using the statsmodels [1] library. This was done by retrieving the monthly house prices data and FTSE100 index from 01-Jan-1991 to 31-Dec-2016, matching both datasets by their corresponding dates. Calculating the monthly returns for both datasets involved using pandas. Then, as the problem instructed, I added a constant to the explanatory variable (house prices monthly returns) and used statsmodel to create the linear regression model shown above. I used scipy's rvalue [2] method to calculate the correlation coefficient.

b. The results tell us that there is very little correlation between the house prices monthly returns and the FTSE100 index monthly returns. This is shown in the correlation coefficient that has a value very close to 0 and in the graph where the predicted line has a very small slope.

c. My null hypothesis is that there's no relationship between the house prices monthly returns and the FTSE100 index monthly returns. The alternative hypothesis is that there does exist a significant relationship between these two variables.

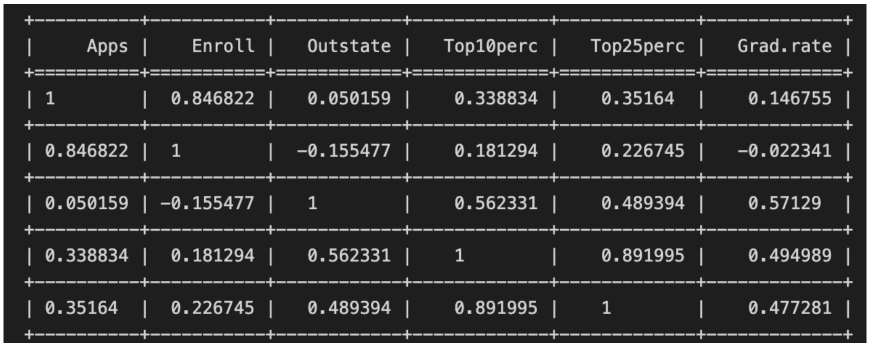
```
p-value: 0.6050395332131359
```

Based on the p-value, since it's greater than alpha, null hypothesis is not rejected. Hence, there's no significant relationship between house prices monthly returns and FTSE100 index monthly returns.

Calculating the p-value needed for hypothesis testing was done using scipy's pvalue [2] method.

Q2

a.



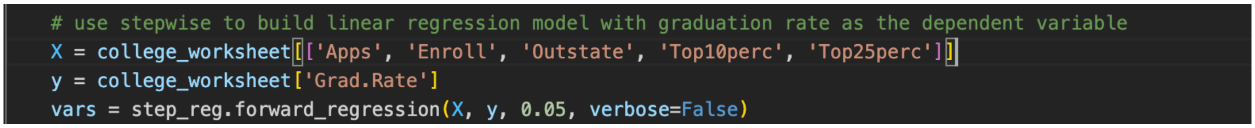
```
+----------+-----------+------------+-------------+-------------+-------------+
|     Apps |    Enroll |   Outstate |    Top10perc |   Top25perc |   Grad.rate |
+==========+===========+============+=============+=============+=============+
| 1        |  0.846822 |   0.050159 |    0.338834 |    0.35164  |    0.146755 |
+----------+-----------+------------+-------------+-------------+-------------+
| 0.846822 |  1        |  -0.155477 |    0.181294 |    0.226745 |   -0.022341 |
+----------+-----------+------------+-------------+-------------+-------------+
| 0.050159 | -0.155477 |   1        |    0.562331 |    0.489394 |    0.57129  |
+----------+-----------+------------+-------------+-------------+-------------+
| 0.338834 |  0.181294 |   0.562331 |    1        |    0.891995 |    0.494989 |
+----------+-----------+------------+-------------+-------------+-------------+
| 0.35164  |  0.226745 |   0.489394 |    0.891995 |    1        |    0.477281 |
+----------+-----------+------------+-------------+-------------+-------------+
```

Above is the correlation coefficient table of all the aforementioned variables with each other (i.e. value in row i, column j would be the correlation coefficient of variable i and variable j). The last column contains the correlation coefficients of all the predictor variables against the target variable, Graduation rate.

Calculating the correlation coefficient for every variable was done with using scipy [3], displaying the coefficients in a table with the help of tabulate.

Based on the coefficients between the predictor and target variables, it seems that the number of out of state students most correlations with graduation rate.

b.

