DATA1002 Project Stage 3

Part 1

Linear Regression Analysis

The COVID-19 pandemic has affected society in many ways throughout 2020. One such area of society that has been impacted by COVID-19 is the financial markets. The financial markets are where the buying and selling of financial products takes place. The value of financial products are dictated by the market forces of demand and supply. Amongst other factors, the increase in COVID-19 cases has influenced these market forces, and thus the values of financial products. With production of a viable vaccine expected in the medium to long term, the COVID-19 pandemic is set to continue affecting society and thus the financial markets. As this is the case, financial analysts may wish to model and predict how future COVID-19 cases may affect the values of financial products.

Indices are financial products that consist of a hypothetical portfolio of securities, representing a particular market or sector within the financial markets. An indice/index essentially provides a measure of value for the market or sector it represents. The Financials, Healthcare, Info Tech and Resources Indices are indices that represent the value of stocks in each of the Financials, Healthcare, Info Tech and Resources sectors respectively. Each sector and its economic performance has been affected by COVID-19. This has led to fluctuations in value of each index. The following analysis will attempt to determine how much of an effect increases in COVID-19 cases have had on fluctuations in the value of the indexes. A linear regression model will be constructed and utilised to make predictions on value fluctuations for the indexes, based on increases in COVID-19 cases.

An effective prediction model would have several implications. The most obvious implication would be that investors or speculators would have a better knowledge on when the value of the indexes would be expected to fluctuate. Knowing fluctuation patterns could aid in the deliberation of investment decisions, relating to when to buy or sell index products . Ultimately this could help to improve the performance and profit accrual of index trading investors. Another implication would be the effect of the prediction actually causing value changes. A prediction by the model would influence the decisions of investors. For example, if the model predicts for a certain increase in COVID-19 cases that the value of the index would fluctuate dramatically, then investors would expect a sizable fluctuation in value and thus would either sell or buy index products in great volumes. The significant trading volumes either increase or decrease the demand for the index and thus leads to substantial fluctuations in the value of the index. This example illustrates how expectations of a value change can actually cause a value change, regardless of whether the prediction was correct. Hence, an effective predictive model could be a significant factor in value fluctuations. An additional implication would be the potential to use the predictive model for nefarious purposes. An effective predictive model, as discussed, has the potential to greatly influence investment decisions. The owner of the predictive model can potentially manipulate the output of the model to give a misleading result. This misleading result can cause value changes in the indice that benefit the owner of the predictive model. For example, if the owner of the model was to engage in short selling of an indice (a purchasing option that allows an investor to ‘sell’ a financial product and then buy it back at a later time), the owner would wish for the indice to decrease in price. This would entice the owner to manipulate the model so that it predicts a sudden and dramatic fluctuation in value. Subsequently, investors would be anxious about the predicted higher volatility and they would likely sell their index products to avoid being impacted by the volatility. During this sell off, the owner could sell his share of indices for a profit, as he can ‘buy’ them back at a higher price than he ‘sold’ them. This circumstance illustrates the potential for trusted predictive models to be used for misleading purposes and financial market regulators may need to introduce laws and regulations that prevent such purposes being carried out.

The dataset analysed for this predictive model creation is a merged dataset that contains data pertaining to the four aforementioned indices and COVID-19 cases in Australia, India, The United Kingdom (UK) and The United States of America (US). The clean datasets associated with this project contain the individual datasets for each indice and COVID-19 data for each of the four countries. For the purposes of linear regression modelling, all these datasets were merged together to create one dataset that contained all the information necessary for predictive model creation. The data relating to indices shows for each trading day within the period of the 23rd of September 2019 to the 21st of September 2020 the high value (highest value of the index during the trading day), the low value (lowest value of the index during the trading day) and the closing value (the value of the indice when the market closes) of each index. Data pertaining to the opening values, volume of trades and the total value of trades were missing in the original dataset and are thus not considered for the upcoming analysis. The data pertaining to COVID-19 cases shows the daily new COVID-19 cases for Australia, India, the UK and the US. The data is for the same time period as the indices data: the 23rd of September 2019 to the 21st of September 2020, (excluding weekends which are non-trading days on the Australian Securities Exchange, the market where the indices are traded). Hence, the domain situation is essentially measures of value for the aforementioned indices and daily new COVID-19 cases in Australia, India, the UK and the US over a period of close to one year.

With this merged dataset, a linear regression predictive model was consequently created. This linear regression model takes inputs of daily new COVID-19 cases for each of the four aforementioned countries. Specifically, the inputs are the total new daily COVID-19 cases in the dataset. The outputs of the linear regression model are the daily changes in value. The measure of change in value is the difference between the high and the low value. So, the linear regression model is provided with x-value data of the new COVID-19 cases in a day, and the y-value data of the corresponding change in the value of the index. With this data, the linear regression model will establish measures of correlation between the COVID-19 cases and fluctuations in the value of the indexes. Further, the regression model will predict, for any value of new daily COVID-19 cases, an estimated value of the difference in the high and low data of the index. Prior to the creation of the linear regression model, the hypothesis was that increased cases of COVID-19 would create higher uncertainty in financial markets, leading to increased volatility and fluctuation of the index prices. The results of the linear regression model are below:

Finance Index: A linear regression model produces a linear equation that relates the x-values with estimated y-values. With a linear equation, each type of x-value is correlated to the y-value with a gradient coefficient. The gradient coefficient of each country in relation to value fluctuation is:
Indian Cases: -0.00068713
US Cases: -0.00133703
UK Cases: 0.01984603
Australian Cases: 0.29623891

The measures of accuracy of the regression model are:
R-squared score (proportion of value fluctuation explained by change in COVID-19 cases): 0.053507….
Root Mean Squared Error (difference between value fluctuations estimated and value fluctuations observed): 151.030559….

A sample of x-values (Indian Cases: 20,000, UK Cases: 1000, US Cases: 20,000 and Australian Cases: 300) yielded a predicted fluctuation between the high and low value of 169.

Healthcare Index: The gradient coefficient of each country is:
Indian Cases: -3.5368….e-3
US Cases: -2.1343...e-2
UK Cases: 2.5083...e-1
Australian Cases: 3.9142

The measures of accuracy of the regression model are:
R-squared score: -0.07453…
Root Mean Squared Error: 1455.8949…

The same sample of x-values as for the Finance Index yielded a predicted fluctuation between the high and low value of 1765.

Info Tech Index The gradient coefficient of each country is:
Indian Cases: 0.0001...
UK Cases: 0.0023….
US Cases: -0.0001...
Australian Cases: 0.0499...

The measures of accuracy of the regression model are:
R-squared score: -0.10588
Root Mean Squared Error: 27.4483…

The same sample of x-values as for the Finance Index yielded a predicted fluctuation between the high and low value of 50.

Resources Index: The gradient coefficient of each country is:
Indian Cases: -0.0002….
US Cases: -0.0009...
UK Cases: 0.0155...
Australian Cases: 0.1902

The measures of accuracy of the regression model are:
R-squared score: -0.0240….
Root Mean Squared Error (RMSE): 90.8347

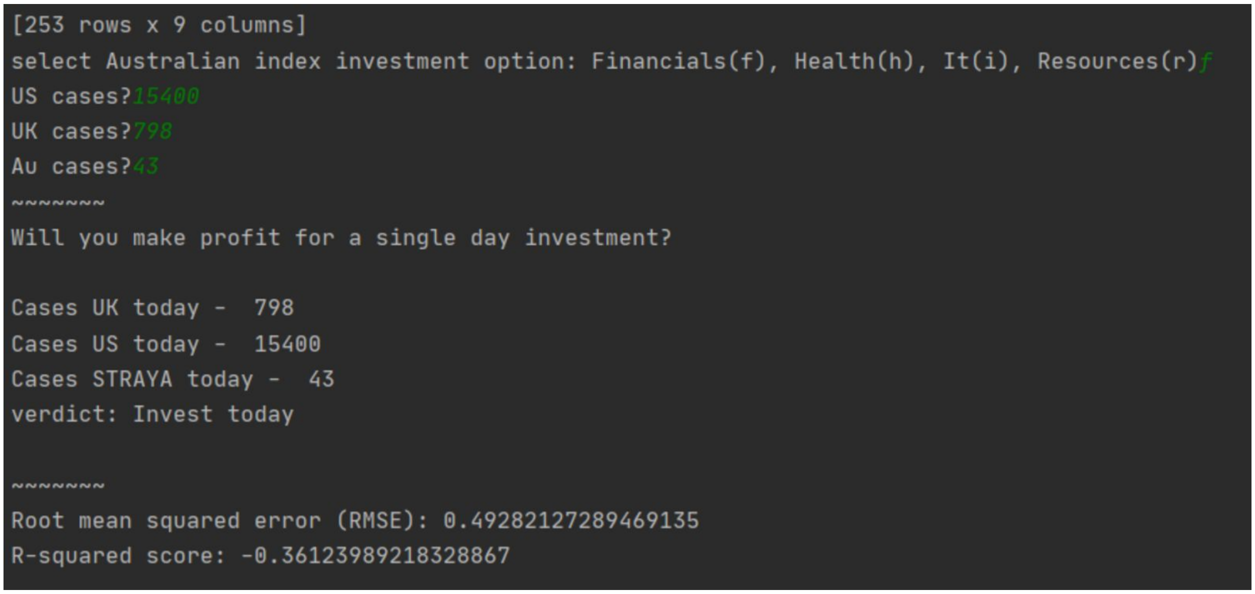
The same sample of x-values as for the Finance Index yielded a predicted fluctuation between the high and low value of 124

