



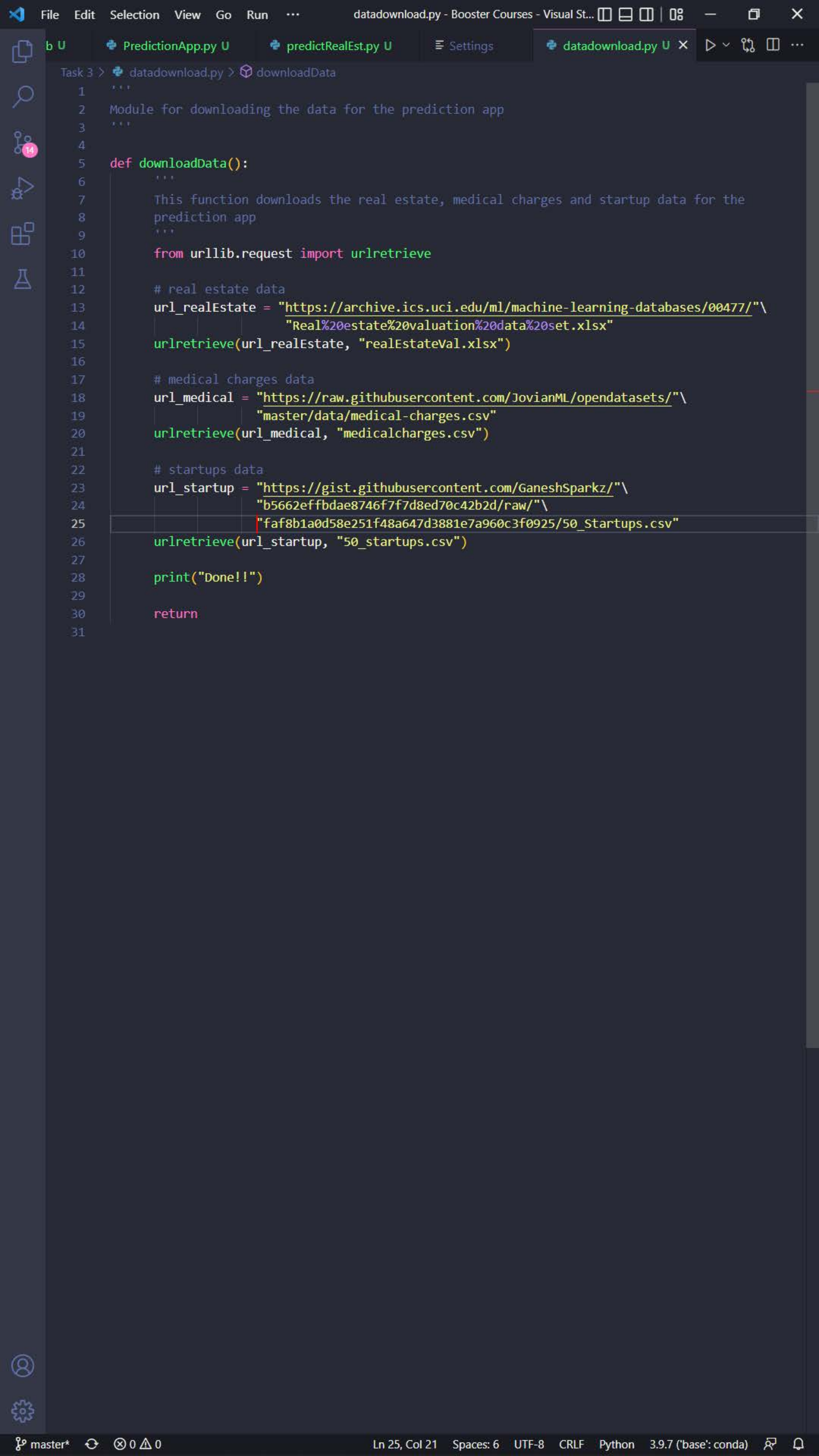
PredictionApp.py U X

PredictionApp.py &gt; ...