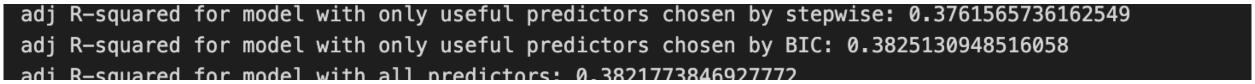


```
    # use stepwise to build linear regression model with graduation rate as the dependent variable
    X = college_worksheet[['Apps', 'Enroll', 'Outstate', 'Top10perc', 'Top25perc']]
    y = college_worksheet['Grad.Rate']
    vars = step_reg.forward_regression(X, y, 0.05, verbose=False)
```

c. The predictor variables that are useful in predicting the graduation rate are the number of out of state students and the number of students who were in the top 25% of their class. I got these variables by using stepwise to build the linear regression model. By using the step_reg.forward_regression method [4] from stepwise_regression, setting a threshold of 0.05 for features to be added, I get the best predictor variables to use.

d. The set of predictor variables would be useful in predicting the graduation rate if we were to use BIC to select the model by using BIC as the criterion with sklearn's LassoIC.coeff_ method [5] and see which features output a coefficient of 0. These features would have to be removed as a feature with a coefficient of 0 is insignificant. As a result, the features chosen by BIC are the number of applications received, number of enrolled students, number of out of state students, and the number of admitted students who were in the top 25% of their class.

e. To compute the accuracy of the two models I will compare (model with only useful predictors and model with all predictor variables) I used R2-score, specifically the adjusted R2-score. Scores are shown below:



```
adj R-squared for model with only useful predictors chosen by stepwise: 0.3761565736162549
adj R-squared for model with only useful predictors chosen by BIC: 0.3825130948516058
adj R-squared for model with all predictors: 0.3821773846927772
```

Since the model with only useful predictors chosen by BIC has the highest adjusted R2-score, it has better accuracy than the model with only useful predictors.

To calculate the R2-score of both models, I used statsmodel's rquared_adj method [6].

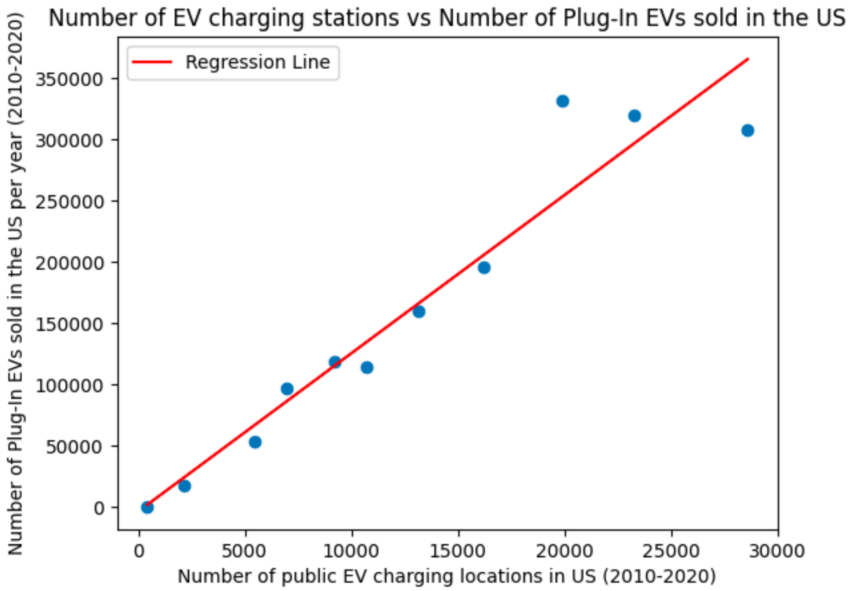
f. The most accurate model predicts that Carnegie Mellon University's graduation rate is 89.12510268464595 %. Actual graduation rate is 74%, so my model over-estimated.

Q3

The trend that I am studying for this report is the relationship between increase in the number of public EV charging stations and the number of EVs sold in the United States between 2010-2020. Datasets that I used were the Hybrid-Electric, Plug-in Hybrid-Electric and Electric Vehicle Sales from the US Department of Transportation [7] and U.S. Public and Private Electric Vehicle Charging Infrastructure from the US Department of Energy [8].

My assumptions that I made were that there is a significant directly proportional relationship between the number of public EV charging stations and the sales of EVs.

