CE9010 INTRO TO DATA SCIENCE AY17/18 PROJECT

(A Look into rental fee on PropertyGuru)

Name: Chong Ke Xin

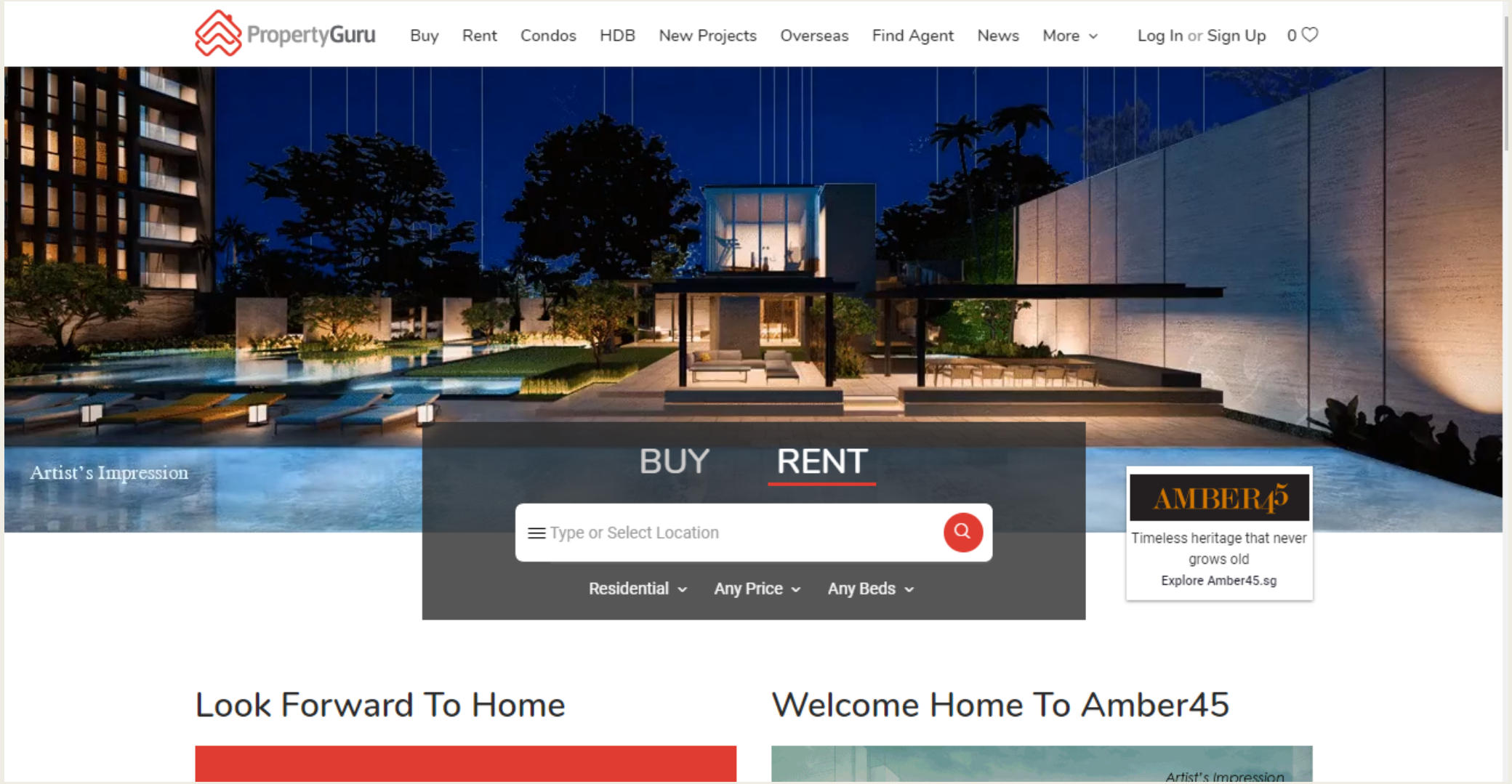
Matric number: U1440121F

Group ID: 9 (3:15-3:22pm)

Step 1: A data problem to solve

- Where can I get the cheapest house to rent with some preference?

