Final Report

Shuqi Ma, Shiwen Xu, Nan Wang

Problem

Using the various features associated with houses to predict the house price. Specifically, we will predict the house price range given features of a new house.

Dataset

This dataset corresponds to a house price associated with 15 features, which contains 500k sample houses. Provided with Mean, standard deviation, and quantiles ,we will arbitrarily choose 1000 and 600 samples with all features except black, indian marble,fiber and fireplace.

Dataset URL: https://www.kaggle.com/greenwing1985/housepricing

Preprocess

From the original dataset, we convert each house information to list and the label corresponding to the price of the house into 10 ranks in the format of numpy matrix.

1. Convert the dataset to 12 features we are going to test and separately store the price. Specifically, we remove features about the color of marbles and narrow down to only one of it, which is white and remove the feature about fiber and number of fireplaces.
2. Randomly select 1000 samples from dataset, and also choose 600 samples from previous 1000 samples selected. We choose 100 samples for testing excluding those 1000 samples. What's more, we choose 500 samples for tuning parameters excluding all of the above samples.
3. Based on the minimum price and maximum price, we divide houses' prices into 10 ranges evenly working as labels.

Algorithm and Language

Algorithm: We will use K nearest neighbor and Decision Trees from scikit-learn.
Language: Python 3.7

Hyperparameters

Decision tree: min_sample_leaf, max_feature, max_depth
k-nearest neighbors: k

Cross-Validation

Cross-validation technique: training/validation/testing and k-fold, and we choose k= 5. We select 500 samples from processed data to tune the hyperparameters and choose the best hyperparameter suitable for our two algorithms

Evaluation

Training: randomly picking 1000 and 600 data
Testing set: 100 samples
We evaluate two models using 100 samples in testing subset. We chose to select a testing set separately from the raw dataset and it does not intersect with either training set.

Experimental Results

We used k fold validation with (k = 5) to select the hyperparameter. Parameter tuning is performed in the following parameter:

- **Best K for KNN:**

  max_k=2:
  Accuracy: 0.101010
  Precision: 0.120928
  Recall: 0.101010
  F-1 score: 0.110075

  max_k=3:
  Accuracy: 0.131313
  Precision: 0.137471
  Recall: 0.131313
  F-1 score: 0.134322

  max_k=4:
  Accuracy: 0.141414
  Precision: 0.131101
  Recall: 0.141414
  F-1 score: 0.136062

max_k=5:
Accuracy: 0.111111
Precision: 0.100696
Recall: 0.111111
F-1 score: 0.105648

max_k=6:
Accuracy: 0.131313
Precision: 0.118778
Recall: 0.131313
F-1 score: 0.124731

max_k=7:
Accuracy: 0.161616
Precision: 0.141093
Recall: 0.161616
F-1 score: 0.150659

max_k=8:
Accuracy: 0.191919
Precision: 0.158386
Recall: 0.191919
F-1 score: 0.173547

max_k=9:
Accuracy: 0.242424
Precision: 0.196916
Recall: 0.242424
F-1 score: 0.217313

max_k=10:
Accuracy: 0.191919
Precision: 0.152832
Recall: 0.191919
F-1 score: 0.170160

max_k=11:
Accuracy: 0.191919
Precision: 0.163283
Recall: 0.191919
F-1 score: 0.176447

Best max_k with 500 data: 9

- **Best depth for Decision Tree**:

max_depth=2:
Accuracy: 0.272727
Precision: 0.139207
Recall: 0.272727
F-1 score: 0.184328

max_depth=3:
Accuracy: 0.252525
Precision: 0.162125
Recall: 0.252525
F-1 score: 0.197471

max_depth=4:
Accuracy: 0.292929
Precision: 0.263966
Recall: 0.292929
F-1 score: 0.277695

max_depth=5:
Accuracy: 0.313131
Precision: 0.248293
Recall: 0.313131
F-1 score: 0.276968

max_depth=6:
Accuracy: 0.232323
Precision: 0.221139
Recall: 0.232323

F-1 score: 0.226593

max_depth=7:
Accuracy: 0.252525
Precision: 0.263158
Recall: 0.252525
F-1 score: 0.257732