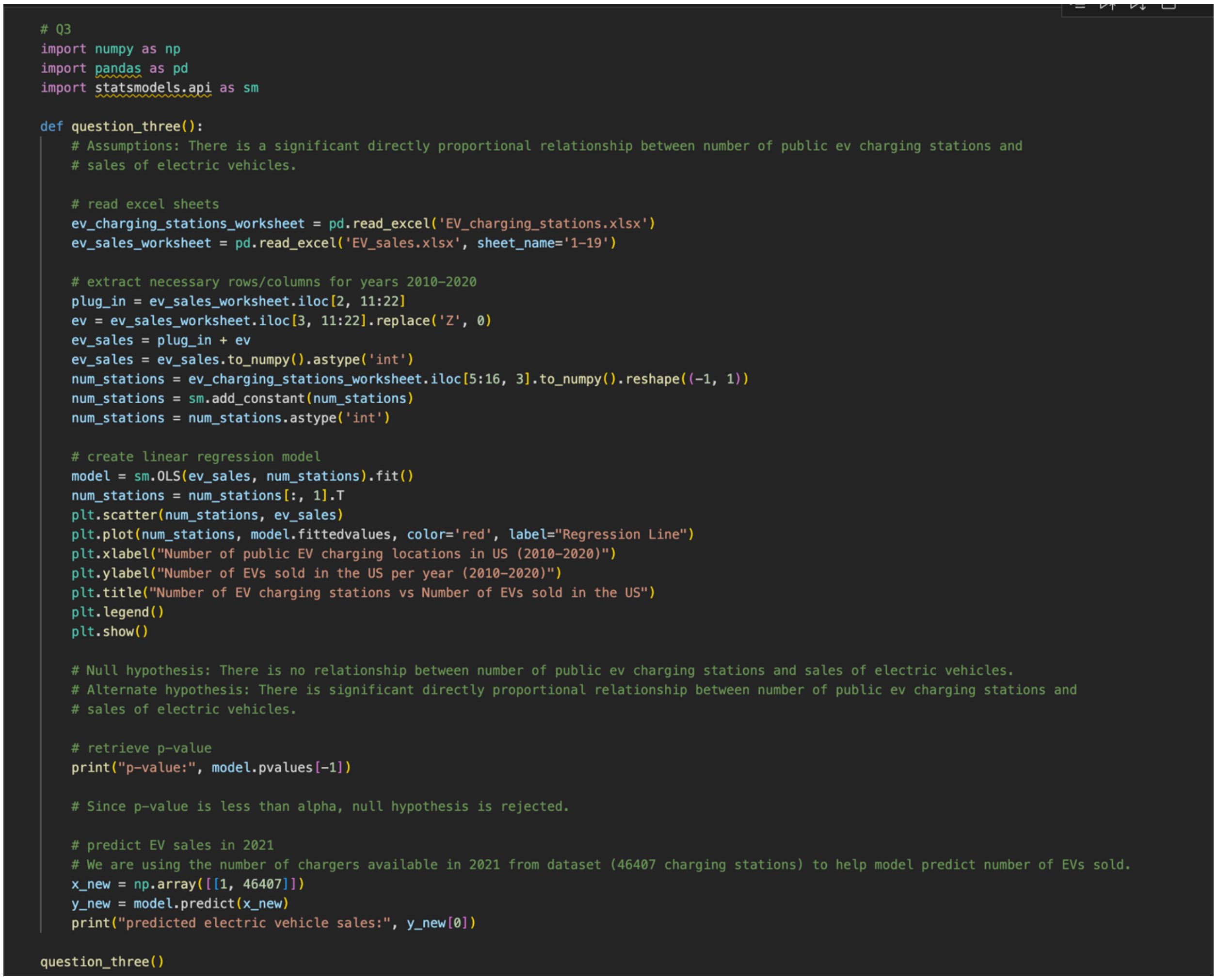
The methodology used to determine whether there is a relationship between these two variables was with hypothesis testing. My null hypothesis was that there is no relationship between the number of public EV charging stations and the sale of electric vehicles in the US. The alternative hypothesis was that there is a significant relationship between the number of public EV charging stations and the number of Plug-In EVs sold per year in the US. It is important to note that by Plug-In EVs, I'm referring to Plug-In Hybrid-Electric and Electric Vehicles.



Above is the linear regression model between the number of public EV charging stations in the US (2010-2020) and the number of Plug-In EVs sold in the US per year (2010-2020). As shown by the model, there does seem to exist a correlation between these two variables as the slope of the regression line does seem to be above 0.

```
p-value: 2.4693241564609166e-06
```

