ECOM90025 Advanced Data Analysis - Tutorial 1

Zheng Fan

July 31, 2023

1 ECOM90025 ADA: Tutorial 1 in Week 2

- Covering materials for week 1 lecture
- Goal: learn some basic commands for Python.

2 My contact:

- Name: Zheng Fan
- Email: fan.z@unimelb.edu.au
- Send me an email if you have any general or conceptual questions. Code or technique issues, that are difficult to answer in an email, should be raised in a consultation.
- I'm also happy to stay for a while after the tutorial.
- Consultataion: meet Dr Yong Song every Tuesday 1-2pm at FBE 360.
- Special consideration: visit Stop 1.

3 Tutorial attendance

- You need to actively participate instead of just showing up.
- Attendance does not guarantee marks.
- You may get a maximum of 10 points out of 11 tutorials.
- If you unable to come to school, seek help from Stop 1.

4 Software:

- Google Colab is a free online platform where you can execute your code (especially for Python) and write text (LaTeX and Html) without any software installed.
- Local Jupyter Lab, which has been demonstrated during the lecture.
- Local Jupyter Notebook from Anaconda (My personal preference). Just download Anaconda, and then open Jupyter Notebook.
- Other IDE such as PyCharm, but I'm not very comfortable with using PyCharm. From what I know, the code saved in PyCharm is .py but not .ipynb, which may requires some conversion.

Although you can always use Colab, It is recommended to have a local machine installed.

5 Tutorial Questions

You have seen the popular student competition case.

Use the unconditional mean to start prediction.

1. Make a copy of this file.
2. Read the training sample to a Pandas dataframe.
3. compute the unconditional mean of the sale price.
4. Submit a file and get your Kaggle score screenshot.
5. Show your screenshot in the notebook.

Use chatGPT for help.

6 Question 1: Make a copy of this file.

- Should be very straightforward.

7 Question 2: Read the training sample to a Pandas dataframe.

[1]:
```
# import pandas to read csv, calculate summary statistics and save csv
import pandas as pd
```

7.1 load data file from local drive

- It is very convenient.
- But make sure you always update file path across device.

[2]:
```
# as an example
df_train = pd.read_csv("house-prices-advanced-regression-techniques/train.csv")
```

- note that the folder "house-prices-advanced-regression-techniques" is in the same folder as the python code file. Otherwise, you may need to properly nevigate to the target folder.

[3]:
```
# Let's look at whether the data has been successfully loaded
df_train.head(3)
```

[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	

	LandContour	Utilities	…	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	\
0	Lvl	AllPub	…	0	NaN	NaN	NaN	0	2	
1	Lvl	AllPub	…	0	NaN	NaN	NaN	0	5	

```
2         Lvl    AllPub  …        0    NaN   NaN         NaN       0      9

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500

[3 rows x 81 columns]
```

- It seems we do successfully load the data file

7.2 load data file from Dropbox

- You first need to create a shared link, and change the last number from 0 to 1, which allows download

[4]:
```
train_file_path = "https://www.dropbox.com/scl/fi/fbl9o4ni02cwn3k0guo2d/train.
↪csv?rlkey=i8ne0y692cca3km160rhqoe87&dl=1"
```

[5]:
```
df_train = pd.read_csv(train_file_path)
```

[6]:
```
# Let's look at whether the data has been successfully loaded
df_train.head(3)
```

[6]:
```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
0   1          60       RL         65.0     8450   Pave   NaN      Reg
1   2          20       RL         80.0     9600   Pave   NaN      Reg
2   3          60       RL         68.0    11250   Pave   NaN      IR1

  LandContour Utilities  … PoolArea PoolQC Fence MiscFeature MiscVal MoSold  \
0         Lvl    AllPub  …        0    NaN   NaN         NaN       0      2
1         Lvl    AllPub  …        0    NaN   NaN         NaN       0      5
2         Lvl    AllPub  …        0    NaN   NaN         NaN       0      9

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500

[3 rows x 81 columns]
```

- It seems we do successfully load the data file

7.3 load data file from Google drive

- convenient to use in Google Colab just as shown in the lecture material
- not going to demonstrate, as it has already been shown in the lecture

- check my Google Colab file https://colab.research.google.com/drive/1AuxeUm6nGe8QOlbHRsdQaDanVSPg

However, I personally don't really like this. It alwasy requires connecting to google drive. In addition, it is hard for other people to replicate your work. I always prefer to read open shared link data files.

8 Question 3: compute the unconditional mean of the sale price.

- Here we use the unconditional mean as our prediction.
- No unique solution.

Before do the calculation, let's first get familar with the data set

[7]:
```
# the dimension of the data set
df_train.shape
```

[7]:
```
(1460, 81)
```

[8]:
```
# we have checked the top of the data set, let's look at the bottom of the data␣
 ↪set
df_train.tail(3)
```

[8]:
```
        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
1457  1458          70       RL         66.0     9042   Pave   NaN      Reg
1458  1459          20       RL         68.0     9717   Pave   NaN      Reg
1459  1460          20       RL         75.0     9937   Pave   NaN      Reg

     LandContour Utilities  … PoolArea PoolQC  Fence MiscFeature MiscVal  \
1457         Lvl    AllPub  …        0    NaN  GdPrv        Shed    2500
1458         Lvl    AllPub  …        0    NaN    NaN         NaN       0
1459         Lvl    AllPub  …        0    NaN    NaN         NaN       0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[3 rows x 81 columns]
```

- Indeed, we have 1460 observations. 81 columns suggest for 81 variables.