```
1  print("-----"*10)
2  print("\t\tPrediction App")
3  print("-----"*10)
4
5  i = 1
6  while i == 1:
7      # printing options
8      print('''
9      Choose your option. Remember to download data
10     [1]  Download the required data
11     [2]  Predict Real Estate values of properties in Taiwan
12     [3]  Predict Medical Charges of patients at Bowen hospital
13     [4]  Predict profit of startups in the US
14     ''')
15
16     userOpt = int(input("What is your option? "))
17     if userOpt == 1:
18         import datadownload
19         print("Downloading required data")
20         datadownload.downloadData()
21     elif userOpt == 2:
22         import predictRealEst
23         predictRealEst.fitModel()
24         predictRealEst.realEstateVal()
25
26         cont = input("\nWill you like to continue? [Yes/No]")
27         if cont.lower() == 'yes':
28             i = 1
29         else:
30             i = 0
31     elif userOpt == 3:
32         import predictMedical
33         predictMedical.fitModel()
34         predictMedical.medicalChargesCalc()
35
36         cont = input("\nWill you like to continue? [Yes/No]")
37         if cont.lower() == 'yes':
38             i = 1
39         else:
40             i = 0
41     elif userOpt == 4:
42         import predictStartUps
43         predictStartUps.fitModel()
44         predictStartUps.startUpEval()
45
46         cont = input("\nWill you like to continue? [Yes/No]")
47         if cont.lower() == 'yes':
48             i = 1
49         else:
50             i = 0
51     else:
52         print("Wrong option entered")
```









Task 3 &gt; predictMedical.py &gt; medicalChargesCalc

```
1 '''
2 For predicting the medical charges of patients in a hospital based on age,
3 bmi, sex, smoker/non smoker,
4 number of children and region in the US
5 '''
6
7 def fitModel():
8     '''
9     Fits the model based on historical data and returns the model
10    '''
11    import pandas as pd
12    import numpy as np
13    from sklearn.linear_model import LinearRegression
14    from sklearn.preprocessing import OneHotEncoder
15    medicalCharges = pd.read_csv("medicalcharges.csv")
16
17    # converting categorical data into 1s and 0s
18    sex_codes = {'female':0, 'male':1}
19    smoker_codes = {'yes':1, 'no':0}
20    medicalCharges['sex_code'] = medicalCharges.sex.map(sex_codes)
21    medicalCharges['smoker_code'] = medicalCharges.smoker.map(smoker_codes)
22
23    # one hot encoding the region column
24    enc = OneHotEncoder()
25    enc.fit(medicalCharges[['region']])
26    one_hot = enc.transform(medicalCharges[['region']]).toarray()
27    medicalCharges[enc.categories_[0]] = one_hot
28
29    # Create inputs and target
30    input_cols = ['age', 'bmi', 'children', 'sex_code', 'smoker_code',
31                  'northeast', 'northwest', 'southeast', 'southwest']
32    inputs = medicalCharges[input_cols]
33    target = medicalCharges.charges
34
35    # create and train the model
36    model = LinearRegression()
37    model.fit(inputs, target)
38
39    return model
40
41
42 def medicalChargesCalc():
43     '''
44     Get how much a customer will pay as a premium each year
45     '''
46     print('''
47         Time to predict your annual medical charge.
48         ''')
49
50     # collecting arguments
51     age = int(input("Enter your age: "))
52     bmi = float(input("Enter your BMI (Body Mass Index): "))
53     children = int(input("How many children do you have? "))
54     smoker = input("Are you a smoker? [Yes/No] ")
55     smoker = smoker.lower()
56     if smoker == "yes":      # to find the smoker_code based on response
57         smoker_code = 1
58     else:
59         smoker_code = 0
60
61     sex = input("What is your gender [Male/Female]? ")
62     sex = sex.lower()
63     if sex == "male":      # to find the sex code based on response
64         sex_code = 1
65     else:
66         sex_code = 0
67     region = input("What region are you from? NorthEast or NorthWest\
68                  " or SouthEast or SouthWest? ")
69     region = region.lower()
70     northeast, northwest, southeast, southwest = 0,0,0,0
71     if region == "northeast":
72         northeast = 1
73     elif region == "northwest":
74         northwest = 1
75     elif region == "southeast":
```







```
Model Creation.ipynb U PredictionApp.py U predictRealEst.py U X Settings
Task 3 > predictRealEst.py > realEstateVal
1 '''
2 The market historical data set of real estate valuation are collected from Sindian Dist.,
3 New Taipei City, Taiwan.
4
5 The inputs are as follows
6 - TransactionDate = the transaction date (for example, 2013.250=2013 March,
7   |               | 2013.500=2013 June, etc.)