The results of fluctuations in value of each Index reveal some commonalities. Firstly, it appears to be that Indian and US cases are inversely related to volatility for each index, that is, the rise in Indian and US cases causes a decrease in the difference between the high and the low value. This is due to each gradient coefficient being negative. Conversely, the positive coefficients for UK and Australian Cases convey that increases in COVID-19 cases for these countries are linked to increases in fluctuation. Hence, the predictive model suggests fluctuation is decreased with US and Indian COVID-19 case rise, but increased with UK and Australian COVID-19 case increases. This both supports and discredits the prior hypothesis that increases in COVID-19 cases result in higher volatility. Another trend is that relative to UK and Australian Cases, the correlation between Indian and US cases with value fluctuations is to a lesser degree. This is due to the magnitude of coefficients for Indian and US cases being smaller than UK and Australian Cases. Consequently, Indian and US COVID-19 cases have a less significant impact on fluctuation than Australian and UK cases. Another trend is that Australian COVID-19 cases have the strongest correlation with index fluctuations. This is due to their coefficients having the largest values relatively. As the Indexes in question are traded in Australian markets, the fact that Australian COVID-19 cases have the greatest impact is unsurprising.

The linear regression predictive model however, is ultimately unable to make effective predictions of fluctuations in value, based on COVID-19 cases. Values equal to zero for the RMSE mean that there is no error difference between estimated and observed values. Values close to zero for the RMSE indicate only a minimal error. With none of the RMSE values being close to zero, it can be concluded that there is a high degree of error in the predictive model. Further, the results of the r-squared value indicate a flawed model. An r-squared value of 1 indicates that a change in fluctuation is completely a result of changes in COVID-19. A value close to 1 indicates a strong connection. A value closer to zero indicates a weak connection. A negative value indicates that a horizontal line would account for a greater variation of proportion than the existing fit. This means that the model contains data that doesn’t help to predict the response. For the Finance Index, the r-squared value suggests that only roughly 5% of the variation in fluctuation is a result of the COVID-119 case data. For the other indexes, they all have negative r-squared values, that as mentioned, undermine the accuracy of the predictive model to a great degree. Ultimately, while predictions can be made, and the observations of these predictions validate hypotheses or expected outcomes, the measures of accuracy render the predictive model as a model that cannot make good predictions.

K-nearest Neighbour regression

A further exploration of the question concerning the impact of covid cases on financial markets was undertaken of the merged dataset of Coivd-19 cases and stock index prices in the form of a k-nearest neighbours regression model. This predictive model takes inputs of daily new COVID-19 cases for Australia, the US or the UK and compares this to the historic relative daily change of the 4 index prices, Financial, Healthcare, IT and Resources. As such, the model predicts whether the stock price will increase of decrease based on similarity between the number of new Covid-19 cases, and how much the stock price historically moved on days with similar Covid-19 cases. The outputs of the model then instructs the user or automated trading machine whether they should buy a certain index stock or not. If the model predicts a greater then 1% increase in stock price, based on the number of new Covid cases that day, it will output ‘invest today’. If the model predicts a smaller then 1% increase or a decrease in the stock price it will output “not today”. Prior to the creation of the k-nearest neighbours model, the hypothesis was that a larger number of cases of COVID-19 would lead to a greater decrease of an index price. The results of the k-nearest neighbour regression model are below:



```
[253 rows x 9 columns]
select Australian index investment option: Financials(f), Health(h), It(i), Resources(r)f
US cases?15400
UK cases?798
Au cases?43
~~~~~~~
Will you make profit for a single day investment?

Cases UK today -  798
Cases US today -  15400
Cases STRAYA today -  43
verdict: Invest today

~~~~~~~
Root mean squared error (RMSE): 0.49282127289469135
R-squared score: -0.36123989218328867
```

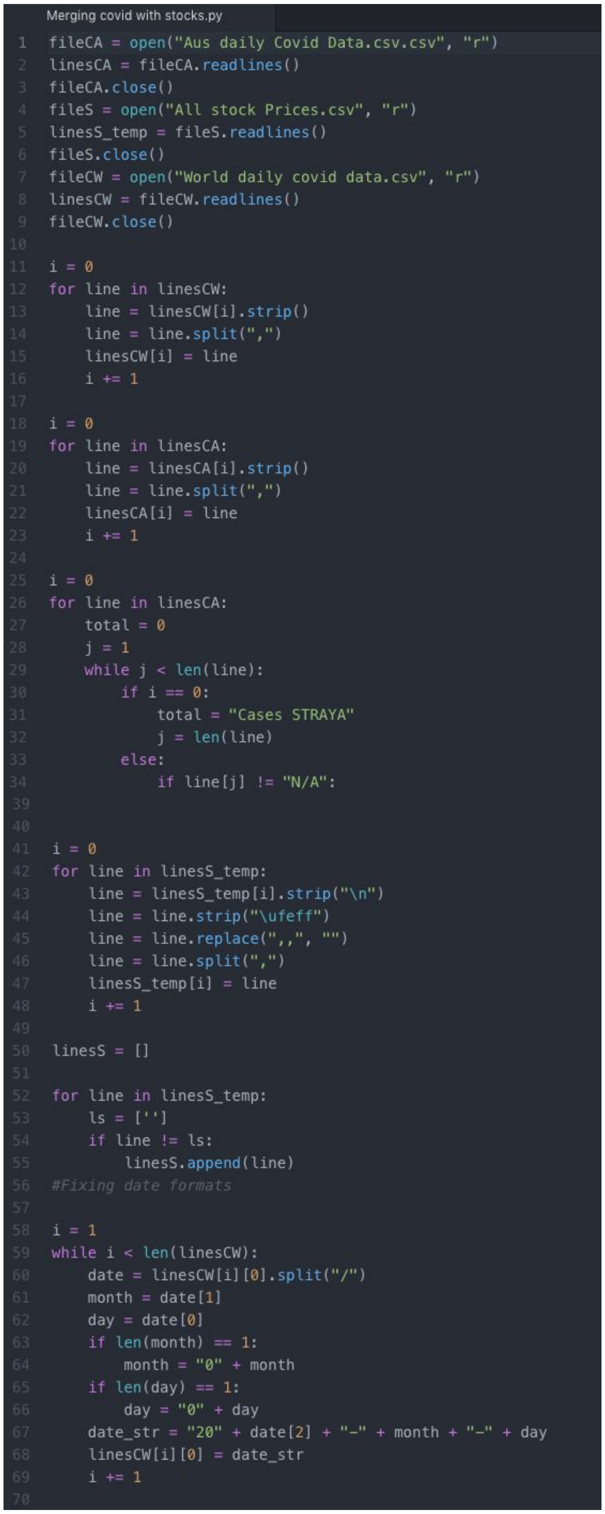
Effective prediction by this model could have large impacts in the financial and trading world. If the k-nearest neighbour regression model could effectively predict movements of stock prices based on new Covid cases, speculative stock traders would be doing everything in their power to obtain this technology as being able to predict the movement of stock prices would have a massive impact on the financial world.

Using the example above the model reveals that for the financial index if there are 15400 US cases 798 UK cases and 43 Australian cases the model predicts a greater then 1% rise in the stock price and subsequently informs the user / computer to “invest today”. The Root mean squared error and R-squared score then reveals the accuracy of the prediction. An RMSE score of 0 representing a prediction which is “free from error” and a r-squared score of 1 representing complete correlation between changes in COVID-19 cases and changes in the value of the index. The values of -0.3612… and 0.492821 for r-squared score and the RMSE value show that the prediction model is not 100% accurate and not free from error in this domain.

Part 2

Linear Regression

To create the linear regression model, as discussed prior, all the index and COVID-19 datasets needed to be merged into the one dataset. The coding and textual explanation for how this was undertaken and achieved is as such:



```python
1   fileCA = open("Aus daily Covid Data.csv.csv", "r")
2   linesCA = fileCA.readlines()
3   fileCA.close()
4   fileS = open("All stock Prices.csv", "r")
5   linesS_temp = fileS.readlines()
6   fileS.close()
7   fileCW = open("World daily covid data.csv", "r")
8   linesCW = fileCW.readlines()
9   fileCW.close()
10
11  i = 0
12  for line in linesCW:
13      line = linesCW[i].strip()
14      line = line.split(",")
15      linesCW[i] = line
16      i += 1
17
18  i = 0
19  for line in linesCA:
20      line = linesCA[i].strip()
21      line = line.split(",")
22      linesCA[i] = line
23      i += 1
24
25  i = 0
26  for line in linesCA:
27      total = 0
28      j = 1
29      while j < len(line):
30          if i == 0:
31              total = "Cases STRAYA"
32              j = len(line)
33          else:
34              if line[j] != "N/A":
39
40
41  i = 0
42  for line in linesS_temp:
43      line = linesS_temp[i].strip("\n")
44      line = line.strip("\ufeff")
45      line = line.replace(",,", "")
46      line = line.split(",")
47      linesS_temp[i] = line
48      i += 1
49
50  linesS = []
51
52  for line in linesS_temp:
53      ls = ['']
54      if line != ls:
55          linesS.append(line)
56  #Fixing date formats
57
58  i = 1
59  while i < len(linesCW):
60      date = linesCW[i][0].split("/")
61      month = date[1]
62      day = date[0]
63      if len(month) == 1:
64          month = "0" + month
65      if len(day) == 1:
66          day = "0" + day
67      date_str = "20" + date[2] + "-" + month + "-" + day
68      linesCW[i][0] = date_str
69      i += 1
70
```

Lines 1-10: Opens files Australian covid data, all stock prices, and world daily covid data and reads each line into a list.