Step 2 : Data acquisition



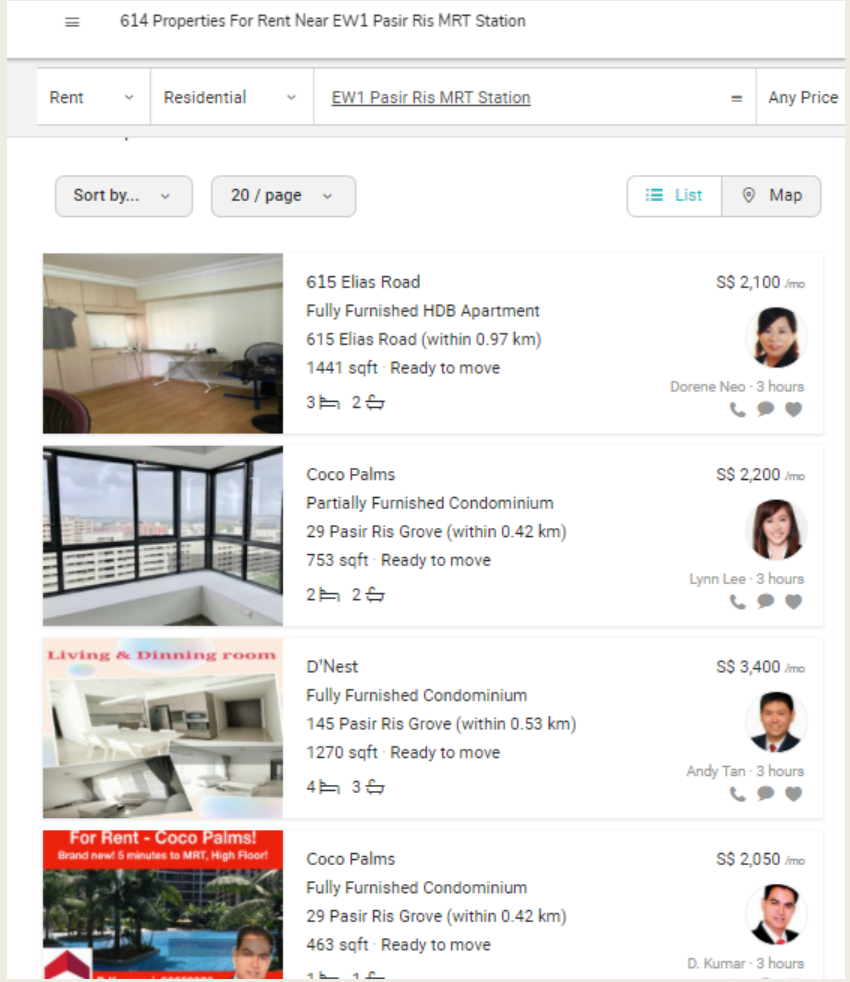
- Scrapped data VS original page:

Scrapped data:

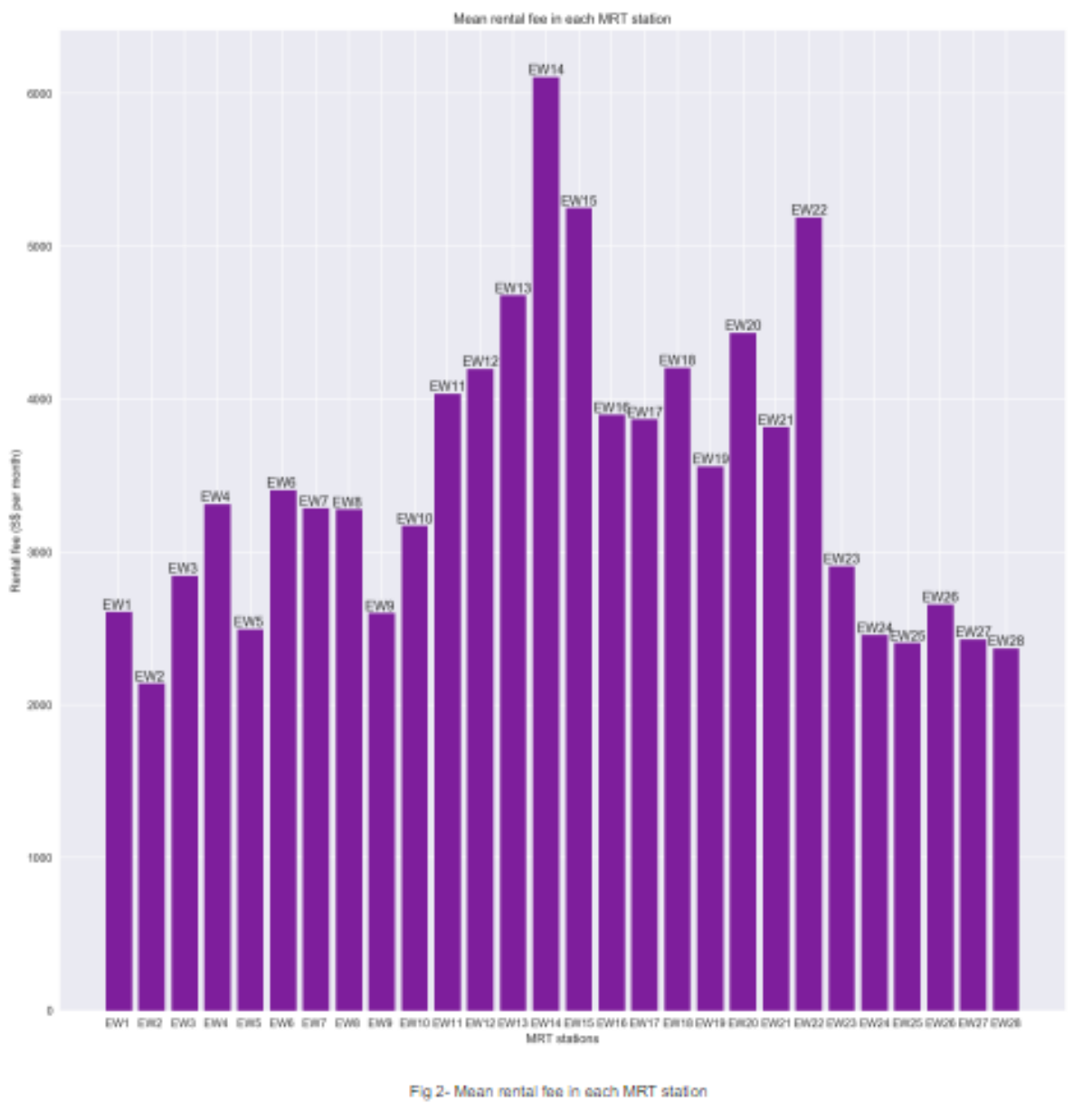
```
data.iloc[:15,:]
```

index	name	details	size	address	available time	bed	bath	price	nearest MRT ID	distance
0	Coco Palms	Partially Furnished Condominium	904 sqft	Pasir Ris Drive 1	Ready to move	3	2	S$ 2,999 /mo	EW1 Pasir Ris MRT Station	0.42 km
1	D'Nest	Partially Furnished Condominium	484 sqft	129 Pasir Ris Grove	Ready to move	1	1	S$ 1,850 /mo	EW1 Pasir Ris MRT Station	0.53 km
2	D'Nest	Fully Furnished Condominium	743 sqft	Pasir Ris Grove	Ready to move	2	2	S$ 2,350 /mo	EW1 Pasir Ris MRT Station	0.53 km
3	NV Residences	Fully Furnished Condominium	1087 sqft	87 Pasir Ris Grove	Ready to move	3	2	S$ 2,800 /mo	EW1 Pasir Ris MRT Station	0.54 km
4	Livia	Partially Furnished Condominium	1539 sqft	69 Pasir Ris Grove	Ready to move	4	4	S$ 3,499 /mo	EW1 Pasir Ris MRT Station	0.82 km
5	The Palette	Partially Furnished Condominium	1377 sqft	103 Pasir Ris Grove	Ready to move	4	3	S$ 3,400 /mo	EW1 Pasir Ris MRT Station	0.53 km

PropertyGuru website:



Step 3: Data exploration



- Mean rental fee near **EW14 Raffles Place MRT station** is the highest (approx. S$ 6000 per month).

Classify price into different classes

```
{'min': 230, 'mean': 3700.0913077372834, 'median': 3000.0, 'max': 35000}
{'25% quantile': 2300.0, '50% quantile': 3000.0, '75% quantile': 4200.0}
```

- It can be seen from the graph on the right, most of the data cluster below S$ 5k per month.
- From the graph above, 75% of the data are less than S$ 4,200 per month.