8 - HouseAge = the house age (unit: year)
9 - Dist to Nearest MRT Station = the distance to the nearest MRT (Mass Rapid Transit) station
10   |               |               | (unit: meter)
11 - No of Convenience Stores = the number of convenience stores in the
12   |               |               | living circle on foot (integer)
13 - Latitude = the geographic coordinate, latitude. (unit: degree)
14 - Longitude = the geographic coordinate, longitude. (unit: degree)
15
16 The output is as follow
17 - House Price of Unit Area = house price of unit area (10000 New Taiwan Dollar/Ping,
18   |               |               | where Ping is a local unit,
19   |               |               | 1 Ping = 3.3 meter squared)
20 '''
21
22 def fitModel():
23     '''
24     Fits the model based on historical data and returns the model
25     '''
26
27     import pandas as pd
28     import numpy as np
29     from sklearn.linear_model import LinearRegression
30     realEstateVal = pd.read_excel("realEstateVal.xlsx")
31
32     ## Renaming Columns
33     colNames = ['No', 'TransactionDate', 'HouseAge', 'Dist to Nearest MRT Station',
34                 'No of Convenience Stores', 'Latitude', 'Longitude',
35                 'House Price of Unit Area']
36     realEstateVal.columns = colNames
37
38     # deleting 'No' column since is not needed
39     del realEstateVal['No']
40
41     # creating input and targets
42     input_cols = ['TransactionDate', 'HouseAge', 'Dist to Nearest MRT Station',
43                  'No of Convenience Stores', 'Latitude', 'Longitude']
44     inputs = realEstateVal[input_cols]
45     target = realEstateVal['House Price of Unit Area']
46
47     # create and fit model
48     model = LinearRegression()
49     model.fit(inputs, target)
50
51     return model
52
53 def realEstateVal():
54     '''
55     Get the real estate valuation of properties in Sindian Dist., New Taipei City, Taiwan.
56     '''
57
58     print('''
59         Time to predict the evaluation of real estates in Sindian district, Taiwan.
60         ''')
61
62     # collecting arguments
63     year = int(input("Enter the year of Transaction: "))
64     month = int(input("Enter the month number of transaction (Ex. March = 2): "))
65     transactionDate = year + month/12
66     houseAge = float(input('Enter House age in years: '))
67     distMRT = float(input("What is the distance to the nearest MRT Station (meters)? "))
68     noOfStores = int(input("How many stores are around the property? "))
69     latitude = float(input("What is its latitude? "))
70     longitude = float(input("What is its longitude? "))
71
72     # generating prediction
73     predictors = [transactionDate, houseAge, distMRT, noOfStores, latitude, longitude]
74     model = fitModel()
75     prediction = model.predict([predictors])
76
77     print()
78     print(f'the value of the house is {prediction[0]:.2f} per unit area\'')
```



```
Task 3 > predictRealEst.py > realEstateVal
14 - Longitude = the geographic coordinate, longitude. (unit: degree)
15
16 The output is as follow
17 - House Price of Unit Area = house price of unit area (10000 New Taiwan Dollar/Ping,
18 | | | | | where Ping is a local unit,
19 | | | | | 1 Ping = 3.3 meter squared)
20 '''
21
22 def fitModel():
23     '''
24     Fits the model based on historical data and returns the model
25     '''
26     import pandas as pd
27     import numpy as np
28     from sklearn.linear_model import LinearRegression
29     realEstateVal = pd.read_excel("realEstateVal.xlsx")
30
31     ## Renaming Columns
32     colNames = ['No', 'TransactionDate', 'HouseAge', 'Dist to Nearest MRT Station',
33 | | | | | 'No of Convenience Stores', 'Latitude', 'Longitude',
34 | | | | | 'House Price of Unit Area']
35     realEstateVal.columns = colNames
36
37     # deleting 'No' column since is not needed
38     del realEstateVal['No']
39
40     # creating input and targets
41     input_cols = ['TransactionDate', 'HouseAge', 'Dist to Nearest MRT Station',
42 | | | | | 'No of Convenience Stores', 'Latitude', 'Longitude']
43     inputs = realEstateVal[input_cols]
44     target = realEstateVal['House Price of Unit Area']
45
46     # create and fit model
47     model = LinearRegression()
48     model.fit(inputs, target)
49
50     return model
51
52
53 def realEstateVal():
54     '''