Lines 11-24: Strips unwanted information in the csv files and splits the data values into a nested list, for all of the covid data.

Lines 25-38: Combines all of the Australian states covid data to find the total number of covid cases per day in Australia. Deals with missing values.

Lines 41-49: Strips unwanted information from the stock prices csv file. Also strips empty values. Then splits the csv data into a nested list.

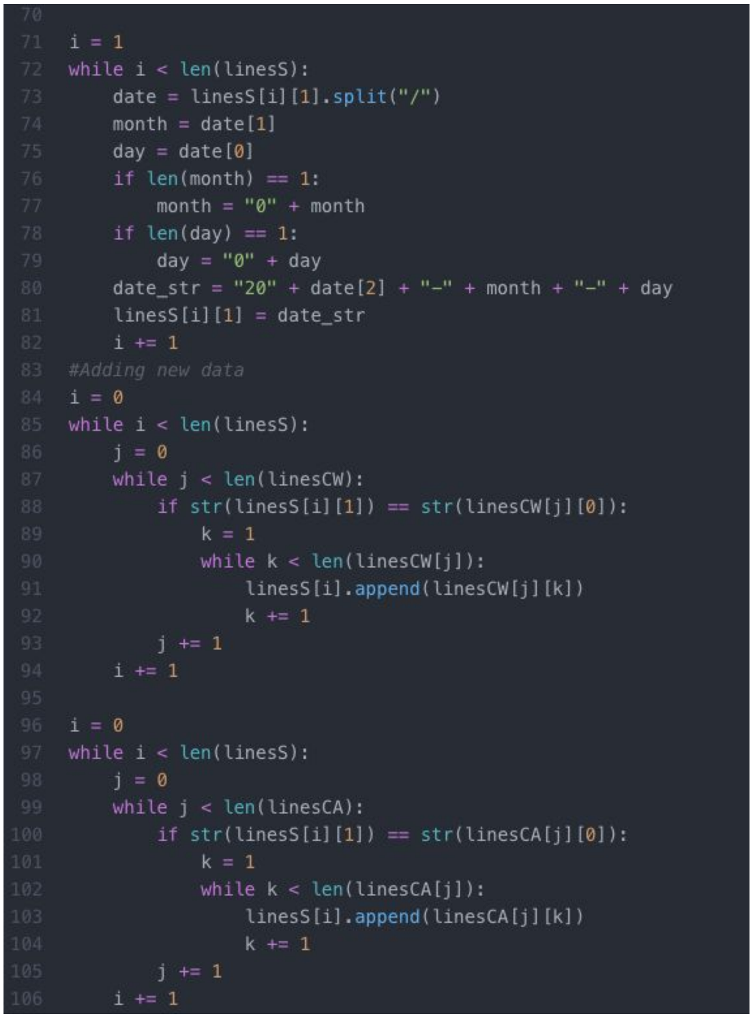
Lines 52-56: Removes empty lines of data from the stock prices.

Lines 59-70: Reformats date of world covid data into ISO8601 standard.

Lines 71-82: Reformats date of world covid data into ISO8601 standard.

Lines 83-95: Adds world covid data to the stock prices, with the corresponding day. Removes duplicate dates.

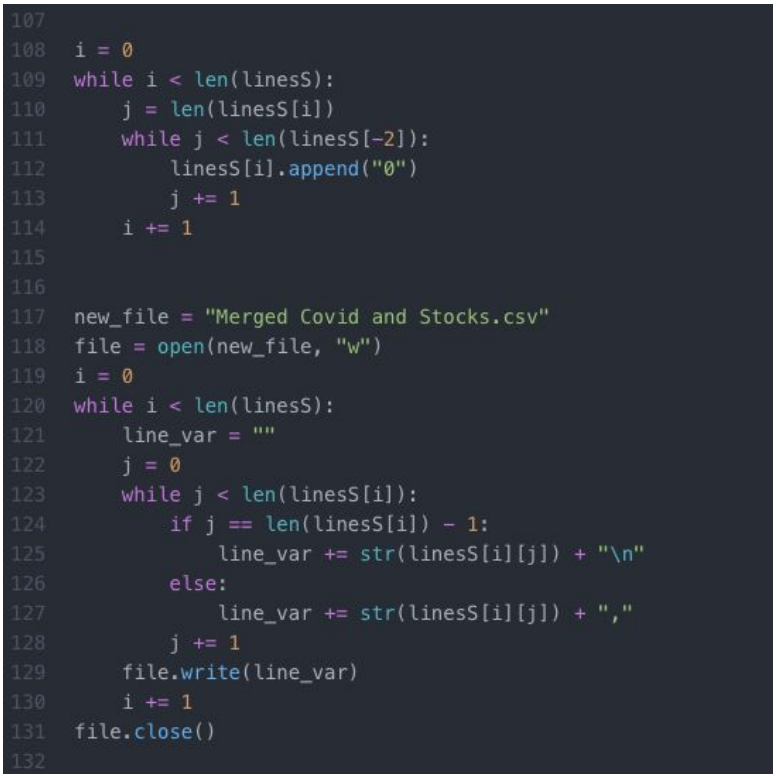
Lines 97-106: Adds Australian covid data to the stock prices, with the corresponding day. Removes duplicate dates.



```
70
71  i = 1
72  while i < len(linesS):
73      date = linesS[i][1].split("/")
74      month = date[1]
75      day = date[0]
76      if len(month) == 1:
77          month = "0" + month
78      if len(day) == 1:
79          day = "0" + day
80      date_str = "20" + date[2] + "-" + month + "-" + day
81      linesS[i][1] = date_str
82      i += 1
83  #Adding new data
84  i = 0
85  while i < len(linesS):
86      j = 0
87      while j < len(linesCW):
88          if str(linesS[i][1]) == str(linesCW[j][0]):
89              k = 1
90              while k < len(linesCW[j]):
91                  linesS[i].append(linesCW[j][k])
92                  k += 1
93          j += 1
94      i += 1
95
96  i = 0
97  while i < len(linesS):
98      j = 0
99      while j < len(linesCA):
100         if str(linesS[i][1]) == str(linesCA[j][0]):
101             k = 1
102             while k < len(linesCA[j]):
103                 linesS[i].append(linesCA[j][k])
104                 k += 1
105         j += 1
106     i += 1
```

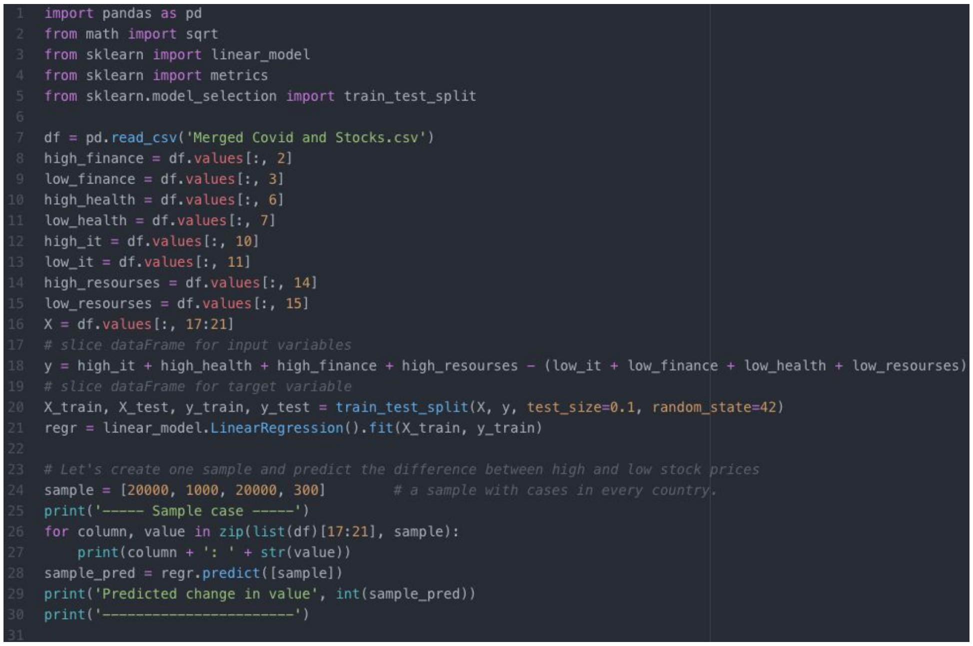
Lines 108-115: Fills in zeros for the days where we have data for index prices, but not for covid data.

