CS376
Machine Learning
Term Project Notice
(modifications)

School of Computing

2018. 11. 08.

Project Goal: Apartment Prices Forecast

- Goal: Predict the prices of apartments in Gotham city.

- You are given apartment prices in Gotham city from 1980-1992 with features of apartments.
- Use this data to estimate apartment prices in Gotham city.
- Training and test datasets are uploaded in KLMS.

- You can use any machine learning algorithm including the ones we learned in class.
  - Linear regression, multi-layer perceptron, xgboost, …

Data Description

- Both the train data and the test data are in csv format.
- Each row represents one transaction.
- The last column in the train data is the actual transaction price.
- The remaining columns are the features of the transaction.
- For test data, only the features are provided.
  - You should predict the prices of the apartments using the features of test data.

- For the input features, data entry for a particular column may be empty depending on the row.
- The reliability of each feature is hidden.
  - Some features might be irrelevant.
  - Some features might be noisy.
  - Some features may be related to each other.
  - You should decide which features to use.

Description of Features

Index	Description	Categorical data
0	Contract date	False
1	Latitude	False
2	Longitude	False
3	Altitude	False
4	1st class region id	True
5	2nd class region id	True
6	Road id	True
7	Apartment id	True
8	Floor	False
9	Angle: The direction the house was built (South, East, West, North…)	False
10	Area: Exclusive area for the traded house	False
11	The limit number of car of parking lot.	False
12	The total area of parking lot.	False

Description of Features

Index	Description	Categorical data
13	Whether an external vehicle can enter the parking lot	True
14	The average management fee of the apartment	False
15	The number of households in the apartment	False
16	Average age of residents	False
17	Builder id	True
18	Construction completion date	False
19	Built year	False
20	The number of schools near the apartment	False
21	The number of bus stations near the apartment	False
22	The number of subway stations near the apartment	False

Scoring

- Performance (60%)

$$1 - \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t}\right|$$

  - N: # of test samples, A_t: actual value, F_t: predicted value
- Uniqueness (30%)
  - Each team should write about unique methods on the report.

$$\frac{1}{\sum_{i \in UM} N_i} * \frac{score\_unique - score\_base}{score\_upper - score\_base}$$

    - UM: a set of unique methods your team has chosen.
    - N_i: total number of teams that used method 'i'
    - score_upper: test score upper bound.
    - score_base: test score without your unique methods.
    - score_unique: test score with your unique methods.
- Clarity (10%)
  - Each team will lose points for unclear writing. (ex. inconsistent use of equations, errata, totally inaccessible, etc.)

Schedule

- Team Assignment: 6th November, 2018
- Term Project Start: 8th November, 2018
- Simulation Test: 30th November, 2018
  - Only for those who want to participate
  - We'll evaluate the performance of your submission.
  - Not graded.
  - We'll open a submission page around the test date.
- Due: 14th December, 2018 (End of the semester)
  - No late submission will be accepted.

Submission Guidelines

- You should submit:
  - A report
  - 2 answer sheets
  - Source codes
  - Any materials necessary for reproduction.

- Report
  - Write it as clear as possible.
  - Follow the instructions in the report format (uploaded in KLMS).

- Answer sheet
  - Submit 2 answer sheets: predicted values on the test data with/without your unique methods.
  - File names of the answer sheets: 'unique.csv' (with the unique methods)
    'base.csv' (without the unique methods)
  - Specify predicted values for each line in the answer sheet.
  - The number and the order of lines in each answer sheet must be the same as the test data.

Submission Guidelines

- Source codes
  - Use python 3.5 or above.
  - Specify the libraries and their versions used in the code.
    - Describe the purpose of each library you used.
    - Do not import unnecessary libraries into your code.
  - The execution results must be reproducible.
    - Make it possible to reproduce your results using random seed.
  - Divide the entire code into a training part and test part.
    - Indicate which parts are for test.
  - You should submit the trained model file.
    - Your test code should perform prediction based on this saved model file.
  - Test codes
    - Include the minimum codes that perform the prediction using the trained model.
    - Describe how TAs can test with another test set.
    - Report test execution time.