55     Get the real estate valuation of properties in Sindian Dist., New Taipei City, Taiwan.
56     '''
57     print('''
58 |         Time to predict the evaluation of real estates in Sindian district, Taiwan.
59 |         ''')
60     # collecting arguments
61     year = int(input("Enter the year of Transaction: "))
62     month = int(input("Enter the month number of transaction (Ex. March = 2): "))
63     transactionDate = year + month/12
64     houseAge = float(input('Enter House age in years: '))
65     distMRT = float(input("What is the distance to the nearest MRT Station (meters)? "))
66     noOfStores = int(input("How many stores are around the property? "))
67     latitude = float(input("What is its latitude? "))
68     longitude = float(input("What is its longitude? "))
69
70     # generating prediction
71     predictors = [transactionDate, houseAge, distMRT, noOfStores, latitude, longitude]
72     model = fitModel()
73     prediction = model.predict([predictors])
74     print()
75     print(f'the value of the house is {prediction[0]:.2f} per unit area\'
76 |         ' (10000 New Taiwan Dollar/Ping)')
```



Task 3 &gt; predictStartUps.py &gt; fitModel

```
1  '''
2  The data is based on records gotten 50 startups from 3 states.
3
4  The inputs are as follows
5  0   R&D Spend      : Research and Development expenses
6  1   Administration : Administration expenses
7  2   Marketing Spend : Marketing expenses
8  3   State          : State in the US.
9  4   Profit         : Profit made by each startup
10
11  '''
12
13  def fitModel():
14      '''
15      Fits the model based on historical data and returns the model
16      '''
17      import pandas as pd
18      import numpy as np
19      from sklearn.linear_model import LinearRegression
20      from sklearn.preprocessing import OneHotEncoder
21      startUp = pd.read_csv("50_startups.csv")
22
23      ## one hot encoding for the `State` column
24      enc = OneHotEncoder()
25      enc.fit(startUp[['State']])
26      one_hot = enc.transform(startUp[['State']]).toarray()
27      startUp[enc.categories_[0]] = one_hot
28
29      # create inputs and target
30      input_cols = ['R&D Spend', 'Administration', 'Marketing Spend',
31                  'California', 'Florida', 'New York']
32      inputs = startUp[input_cols]
33      target = startUp['Profit']
34
35      # create and train the model
36      model = LinearRegression()
37      model.fit(inputs, target)
38
39      return model
40
41
42  def startUpEval():
43      '''
44      Get the expected profit of a start up based on some variables.
45      '''
46      print('''
47          Time to predict the profit of your startup.
48          ''')
49      # collecting arguments
50      RDspend = float(input("What is the R&D expense? "))
51      admin = float(input("What is the Administration expense? "))
52      marketing = float(input("What is the Marketing expense? "))
53      state = input("What state is the startup in? [California / Florida / New York] ")
54      state = state.lower()
55      # converting state into one hot encoded variables
56      california, florida, newyork = 0,0,0
57      if state == "california":
58          california = 1
59      elif state == "florida":
60          florida = 1
61      elif state == "new york":
62          newyork = 1
63      else:
64          print("Invalid state added")
65          return
66
67      # generating prediction
68      predictors = [RDspend, admin, marketing, california, florida, newyork]
69      model = fitModel()
70      prediction = model.predict([predictors])
71      print()
72      print(f'This startup is expected to make a profit of ${prediction[0]:.2f}')
```