Lines 117-131: Writes on all data onto a new file called “Merged Covid and Stocks.csv”.



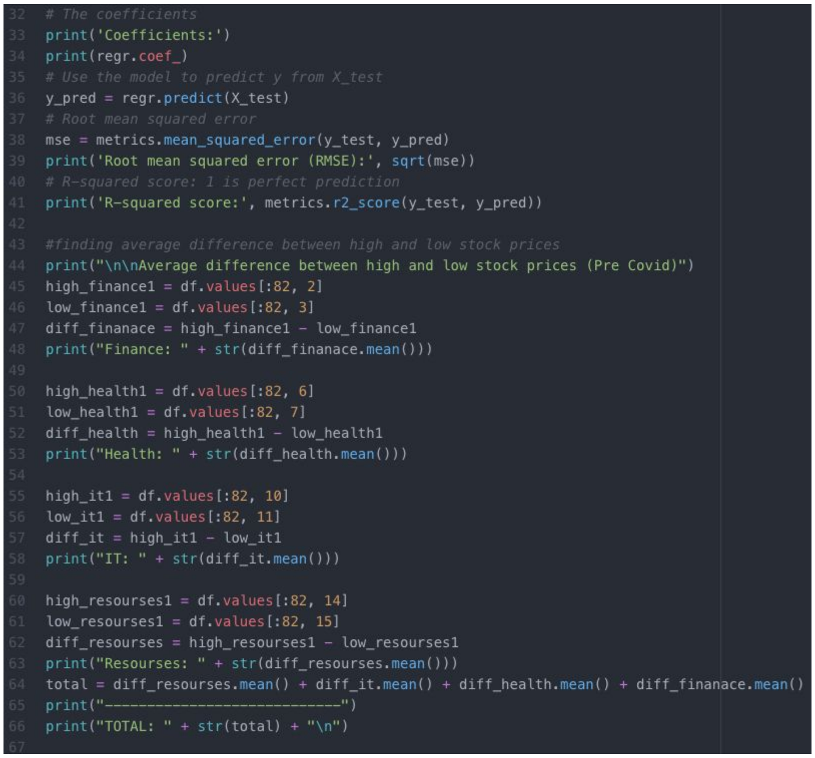
```
107
108 i = 0
109 while i < len(linesS):
110     j = len(linesS[i])
111     while j < len(linesS[-2]):
112         linesS[i].append("0")
113         j += 1
114     i += 1
115
116
117 new_file = "Merged Covid and Stocks.csv"
118 file = open(new_file, "w")
119 i = 0
120 while i < len(linesS):
121     line_var = ""
122     j = 0
123     while j < len(linesS[i]):
124         if j == len(linesS[i]) - 1:
125             line_var += str(linesS[i][j]) + "\n"
126         else:
127             line_var += str(linesS[i][j]) + ","
128         j += 1
129     file.write(line_var)
130     i += 1
131 file.close()
132
```

The creation of the linear regression predictive model and the evaluation of the quality of the model was undertaken using the Python coding language and the libraries: math, pandas and scikit-lean. The coding and textual explanation is as follows:



```
1   import pandas as pd
2   from math import sqrt
3   from sklearn import linear_model
4   from sklearn import metrics
5   from sklearn.model_selection import train_test_split
6
7   df = pd.read_csv('Merged Covid and Stocks.csv')
8   high_finance = df.values[:, 2]
9   low_finance = df.values[:, 3]
10  high_health = df.values[:, 6]
11  low_health = df.values[:, 7]
12  high_it = df.values[:, 10]
13  low_it = df.values[:, 11]
14  high_resourses = df.values[:, 14]
15  low_resourses = df.values[:, 15]
16  X = df.values[:, 17:21]
17  # slice dataFrame for input variables
18  y = high_it + high_health + high_finance + high_resourses - (low_it + low_finance + low_health + low_resourses)
19  # slice dataFrame for target variable
20  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
21  regr = linear_model.LinearRegression().fit(X_train, y_train)
22
23  # Let's create one sample and predict the difference between high and low stock prices
24  sample = [20000, 1000, 20000, 300]       # a sample with cases in every country.
25  print('----- Sample case -----')
26  for column, value in zip(list(df)[17:21], sample):
27      print(column + ': ' + str(value))
28  sample_pred = regr.predict([sample])
29  print('Predicted change in value', int(sample_pred))
30  print('-----------------------')
31
```

Lines 1-6: Imports all the modules needed for the machine learning code
Lines 7-16: Separating data into specific dataframes, for each specific data type.
Lines 17-18: Adds difference between all high and low index prices. This is necessary for a predictive model that predicts the sum variation for all four indexes. The model that was used for this project, examined each index separately. As such, for separate index analysis, the y-value is only the difference between the high and the low value for one index. A predictive model for the sum of the index variations is described in order to avoid having to explain the code for the predictive model for each index.



```
32  # The coefficients
33  print('Coefficients:')
34  print(regr.coef_)
35  # Use the model to predict y from X_test
36  y_pred = regr.predict(X_test)
37  # Root mean squared error
38  mse = metrics.mean_squared_error(y_test, y_pred)
39  print('Root mean squared error (RMSE):', sqrt(mse))
40  # R-squared score: 1 is perfect prediction
41  print('R-squared score:', metrics.r2_score(y_test, y_pred))
42
43  #finding average difference between high and low stock prices
44  print("\n\nAverage difference between high and low stock prices (Pre Covid)")
45  high_finance1 = df.values[:82, 2]
46  low_finance1 = df.values[:82, 3]
47  diff_finanace = high_finance1 - low_finance1
48  print("Finance: " + str(diff_finanace.mean()))
49
50  high_health1 = df.values[:82, 6]
51  low_health1 = df.values[:82, 7]
52  diff_health = high_health1 - low_health1
53  print("Health: " + str(diff_health.mean()))
54
55  high_it1 = df.values[:82, 10]
56  low_it1 = df.values[:82, 11]
57  diff_it = high_it1 - low_it1
58  print("IT: " + str(diff_it.mean()))
59
60  high_resourses1 = df.values[:82, 14]
61  low_resourses1 = df.values[:82, 15]
62  diff_resourses = high_resourses1 - low_resourses1
63  print("Resourses: " + str(diff_resourses.mean()))
64  total = diff_resourses.mean() + diff_it.mean() + diff_health.mean() + diff_finanace.mean()
65  print("----------------------------")
66  print("TOTAL: " + str(total) + "\n")
67
```

Lines 19-21: Creates a regression model, based off 90% of data. This 90% of the data is used to train and create the predictive model. The test_size = 0.1 is responsible for allocating 10% of the data for accuracy testing purposes and the random_state = 42 is responsible for creating a seed that makes the random sampling process of data behave the same across multiple runs.
Lines 22-31: Defines a sample set of daily covid cases, and organises and prints it as a readable format. Then prints prediction of the difference between high and low index prices based on that sample set of data.
Lines 32-34: Prints regression coefficients.
Lines 35-42: Calculates the root mean square error and R-squared score of the prediction, based off of the testing data. Then prints out these values.

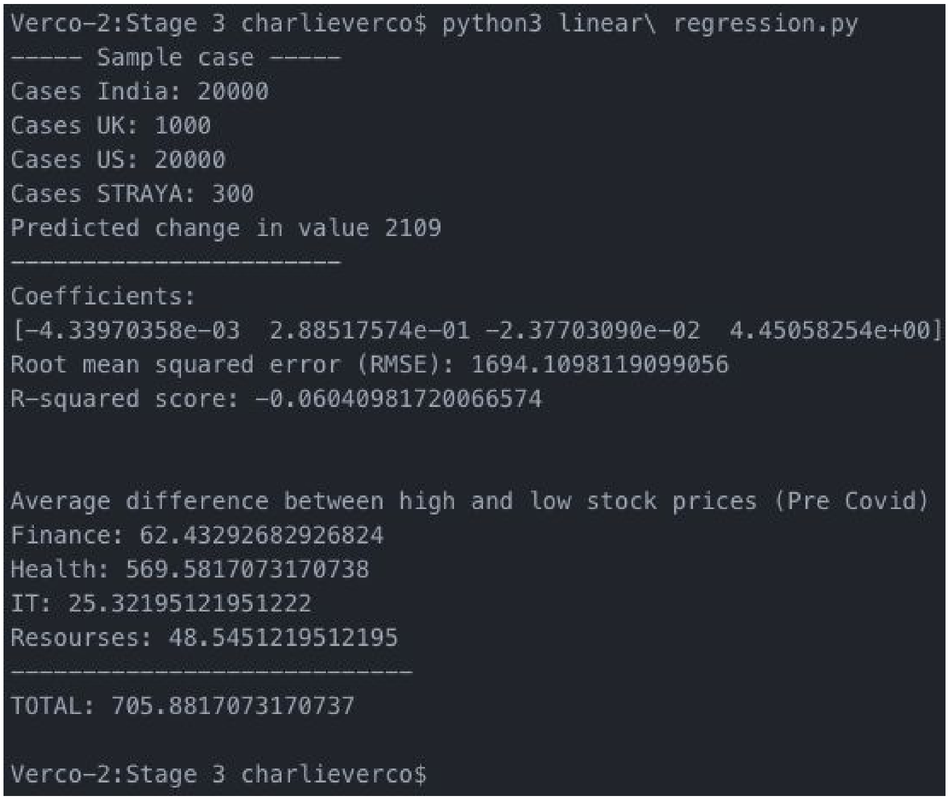
Lines 43-63: Calculates the average difference between the high and low daily stock prices for each index, for before there were any Coronavirus cases. From here the predicted difference between high and low stock price can be compared to when there were zero covid cases. While this isn’t an actual attribute of the predictive model, knowing the average difference prior to COVID-19 can provide some further potential valuable insight into how COVID-19 has affected the variations in value

Lines 64-67: Finds total difference between the high and low daily stock prices for each index, for before there were any Coronavirus cases. This is to be compared to the predicted value of the sum variation for all of the indexes.

