

- Developer: Sophia Xiao @ 2021/1/2
- Dataset downloaded from Tianchi  
<https://tianchi.aliyun.com/dataset/dataDetail?dataId=83994>
- Goal: predicting the interest level (low-med-high) of testing file by analyzing features in training file
- Website cited:
  - <https://stackoverflow.com/questions/51452031/how-to-use-the-test-data-against-the-trained-model>
  - <https://stackoverflow.com/questions/45681387/predict-test-data-using-model-based-on-training-data-set>
  - <https://www.kaggle.com/c/titanic/discussion/54683>
- Steps:
  - Pull and see useful features that can help in prediction: in this case, I only handle the numeric features
  - I need to predict categorical variable (low-med-high) in testing file, so when I analyse the training file, I need to convert the responsible variable to numbers in training file (low - 1 , med - 2, high -3)
  - Using module in python package installed (I choose decision tree here, but I yet to know the difference between so many prediction tool such as something like “forest tree(?)” )
  - Convert the response variable 1, 2, 3 back to interest level
- Questions I had:
  - Can features other than numerical also be mixed in the prediction process?

- Is there a way to convert the response variable other than I create a dictionary?

The screenshot shows the PyCharm IDE with a project named 'feature'. The file explorer on the left lists various files, including 'Scripts', 'backup.py', 'boxplot.py', 'des\_keywords.jpg', 'description keywords.py', 'feature\_corr.png', 'feature\_corr.py', 'h1b.py', 'h1b\_kaggle.csv', 'interest\_level\_pred.py', 'lat&logprice\_boxplot.png', 'para\_corr.png', 'parallel\_coordinate.py', 'px\_parallel\_coo.PNG', 'pyenv.cfg', 'test.json', 'testing\_data.csv', 'train.json', and 'training\_data.csv'. The main editor displays the code for 'interest\_level\_pred.py'.

```

18
19 d = {'low':1, 'medium':2, 'high':3}
20 df1["interest_level"] = df1["interest_level"].map(d)
21 ytrain = df1["interest_level"]
22
23
24 df2 = pd.read_json('test.json')
25 xtest = df2[x_col]
26
27
28 model = DecisionTreeRegressor()
29 model.fit(xtrain, ytrain)
30 pred = model.predict(xtest)
31 df2['pred_level'] = pred
32 df2['pred_level'] = df2['pred_level'].astype(int)
33 d2 = {1:'low', 2:'medium', 3:'high'}
34 df2['pred_level'] = df2['pred_level'].map(d2)
35 df1["interest_level"] = df1["interest_level"].map(d2)
36
37 df1.to_csv("training_data.csv")
38 df2.to_csv("testing_data.csv")

```

The Run window shows the command: `C:\Users\xht\PycharmProjects\feature\venv\Scripts\python.exe C:/Users/xht/PycharmProjects/feature`. The output displays a table with columns: bathrooms, bedrooms, ..., street\_address, pred\_level.

	bathrooms	bedrooms	...	street_address	pred_level
0	1.0	1	...	99 Suffolk Street	medium
1	1.0	2	...	176 Thompson Street	high
2	1.0	0	...	115 Sullivan Street	low
3	1.0	2	...	23 Jones Street	high
5	1.0	1	...	20 Exchange Place	low
...	...	...	...	...	...
124003	1.0	1	...	158 EAST 107TH STREET	medium
124005	1.0	2	...	141 E 33rd St.	low

The bottom status bar shows: IDE and Plugin Updates: PyCharm is ready to update. (40 minutes ago) 40:11 CRLF UTF-8 4 spaces