Above is the p-value of the model. Since the p-value < alpha, the null hypothesis is rejected. Therefore, by hypothesis testing, there is a relationship between the number of public EV charging stations in the US and the number of EVs sold in the US per year. Using this model, I was able to predict the number of EVs sold in 2021 to be around 594634 vehicles. The true number of Plug-In EVs sold in the US was 632,883.



```
# Q3
import numpy as np
import pandas as pd
import statsmodels.api as sm

def question_three():
    # Assumptions: There is a significant directly proportional relationship between number of public ev charging stations and
    # sales of electric vehicles.

    # read excel sheets
    ev_charging_stations_worksheet = pd.read_excel('EV_charging_stations.xlsx')
    ev_sales_worksheet = pd.read_excel('EV_sales.xlsx', sheet_name='1-19')

    # extract necessary rows/columns for years 2010-2020
    plug_in = ev_sales_worksheet.iloc[2, 11:22]
    ev = ev_sales_worksheet.iloc[3, 11:22].replace('Z', 0)
    ev_sales = plug_in + ev
    ev_sales = ev_sales.to_numpy().astype('int')
    num_stations = ev_charging_stations_worksheet.iloc[5:16, 3].to_numpy().reshape((-1, 1))
    num_stations = sm.add_constant(num_stations)
    num_stations = num_stations.astype('int')

    # create linear regression model
    model = sm.OLS(ev_sales, num_stations).fit()
    num_stations = num_stations[:, 1].T
    plt.scatter(num_stations, ev_sales)
    plt.plot(num_stations, model.fittedvalues, color='red', label="Regression Line")
    plt.xlabel("Number of public EV charging locations in US (2010-2020)")
    plt.ylabel("Number of EVs sold in the US per year (2010-2020)")
    plt.title("Number of EV charging stations vs Number of EVs sold in the US")
    plt.legend()
    plt.show()

    # Null hypothesis: There is no relationship between number of public ev charging stations and sales of electric vehicles.
    # Alternate hypothesis: There is significant directly proportional relationship between number of public ev charging stations and
    # sales of electric vehicles.

    # retrieve p-value
    print("p-value:", model.pvalues[-1])

    # Since p-value is less than alpha, null hypothesis is rejected.

    # predict EV sales in 2021
    # We are using the number of chargers available in 2021 from dataset (46407 charging stations) to help model predict number of EVs sold.
    x_new = np.array([[1, 46407]])
    y_new = model.predict(x_new)
    print("predicted electric vehicle sales:", y_new[0])

question_three()
```

Q4

The likely rate of unemployment by the year 2020 is 12.078691048979863%. One can evaluate the accuracy of the estimate by using MAPE. The MAPE of my estimate is 21.992602606313422%.

I approached this problem by downloading the data from Quandl and extracting the necessary data from 1980-12-31 to 2013-09-02, trimming the rest. Converting the dates to numbers was important as dates can't be used to perform linear regression. Using sklearn's LinearRegression class [9] I created a linear regression model of the unemployment rates. To predict the unemployment rate for 2020, I had to add explanatory variables to the model (i.e. dates from 2013-2020) so the model could predict the unemployment rate for years after 2012, which is where the data ends. Using sklearn's mean_absolute_percentage_error library [10], I was able to calculate the MAPE of the system for accuracy.

REFERENCES

[1] “Linear regression¶,” Linear Regression - statsmodels 0.14.0, https://www.statsmodels.org/stable/regression.html (accessed Oct. 23, 2023).

[2] “Scipy.stats.linregress#,” scipy.stats.linregress - SciPy v1.11.3 Manual, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html (accessed Oct. 23, 2023).

[3] “Scipy.stats.pearsonr#,” scipy.stats.pearsonr - SciPy v1.11.3 Manual, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html (accessed Oct. 23, 2023).

[4] “Stepwise-regression,” PyPI, https://pypi.org/project/stepwise-regression/ (accessed Oct. 23, 2023).

[5] “Sklearn.linear_model.Lassolarsic,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoLarsIC.html#sklearn.linear_model.LassoLarsIC.score (accessed Oct. 23, 2023).

[6] “Statsmodels.regression.linear_model.Olsresults.rsquared_adj¶,” statsmodels.regression.linear_model.OLSResults.rsquared_adj - statsmodels 0.14.0, https://www.statsmodels.org/stable/generated/statsmodels.regression.linear_model.OLSResults.rsquared_adj.html (accessed Oct. 23, 2023).

[7] “Hybrid-electric, plug-in hybrid-electric and electric vehicle sales,” Hybrid-Electric, Plug-in Hybrid-Electric and Electric Vehicle Sales | Bureau of Transportation Statistics, https://www.bts.gov/content/gasoline-hybrid-and-electric-vehicle-sales (accessed Oct. 23, 2023).

[8] “Maps and Data,” Alternative Fuels Data Center: Maps and Data, https://afdc.energy.gov/data/search?q=U.S.%2BPublic%2Band%2BPrivate%2BElectric%2BVehicle%2BCharging%2BInfrastructure%2B (accessed Oct. 23, 2023).

[9] “Sklearn.linear_model.linearregression,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html (accessed Oct. 23, 2023).

[10] “SKLEARN.METRICS.MEAN_ABSOLUTE_PERCENTAGE_ERROR,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_percentage_error.html (accessed Oct. 23, 2023).