This concludes the textual explanation of the coding. Important to clarify is that the above code is not the actual code for the linear regression model studied in part 1. The code above summates the individual high-low variations for each index. This summation gives a total variation for all the indexes, meaning that the model predicts the summed variation for each index. For this report, the linear regression model studied was a model that predicts the variations in value for a single index. As such, this linear regression model is slightly different to the linear regression model code above. The only difference is that for the part 1 model, the y-value is only the difference of high and low value for a single index, not the summation of the difference for multiple indexes. Thus, the above code still explains the process for creating the actual predictive model, but the clarification of the minor difference is needed to avoid confusion as to whether the model is predicting total variation or only variation for one index.

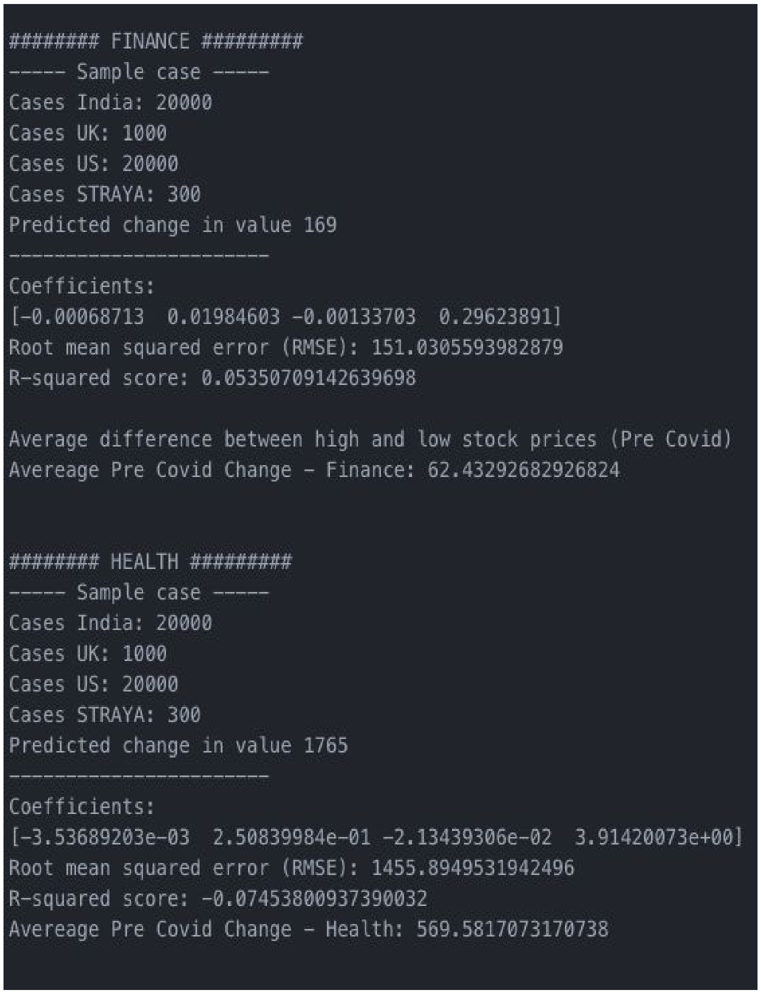
The output of the above code is depicted:

Here the sample case data has been chosen to replicate the data from about 1-2 weeks after an outbreak of Coronavirus cases in each respective country (India, UK, US and STRAYA (Australia). The predicted total difference between the high and low index prices for the was $2109. The coefficients for the regression lines, the root mean square error, and the R-squared score all follow.
Then the average difference between the high and low index prices for before Covid hit can be compared to the predicted value. As expected the predicted value is greater than the average total difference.

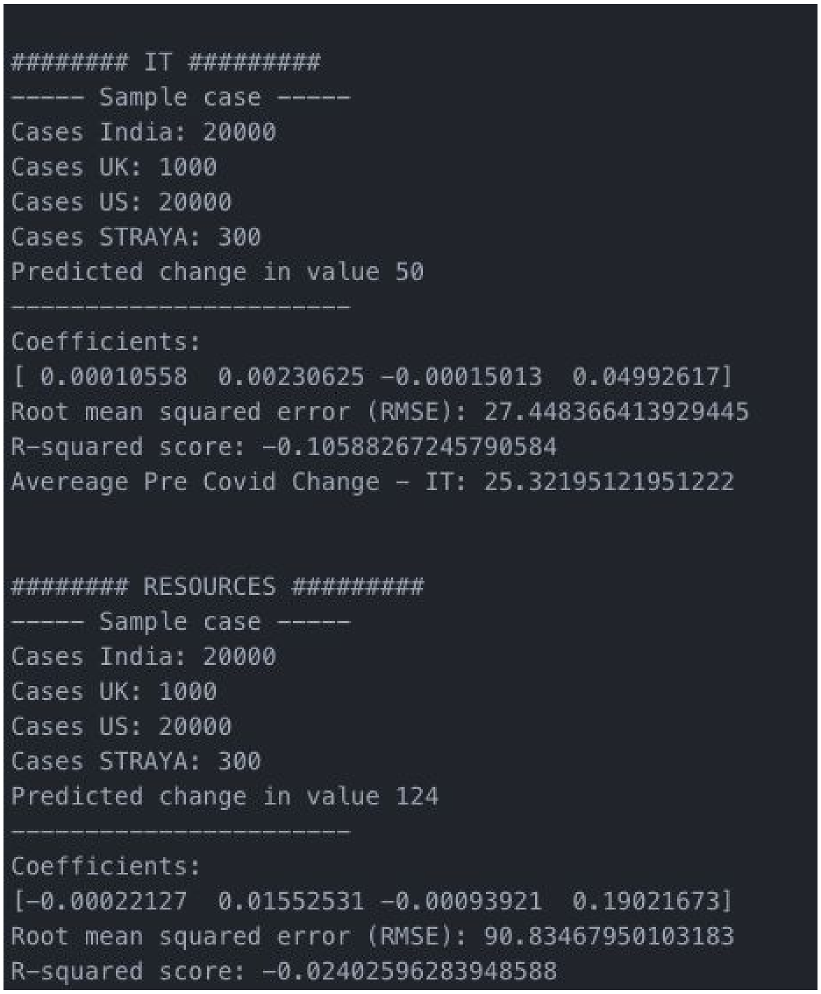


```
Verco-2:Stage 3 charlieverco$ python3 linear\ regression.py
----- Sample case -----
Cases India: 20000
Cases UK: 1000
Cases US: 20000
Cases STRAYA: 300
Predicted change in value 2109
-----------------------
Coefficients:
[-4.33970358e-03  2.88517574e-01 -2.37703090e-02  4.45058254e+00]
Root mean squared error (RMSE): 1694.1098119099056
R-squared score: -0.06040981720066574


Average difference between high and low stock prices (Pre Covid)
Finance: 62.43292682926824
Health: 569.5817073170738
IT: 25.32195121951222
Resourses: 48.5451219512195
----------------------------
TOTAL: 705.8817073170737

Verco-2:Stage 3 charlieverco$
```

The output of the above code, altered only slightly (by setting the ‘y’ variable as the high-low difference of only one index), produces the predicted high-low fluctuation differences for a single index. Altering which high-low difference is given to the ‘y’ variable allows for the predicted values for all four indexes to be outputted. This output is shown below.



```
######## FINANCE #########
----- Sample case -----
Cases India: 20000
Cases UK: 1000
Cases US: 20000
Cases STRAYA: 300
Predicted change in value 169
-----------------------
Coefficients:
[-0.00068713  0.01984603 -0.00133703  0.29623891]
Root mean squared error (RMSE): 151.0305593982879
R-squared score: 0.05350709142639698

Average difference between high and low stock prices (Pre Covid)
Avereage Pre Covid Change - Finance: 62.43292682926824


######## HEALTH #########
----- Sample case -----
Cases India: 20000
Cases UK: 1000
Cases US: 20000
Cases STRAYA: 300
Predicted change in value 1765
-----------------------
Coefficients:
[-3.53689203e-03  2.50839984e-01 -2.13439306e-02  3.91420073e+00]
Root mean squared error (RMSE): 1455.8949531942496
R-squared score: -0.07453800937390032
Avereage Pre Covid Change - Health: 569.5817073170738
```