Therefore, the price range for each classes are being set as follow:

class 0 : 1-500
class 1 : 501-1000
class 2 : 1001-1500
class 3 : 1501-2000
class 4 : 2001-2500
class 5 : 2501-3000
class 6 : 3001-3500
class 7 : 3501-4000
class 8 : 4001-4500
class 9 : > 4500

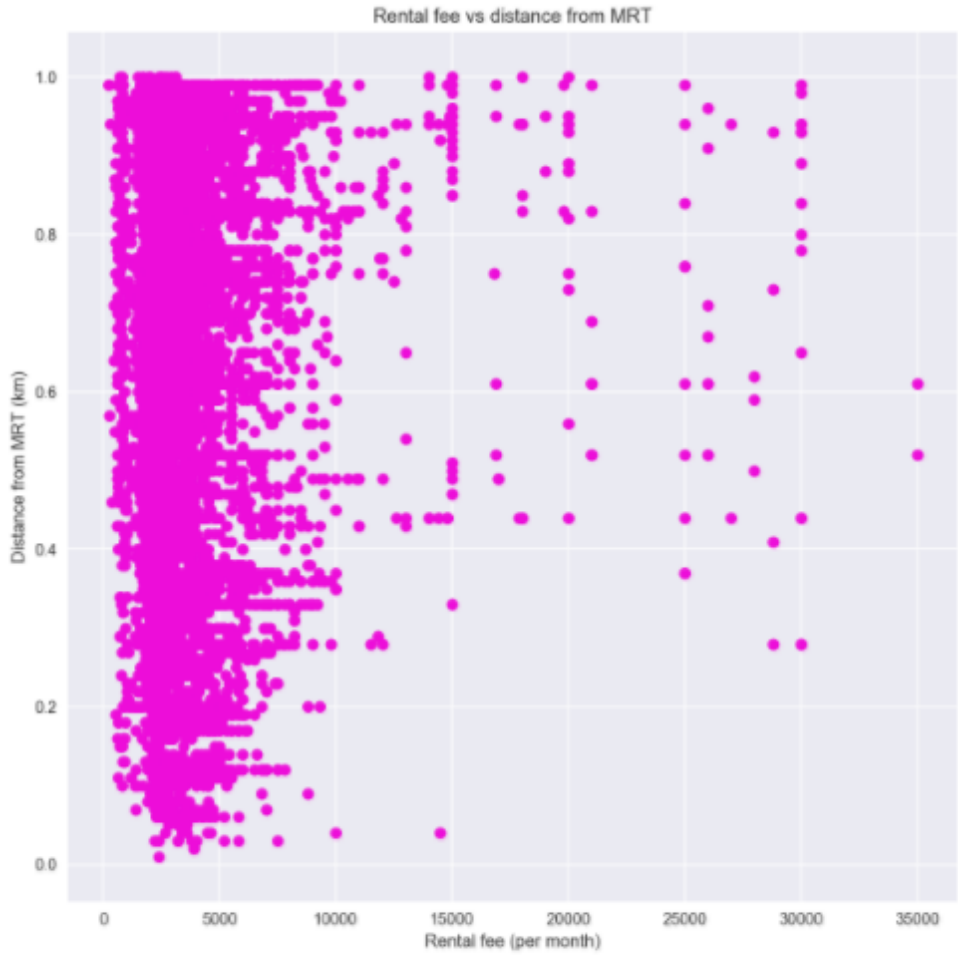


Fig 4 - Rental fee vs distance from MRT

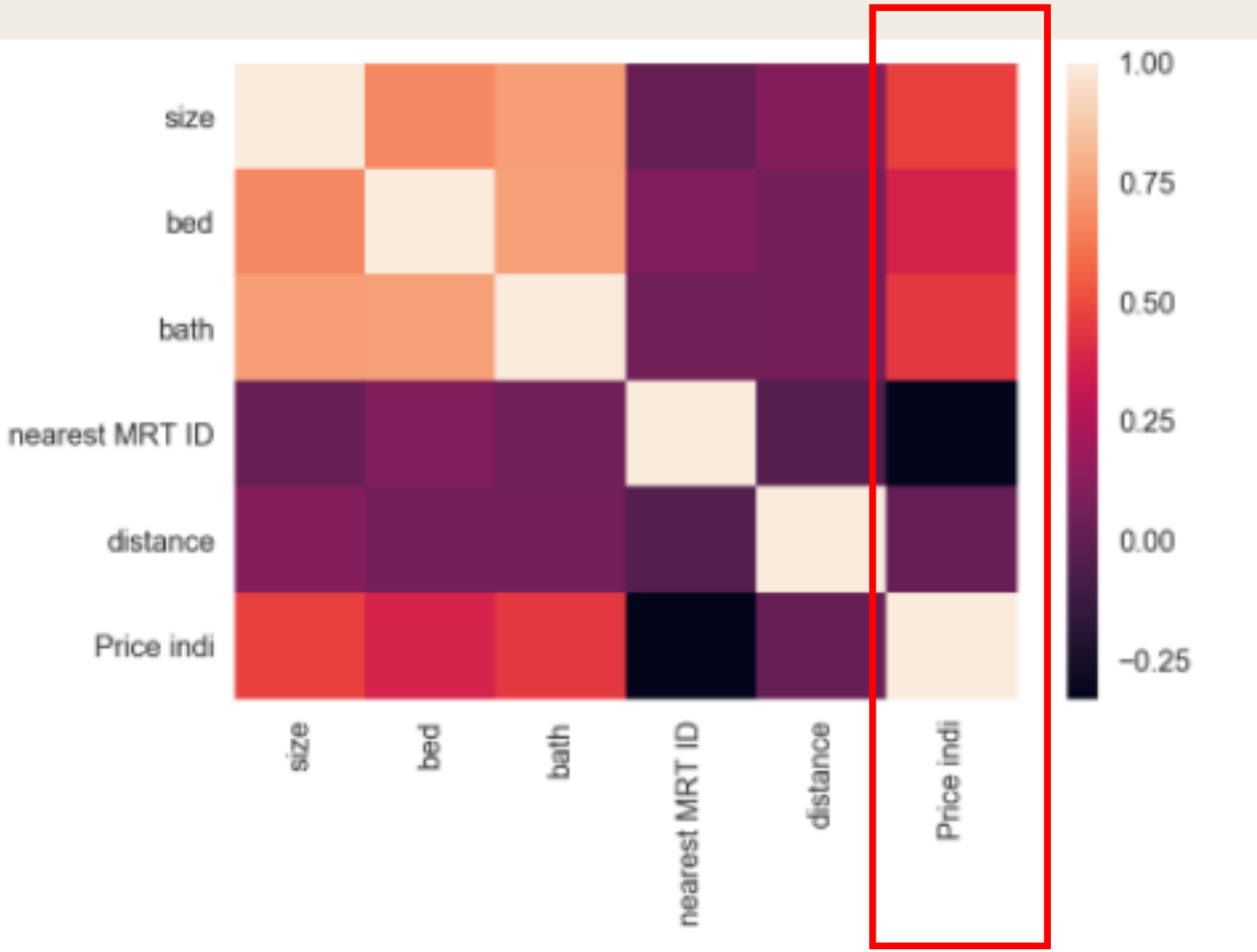


Fig 1 - Correlation heatmap

- I will be predicting “price indi” (The class of price range).
- Correlated:
  - *Size of the house*
  - *Total number of bedrooms*
  - *Total number of bathrooms*
  - *Nearest MRT*
- Weak correlation:
  - *Distance*

Step 4: Pre-processing

- Remove unwanted information (address, available time).
- Dropping data with NA values.
- Remove duplicate records.
- One-hot-encoding for nearest MRT ID (avoid unintentionally weight biase for ‘nearest MRT ID’ feature).
- Normalizing (Z-scoring).

Step 5: Data analysis

- 5 features, predict 10 classes

1) Random Forest

```
start = time.time()
clf = RandomForestClassifier(n_estimators=1000)
clf.fit(data_train_x, data_train_y)
print('Time=',time.time() - start)

# Check accuracy
sklearn.metrics.accuracy_score(data_test_y, clf.predict(data_test_x))
```

```
Time= 49.768866777420044

0.636629322942036
```

```
array([[  1,   1,   0,   0,   0,   0,   0,   0,   0,   0],
       [  0,  25,   2,   7,   9,   4,   0,   1,   1,   0],
       [  0,  10,   5,   9,   4,   3,   1,   2,   0,   1],
       [  0,   2,   4, 177,  61,   6,   5,   1,   2,   0],
       [  0,   3,   0,  65, 281,  53,  13,   2,   4,   1],
       [  0,   1,   0,   6,  75, 190,  43,   8,   7,  15],
       [  0,   0,   1,   2,  19,  49,  85,  29,  14,  11],
       [  0,   0,   0,   0,   6,   7,  36,  83,  41,  13],
       [  0,   0,   0,   0,   3,   4,   8,  24,  55,  30],
       [  0,   0,   0,   0,   4,  14,   1,  11,  24, 368]], dtype=int64)
```

- Using 1000 trees.
- Accuracy 63.66%.