Best max_depth with 500 data: 4

- **Min_sample split Decision Tree**:

min_samples_split=2:
Accuracy: 0.262626
Precision: 0.288829
Recall: 0.262626
F-1 score: 0.275105

min_samples_split=3:
Accuracy: 0.262626
Precision: 0.295457
Recall: 0.262626
F-1 score: 0.278076

min_samples_split=4:
Accuracy: 0.242424
Precision: 0.278373
Recall: 0.242424
F-1 score: 0.259158

min_samples_split=5:
Accuracy: 0.202020
Precision: 0.247165
Recall: 0.202020
F-1 score: 0.222324

min_samples_split=6:
Accuracy: 0.212121
Precision: 0.260488
Recall: 0.212121
F-1 score: 0.233830

min_samples_split=7:
Accuracy: 0.222222
Precision: 0.271501
Recall: 0.222222
F-1 score: 0.244402

min_samples_split=8:
Accuracy: 0.242424
Precision: 0.288842
Recall: 0.242424
F-1 score: 0.263605

min_samples_split=9:
Accuracy: 0.272727
Precision: 0.321318
Recall: 0.272727
F-1 score: 0.295035

Best min_samples_split with 500 data: 9

- **Max feature for Decision Tree:**

max_feature=2:
Accuracy: 0.171717
Precision: 0.176163
Recall: 0.171717
F-1 score: 0.173912

max_feature=3:
Accuracy: 0.242424

Precision: 0.250973
Recall: 0.242424
F-1 score: 0.246625

max_feature=4:
Accuracy: 0.323232
Precision: 0.336422
Recall: 0.323232
F-1 score: 0.329695

max_feature=5:
Accuracy: 0.181818
Precision: 0.173274
Recall: 0.181818
F-1 score: 0.177443

max_feature=6:
Accuracy: 0.181818
Precision: 0.198675
Recall: 0.181818
F-1 score: 0.189873

max_feature=7:
Accuracy: 0.242424
Precision: 0.234552
Recall: 0.242424
F-1 score: 0.238423

max_feature=8:
Accuracy: 0.343434
Precision: 0.339261
Recall: 0.343434
F-1 score: 0.341335

max_feature=9:
Accuracy: 0.262626
Precision: 0.282605
Recall: 0.262626
F-1 score: 0.272250

Best max_feature with 500 data: 9

Result

	Accuracy	Precision	Recall	F-1 score
KNN with 600 samples	0.180000	0.196027	0.180000	0.187672
KNN with 1000 samples	0.180000	0.209722	0.180000	0.193728
Decision-Tree with 600 samples	0.290000	0.266118	0.290000	0.277546
Decision-Tree with 1000 samples	0.350000	0.405048	0.350000	0.375517

Output:
KNN_eval_600:
Accuracy: 0.180000
Precision: 0.196027
Recall: 0.180000
F-1 score: 0.187672

KNN_eval_1000:
Accuracy: 0.180000
Precision: 0.209722
Recall: 0.180000
F-1 score: 0.193728

DT_eval_600:
Accuracy: 0.290000
Precision: 0.266118
Recall: 0.290000
F-1 score: 0.277546

DT_eval_1000:
Accuracy: 0.350000
Precision: 0.405048
Recall: 0.350000
F-1 score: 0.375517

Comparison and Conclusion

Since we have labels from 10 different intervals(classes), and we use the weight(number) of true class to do the calculations of accuracy and recall, so they are the same. We want to compare two models--K-nearest-neighbor and Decision-Tree in predicting data with multiple class labels. From the results obtained by using 600 and 1000 random data as unbalanced dataset, the model trained on decision trees performs better than K-NN. We only have 15 features in total and we choose 12 of them to evaluating the models, and the range for most features

are small, so maybe it is the reason why K-NN has lower precision than Decision Tree.