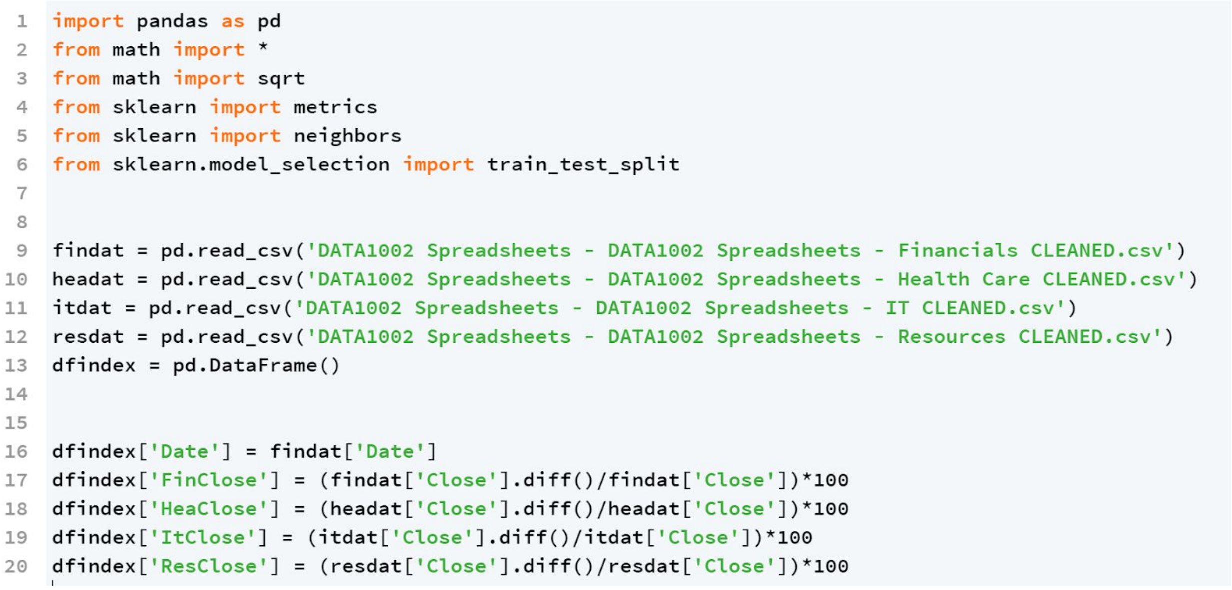
```
######## IT #########
----- Sample case -----
Cases India: 20000
Cases UK: 1000
Cases US: 20000
Cases STRAYA: 300
Predicted change in value 50
-----------------------
Coefficients:
[ 0.00010558  0.00230625 -0.00015013  0.04992617]
Root mean squared error (RMSE): 27.448366413929445
R-squared score: -0.10588267245790584
Avereage Pre Covid Change - IT: 25.32195121951222


######## RESOURCES #########
----- Sample case -----
Cases India: 20000
Cases UK: 1000
Cases US: 20000
Cases STRAYA: 300
Predicted change in value 124
-----------------------
Coefficients:
[-0.00022127  0.01552531 -0.00093921  0.19021673]
Root mean squared error (RMSE): 90.83467950103183
R-squared score: -0.02402596283948588
```

The linear regression model was deemed appropriate and developed for this report simply because it was an appropriate predictive model to answer the question, the question being whether COVID-19 cases impact value fluctuation in indexes. This report wished to explore the impact of numerical changes in COVID-19 cases on the numerical change in fluctuation of value. A linear regression model derives a predicted linear relationship between numerical input data and numerical output data. For any input value, it provides a predicted output value and so specifically, it has provided a predicted numerical fluctuation value for any numerical value of COVID-19 cases. This type of predictive model achieves exactly what was desired with the question and as such, the choice was made to train and develop a linear regression model. The training of the data was undertaken on 90% of the data supplied to the linear regression model. 10% of the data was set aside for the purposes of testing whether the training of the data had resulted in a model that could effectively predict the fluctuations in high-low value, dependent on COVID-19 cases. This process of splitting the data for training and testing purposes was undertaken simplistically using the function ‘train_test_split’ from the pandas library. Then the approach to training the model involved applying the fit() method to the training data and then applying this method and its argument to the ‘LinearRegression’ object that is provided by the scikit-learn python library. This ‘LinearRegression’ method is what is ultimately responsible for training the linear regression model. Due to the simplicity of this approach, it can be achieved in only 2 lines of code, this is why this particular approach was chosen.

Evaluation of the model was undertaken through the development and outputting of measures of accuracy. The RMSE and the r-squared values were two measures of accuracy outputted to allow for an evaluation of the effectiveness of the model. The RMSE is a mathematically derived value that indicates how accurate a predicted value is in relation to the actual observed value, and the r-squared value is a mathematically derived value that indicates how much of the change in the predicted data is as a result of a change in the input data. As such, these values can provide a legitimate and valuable mathematically based evaluation of how accurate or effective the predictive model is. This is why the decision was made to produce these values. For this particular linear regression model, all the RMSE values were large (exact values are located in the pictures above), indicating a high degree of error and that the predicted fluctuations values were a significant distance away from actual observed values of fluctuation. Hence the prediction was different to reality. All the r-squared values (located in the above pictures were either negative or very close to zero, indicating that the values for COVID-19 had either very little or no impact on the change in the value fluctuations of each index. Ultimately, these measures of accuracy/effectiveness have allowed the evaluation to be made that the predictive model has significant limitations.

K-nearest Neighbour regression

The creation of the K-nearest Neighbours regression model and evaluation of it was undertaken using the python coding language and its libraries of: pandas, maths and scikit-learn. The code for it is included below, accompanied by textual explanation:



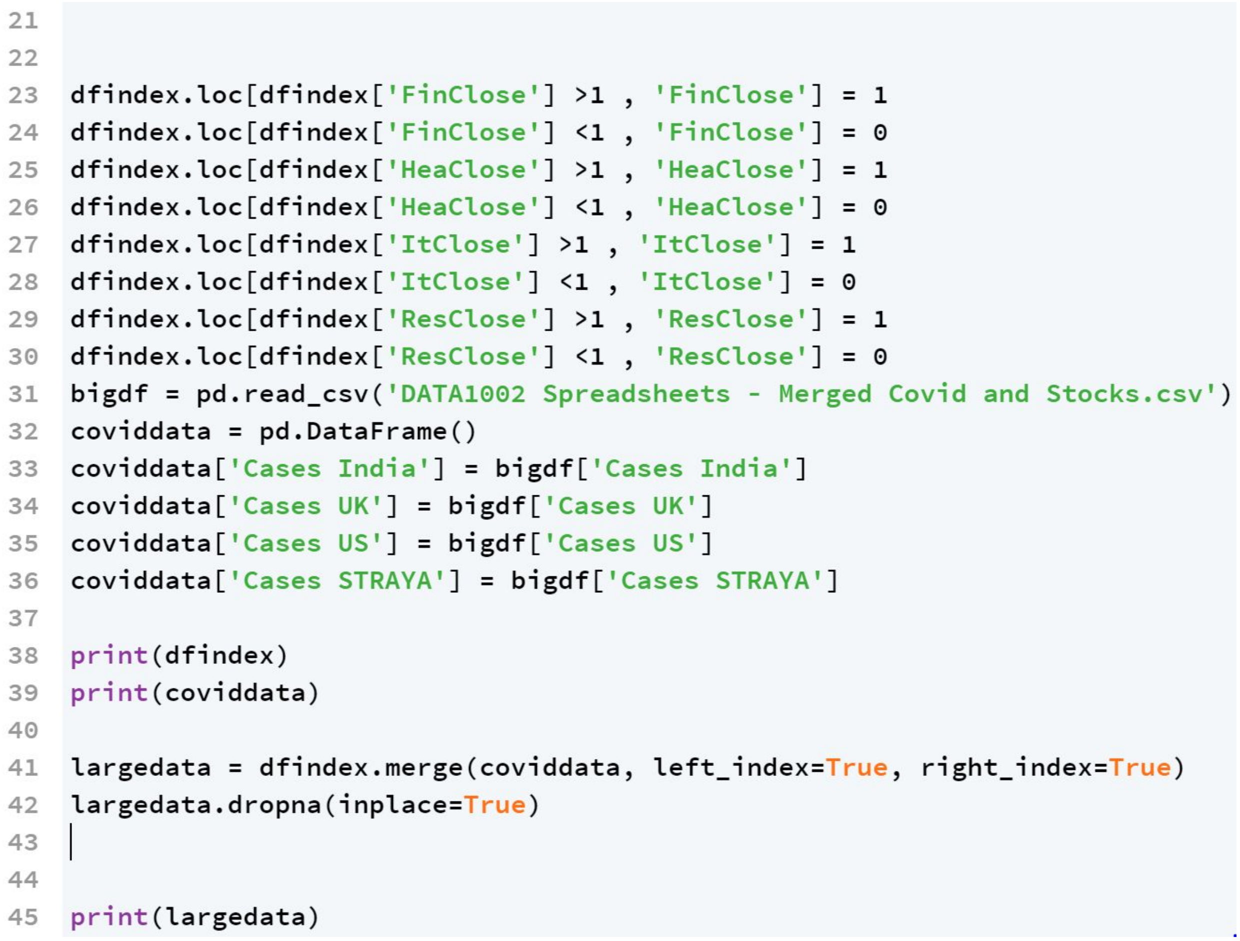
```
 1  import pandas as pd
 2  from math import *
 3  from math import sqrt
 4  from sklearn import metrics
 5  from sklearn import neighbors
 6  from sklearn.model_selection import train_test_split
 7
 8
 9  findat = pd.read_csv('DATA1002 Spreadsheets - DATA1002 Spreadsheets - Financials CLEANED.csv')
10  headat = pd.read_csv('DATA1002 Spreadsheets - DATA1002 Spreadsheets - Health Care CLEANED.csv')
11  itdat = pd.read_csv('DATA1002 Spreadsheets - DATA1002 Spreadsheets - IT CLEANED.csv')
12  resdat = pd.read_csv('DATA1002 Spreadsheets - DATA1002 Spreadsheets - Resources CLEANED.csv')
13  dfindex = pd.DataFrame()
14
15
16  dfindex['Date'] = findat['Date']
17  dfindex['FinClose'] = (findat['Close'].diff()/findat['Close'])*100
18  dfindex['HeaClose'] = (headat['Close'].diff()/headat['Close'])*100
19  dfindex['ItClose'] = (itdat['Close'].diff()/itdat['Close'])*100
20  dfindex['ResClose'] = (resdat['Close'].diff()/resdat['Close'])*100
```