- mis-predicted classes are always lies one class above or below the actual class.

2) Logistic Regression (multinomial)

```
start = time.time()
logreg_sklearn = LogisticRegression(max_iter=300,C=1e6,solver='newton-cg',multi_class='multinomial')
logreg_sklearn.fit(data_train_x, data_train_y) # learn the model parameters
print('Time=',time.time() - start)

sklearn.metrics.accuracy_score(data_test_y, logreg_sklearn.predict(data_test_x))
```

```
Time= 16.496556758880615

0.48319532391622017
```

- Multinomial.
- Accuracy only 48.31%.
- Cross-validation on C have been done.

5-fold cross-validation for C

```
kf = KFold(n_splits=5)
cross_c = []
for C in [1e-6,1e-4,1,1e-2,1e2,1e4,1e6]:
    temp_cost = []
    for train_index, test_index in kf.split(data_train_x):
        print("TRAIN:", train_index, "TEST:", test_index)
        X_train, X_test = cv_normalize(data_train_x,train_index,test_index)
        # X_train, X_test = data_train_x.loc[train_index], data_train_x.loc[test_index]
        y_train, y_test = data_train_y.loc[train_index], data_train_y.loc[test_index]

        logreg_sklearn = LogisticRegression(max_iter=300,C=C,solver='newton-cg',
                                            multi_class='multinomial')
        logreg_sklearn.fit(X_train, y_train) # Learn the model parameters

        # Check for loss
        one_hot = OneHotEncoder(n_values=10, sparse=False)

        y_true = one_hot.fit_transform(np.array(y_test).reshape(-1, 1))
        y_pred = one_hot.fit_transform(logreg_sklearn.predict(X_test).reshape(-1, 1))
        temp_cost.append(log_loss(y_true, y_pred))

    cross_c.append(np.mean(temp_cost))
```

Confusion matrix:

- Similarly, mis-predicted classes are always lies one class above or below the actual class.
- Class 0 is in 0% accuracy. (class-imbalance problem)

```
array([[  0,   1,   0,   1,   0,   0,   0,   0,   0,   0],
       [  0,  20,   0,  12,  12,   4,   0,   1,   0,   0],
       [  0,   8,   0,  14,   9,   2,   0,   0,   0,   2],
       [  0,   6,   0, 146,  85,  18,   0,   1,   0,   2],
       [  0,   3,   0,  62, 251,  75,  15,   3,   1,  12],
       [  0,   1,   0,  21,  94, 161,  23,   7,   4,  34],
       [  0,   0,   0,   8,  35,  80,  22,  13,   2,  50],
       [  0,   0,   0,   1,  15,  51,  36,  17,   1,  65],
       [  0,   0,   0,   0,   5,  32,  20,  11,   5,  51],
       [  0,   0,   0,   2,   3,  21,  11,   9,   6, 370]], dtype=int64)
```

Step 6: Analysis of results

- As logistic regression is linear with its predictive functions, and from Figure 4 in slide 6, the relationship between rental fee and distance is not linear. Logistic regression model is not a good choice for this situation.
- It can still achieve the accuracy of 48.31% as some of the features is in linear relationship (as the graph on the right).

Fig 5 - Rental fee vs size of property

Implication

- Finding optimal distance and price one could get under some preferences. (Please refer to my report for more details)
  1. *Rough idea: input (preference size, # of bedrooms,# of bathrooms, preference MRT station ID in one hot encoding mode).*
  2. *Run through distance (e.g. 0 to 1 km, with 0.1km per step), get the predicted price range and check which distance will resulting in cheapest rental fee.*

Improvements

- Add more data features (details, available time, agent's rating…).
- Increase data size (scrap from more websites).
- Class imbalance.
- Could be generalized to North-East line, Circle line, etc.

Thank you! ☺