Submission Guidelines

- Submissions are only accepted via KLMS.
- Plagiarism will not be tolerated by University rules.
  - Plagiarism check program will be used to check for plagiarism.

Virtual Machine Usage

- Technically, you should use your own computers/resources to develop your project.
- However, you might be falling short of computing resources, so we will provide you with Linux virtual machines.
- Specs
  - OS: Ubuntu 16.04.1 LTS
  - CPU: 8 cores, 2600MHz
  - RAM: 8GB
  - Storage: ~100GB
  - NO GPU
- Connecting
  - Each team will be assigned a single virtual machine with specs mentioned above.
  - The IP addresses and the designated port numbers are listed in the table below.
  - For example, you can log in to a machine using the ssh port with the following command:
    # ssh team15@143.248.140.251 -p 6422
  - Make sure to change the password to one of your choice after you've logged in.
- Setup
  - The VMs do not have anything pre-installed except the stuff shipped with ubuntu.
  - We recommend setting up a virtual environment for python(e.g. pyenv), Anaconda etc.

List of Machines

Team #	ID	PWD	IP Address	Service Ports (ssh, http, rdp)
1	team01	team01	143.248.140.250	5022, 5080, 5089
2	team02	team02	143.248.140.251	5122, 5180, 5189
3	team03	team03	143.248.140.251	5222, 5280, 5289
4	team04	team04	143.248.140.251	5322, 5380, 5389
5	team05	team05	143.248.140.251	5422, 5480, 5489
6	team06	team06	143.248.140.251	5522, 5580, 5589
7	team07	team07	143.248.140.251	5622, 5680, 5689
8	team08	team08	143.248.140.251	5722, 5780, 5789
9	team09	team09	143.248.140.251	5822, 5880, 5889
10	team10	team10	143.248.140.251	5922, 5980, 5989
11	team11	team11	143.248.140.251	6022, 6080, 6089
12	team12	team12	143.248.140.251	6122, 6180, 6189
13	team13	team13	143.248.140.251	6222, 6280, 6289

List of Machines

Team #	ID	PWD	IP Address	Service Ports (ssh, http, rdp)
14	team14	team14	143.248.140.251	6322, 6380, 6389
15	team15	team15	143.248.140.251	6422, 6480, 6489
16	team16	team16	143.248.140.251	6522, 6580, 6589
17	team17	team17	143.248.140.251	6622, 6680, 6689
18	team18	team18	143.248.140.251	6722, 6780, 6789
19	team19	team19	143.248.140.251	6822, 6880, 6889
20	team20	team20	143.248.140.251	6922, 6980, 6989
21	team21	team21	143.248.140.251	7022, 7080, 7089
22	team22	team22	143.248.140.251	7122, 7180, 7189
23	team23	team23	143.248.140.251	7222, 7280, 7289
24	team24	team24	143.248.140.251	7322, 7380, 7389
25	team25	team25	143.248.140.251	7422, 7480, 7489
26	team26	team26	143.248.140.251	7522, 7580, 7589

Some Hints for Term Project

- There are some missing values in features.
  - You should handle these missing values.
    If not, you probably won't get the right performance.
  - There are many ways to fill a missing value.
    - Replace the missing value with the average value of the feature.
    - Fill missing values with imputer (autoencoder, etc).
    - Use models handling missing values.
    - …

- Divide the train data into training and validation set to prevent overfitting.
  - K-fold is one of the good options.
  - Note: most of the test data is to predict the future.
    - Refer to this when splitting validation code.

- Test runs with several basic algorithms on the given dataset gave the score of 0.6~0.9
  - We didn't tune hyper-parameters.
  - If hyper-parameter tuning is done, the results may be better.