Lines 1-8: Imports all the modules needed for the machine learning code

Lines 9-15: Reads and defines the 4 index spreadsheets which contain our data and creates/define a dataframe

Lines 16: Merges our Covid data with our stick index data for easy use in Machine Learning.

Lines 17-20: Takes the current index stock price percentage, divides this by the previous days stock price multiplied by 100 to find the percentage daily change in each index's stock price.



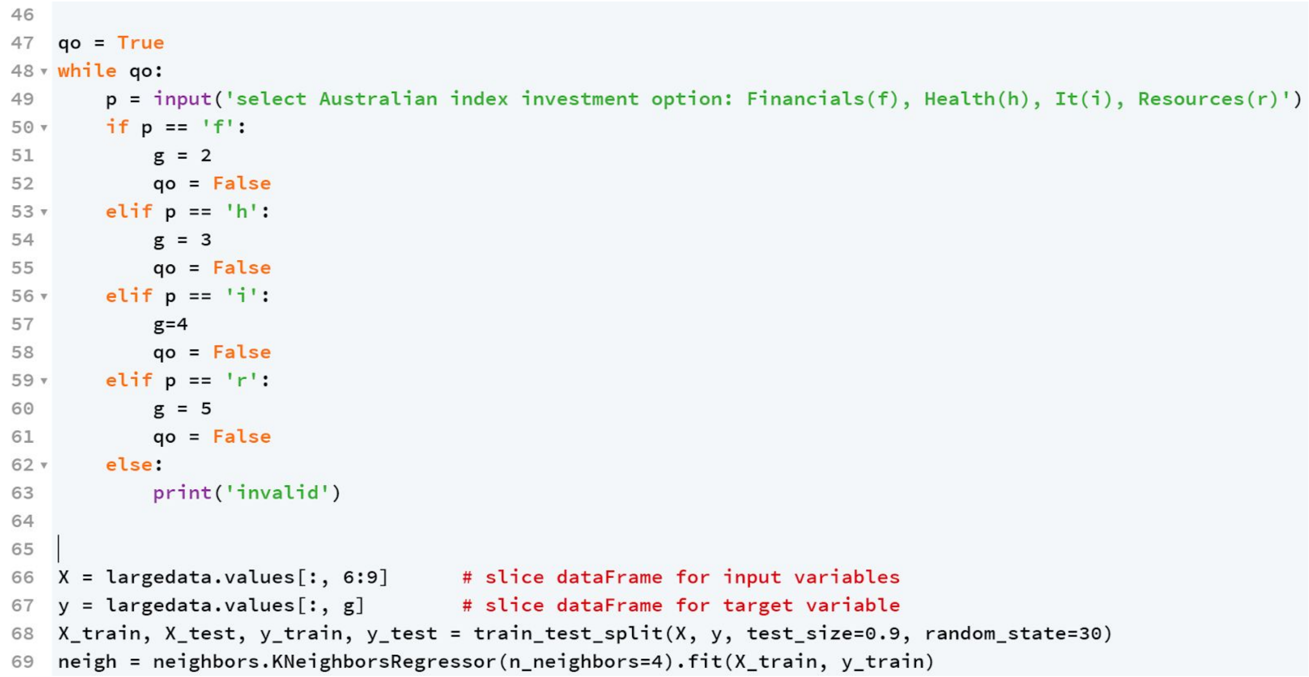
```
21
22
23  dfindex.loc[dfindex['FinClose'] >1 , 'FinClose'] = 1
24  dfindex.loc[dfindex['FinClose'] <1 , 'FinClose'] = 0
25  dfindex.loc[dfindex['HeaClose'] >1 , 'HeaClose'] = 1
26  dfindex.loc[dfindex['HeaClose'] <1 , 'HeaClose'] = 0
27  dfindex.loc[dfindex['ItClose'] >1 , 'ItClose'] = 1
28  dfindex.loc[dfindex['ItClose'] <1 , 'ItClose'] = 0
29  dfindex.loc[dfindex['ResClose'] >1 , 'ResClose'] = 1
30  dfindex.loc[dfindex['ResClose'] <1 , 'ResClose'] = 0
31  bigdf = pd.read_csv('DATA1002 Spreadsheets - Merged Covid and Stocks.csv')
32  coviddata = pd.DataFrame()
33  coviddata['Cases India'] = bigdf['Cases India']
34  coviddata['Cases UK'] = bigdf['Cases UK']
35  coviddata['Cases US'] = bigdf['Cases US']
36  coviddata['Cases STRAYA'] = bigdf['Cases STRAYA']
37
38  print(dfindex)
39  print(coviddata)
40
41  largedata = dfindex.merge(coviddata, left_index=True, right_index=True)
42  largedata.dropna(inplace=True)
43
44
45  print(largedata)
```

Lines 21-30: If the daily change in the stock price is greater than a 1% increase, we define that it is worth investing on that day. If it is worth investing in that index on that day the values will be set at 1 and if it is not worth investing the values will be set at 0. This was done to aid in the computation of the k-nearest neighbours algorithm.

Lines 31-36: Reads and defines the Covid data for the 4 countries India, Uk, US and Australia and creates a new dataframe.

Lines 37-39: Prints out the values within each respective dataframe

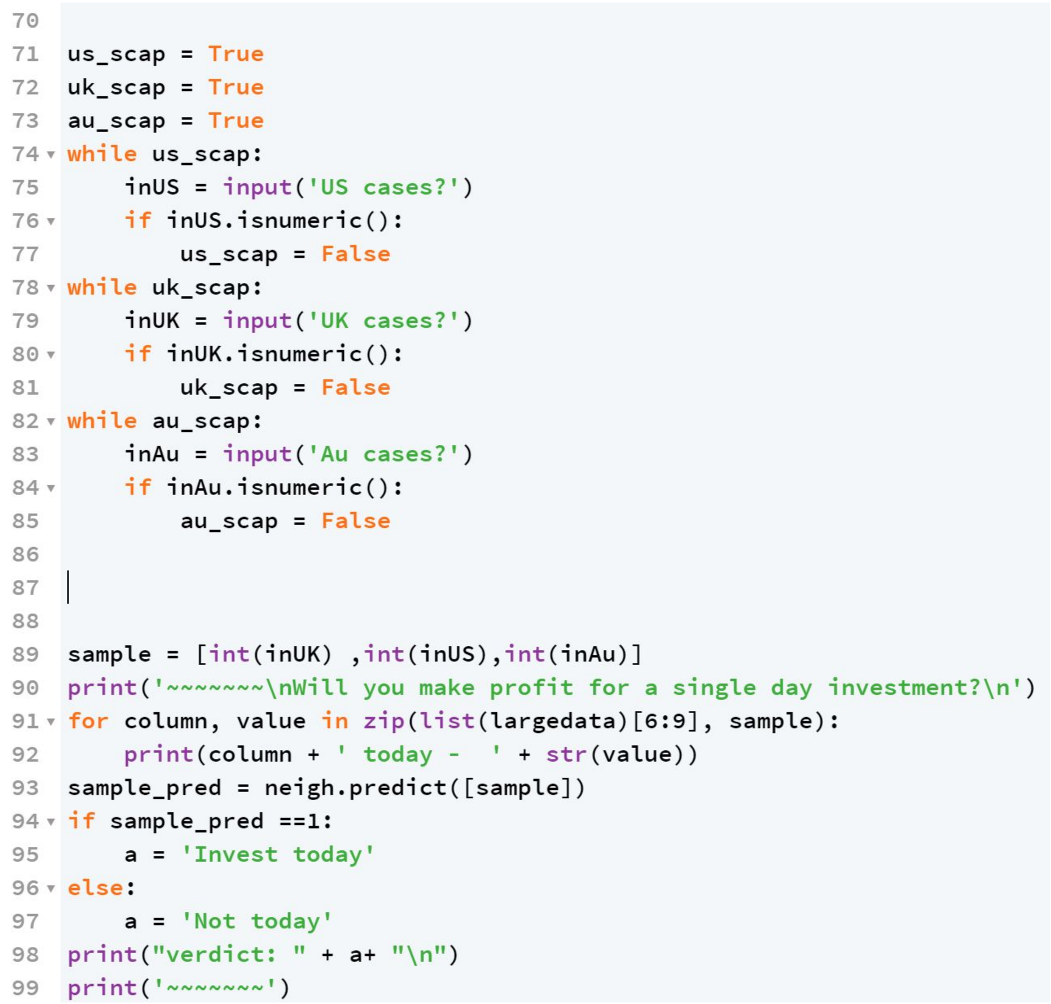
Lines 40-45: Merges the two dataframes together and double checks that no NA values were created during the merge. Then finally prints out the merged dataframe values.



```
46
47  qo = True
48  while qo:
49      p = input('select Australian index investment option: Financials(f), Health(h), It(i), Resources(r)')
50      if p == 'f':
51          g = 2
52          qo = False
53      elif p == 'h':
54          g = 3
55          qo = False
56      elif p == 'i':
57          g=4
58          qo = False
59      elif p == 'r':
60          g = 5
61          qo = False
62      else:
63          print('invalid')
64
65
66  X = largedata.values[:, 6:9]      # slice dataFrame for input variables
67  y = largedata.values[:, g]        # slice dataFrame for target variable
68  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.9, random_state=30)
69  neigh = neighbors.KNeighborsRegressor(n_neighbors=4).fit(X_train, y_train)
```

Lines 46-63: An input loop is built so that the index could be chosen by the user/The automated trading machine. This input is then converted into a value that can reference the respective column in the code for the training.

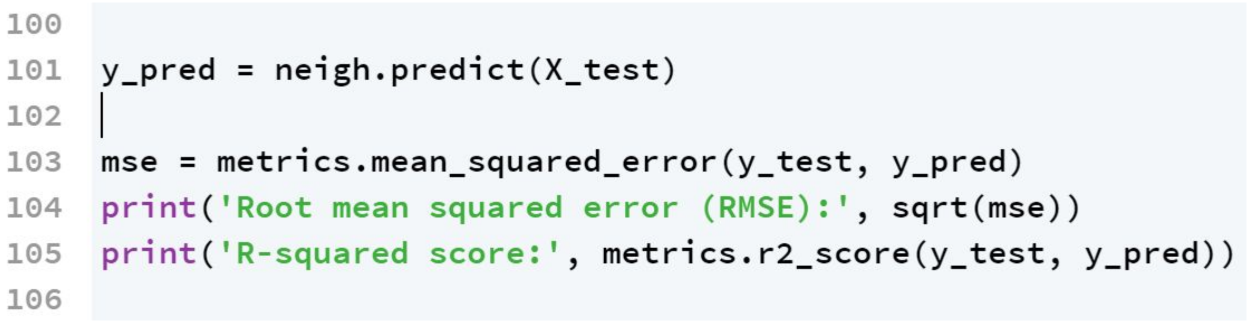
Lines 64-69: Sliced the dataFrame for input variables and target variables and utilised k-nearest neighbours regression



```
70
71  us_scap = True
72  uk_scap = True
73  au_scap = True
74  while us_scap:
75      inUS = input('US cases?')
76      if inUS.isnumeric():
77          us_scap = False
78  while uk_scap:
79      inUK = input('UK cases?')
80      if inUK.isnumeric():
81          uk_scap = False
82  while au_scap:
83      inAu = input('Au cases?')
84      if inAu.isnumeric():
85          au_scap = False
86
87
88
89  sample = [int(inUK) ,int(inUS),int(inAu)]
90  print('~~~~~~~\nWill you make profit for a single day investment?\n')
91  for column, value in zip(list(largedata)[6:9], sample):
92      print(column + ' today -   ' + str(value))
93  sample_pred = neigh.predict([sample])
94  if sample_pred ==1:
95      a = 'Invest today'
96  else:
97      a = 'Not today'
98  print("verdict: " + a+ "\n")
99  print('~~~~~~~')
```

Lines 70-85: A few while loops were created for inputting the reported covid cases for the day, and made sure these values are numeric so that there are no eros in the test.N.B. We decided not to use the covid data from india as it had extremely different volatility and we did't want this to potentially misconstrue the predictions.

Lines 86-99: Our cleaned cases are then fed into the algorithm to predict a 1 or 0. If a 1 is predicted the algorithm returns “invest today”, if the algorithm predicts 0 it returns “not today”.



```
100
101 y_pred = neigh.predict(X_test)
102
103 mse = metrics.mean_squared_error(y_test, y_pred)
104 print('Root mean squared error (RMSE):', sqrt(mse))
105 print('R-squared score:', metrics.r2_score(y_test, y_pred))
106
```

Lines 100-105: Finally the Root mean squared error and the r-squared score is calculated from the predictions and printed.

This concludes the textual explanation of the coding for the predictive model.

The k-nearest neighbour prediction model was deemed appropriate and developed for this report because it was capable of being able to make predictions about the question we were wanting to answer, the question being whether COVID-19 cases impact value fluctuation in indexes. A k-nearest neighbour regression model derives a predicted outcome based on similarity, specifically the similarity between the input data and historical data already stored in the database. For our example above this was comparing the new daily covid-cases against days where the number of Covid cases were similar and by looking at what happened to the index prices on those specific days, the model would predict the stock price change today. The method of approaching and constructing the k-nearest neighbour regression model is outlined in the code above, utilising the ‘train_test_split’ and ‘KNeighborsRegressor()’ functions. The simplicity of construction of the model, the ability of the model to aid in answering our fundamental question regarding how Covid -19 cases impact index prices and along with the user friendly output we’re the reasons why this approach was used.

Similarly to the linear regression model evaluation of the model was undertaken through the development and outputting of measures of accuracy. For the k-nearest neighbour regression model the RMSE were relatively small, indicating a low degree of error and that the predicted values were quite close to the observed values. Ultimately, these measures of accuracy/effectiveness have allowed the evaluation to be made that the predictive model was successful with some limitations.

Conclusion

Both predictive models serve to explore and answer the question of what is the relationship between increases in COVID-19 cases in Australia, the US, the UK and India and the change in value of publicly traded Indexes (Financials, Healthcare, Resources and Info Tech) on the ASX. For the linear regression model specifically, it seeks to determine to what degree the discovery of new daily COVID-19 cases may cause a fluctuation in the value of the index such that the difference between the highest and lowest value of the index in a trading day may increase. Ultimately, the linear regression model was able to confirm expected outcomes such as that increases in Australian and UK COVID-19 cases increased the degree of fluctuation in the index values and that Australian COVID-19 cases had the most significant effect on the difference between the high-low difference. However, the model's output asserted that increases in COVID-19 in India and the US decrease the fluctuation value of the high-low difference. This wasn't expected and appears to be illogical. Consideration of these results must be accompanied by an examination of the quality of the predictive model. The output of the model showed that there were substantial errors between predicted and observed values and that the correlation between COVID-19 cases and index high-low differences were very minimal. This means that the model is severely limited. Despite the advantages of the simplistic training and creation process for the linear regression model and that the linear regression model is perfectly appropriate to modelling the numerical relationship between COVID-19 and index values, this experience has suggested that it is perhaps not ideal for the purposes of answering the report's question.

Rather, the k-nearest neighbours regression model is perhaps a better model. The k-nearest neighbours model attempts to predict the movements of the index stock prices based on the number of new Covid-19 cases. Furthermore based on the results of the model, it could be determined whether or not to buy a certain index stock depending on if the stock price was predicted to increase or decrease. Ultimately, the k-nearest neighbours regression model was able to predict the movement of the stock price based on historical similarity of index price movements on days which showed similar new Covid-19 cases. The effectiveness of the model was revealed within the Root Mean Squared error. As the RMSE score for the k-nearest neighbours regression is relatively less than in the linear regression model, this suggests that the k-nearest neighbours model produced a lower degree of errors in relation to predicted values versus observed values, indicating the model is perhaps more accurate. As a result, the k-nearest neighbours regression model could be deemed more effective than the linear regression model in aiding to explore the question of how Covid-19 cases effect fluctuations in index stock prices as the k-nearest neighbour model had relatively less degree of error in relation to the RMSE. Further, the prediction of whether it is profitable to invest or not has undoubtedly more value to investors than a simple prediction of how much the value will fluctuate during a day. As such, the k-nearest neighbours regression model is better suited to a real life predictive model that can inform and influence investment decisions during the COVID-19 pandemic.