[9]:
```
# compute the summary statistics
df_train['SalePrice'].describe()
```

[9]:
```
count      1460.000000
mean     180921.195890
std       79442.502883
```

```
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

- Note that the variable name is case-sensitive

[10]:
```
# what if we only need mean, standard deviation and max?
df_train['SalePrice'].describe().loc[['mean', 'std','max']]
```

[10]:
```
mean    180921.195890
std      79442.502883
max     755000.000000
Name: SalePrice, dtype: float64
```

[11]:
```
df_train['SalePrice'].describe().loc[['mean']]
```

[11]:
```
mean    180921.19589
Name: SalePrice, dtype: float64
```

[12]:
```
# directly get mean.
df_train['SalePrice'].mean()
```

[12]:
```
180921.19589041095
```

9 Question 4: Submit a file and get your Kaggle score screenshot.

- We first need to create a data file to be submit.
- Fortunately, Kaggle has already provide us with a sample submission file.
- we just need to update our prediction on that file and submit.

[13]:
```
# Let's load it from Dropbox
df_submission = pd.read_csv('https://www.dropbox.com/scl/fi/
 ↪x9kvsruzulw7v3oyp5mzq/sample_submission.csv?
 ↪rlkey=jv8msq35pewlmxmw0a1ehiu9h&dl=1')
```

[14]:
```
# check the sample data
df_submission.head(3)
```

[14]:
```
     Id      SalePrice
0  1461  169277.052498
1  1462  187758.393989
2  1463  183583.683570
```

- You may see the id starts from 1461. These are the observations not in our training data set.

- We are going to give our prediction on the SalePrice for those IDs.

[15]:
```
# replace current sale price with our prediction
df_submission['SalePrice'] = df_train['SalePrice'].mean()
```

[16]:
```
# Let's check the mean.
df_submission['SalePrice'].mean()
```

[16]:
```
180921.19589040437
```

- We have replace all the SalePrice into our predictions.

[17]:
```
df_submission.head(3)
```

[17]:

	Id	SalePrice
0	1461	180921.19589
1	1462	180921.19589
2	1463	180921.19589

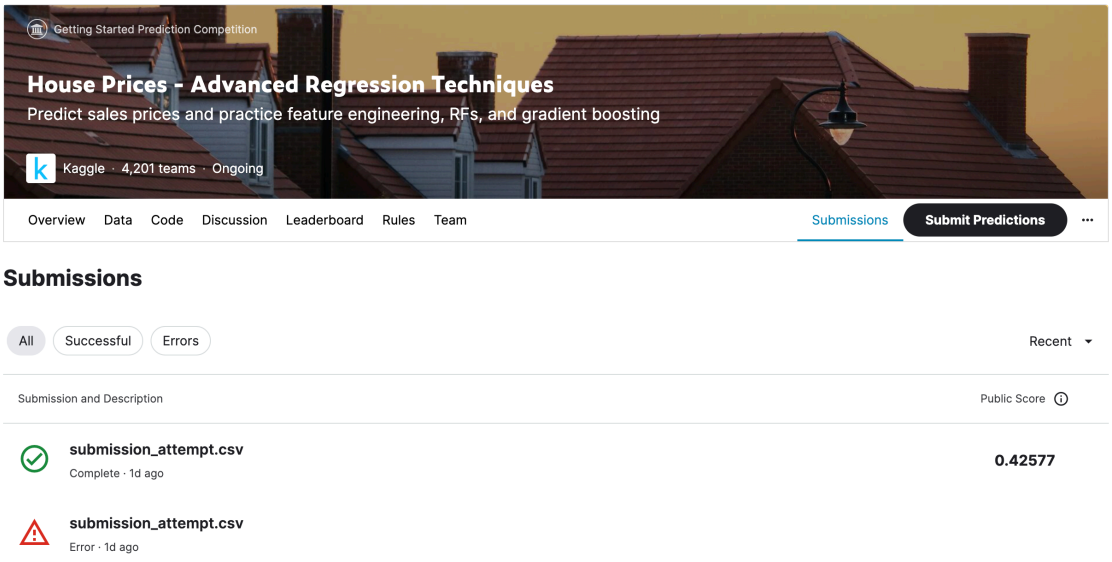
- Quite easily done. Save the file and submit!
- It would be much easier to save into Google Drive via Colab, although you still need to download the file and submit.
- I would prefer, simply save it to your local mechine if you python code is saved locally.
- If you are using Google Colab, again, check the code I shared earlier ago on how to save https://colab.research.google.com/drive/1AuxeUm6nGe8QOlbHRsdQaDanVSPgc1r0?usp=sharing

[18]:
```
df_submission.to_csv("submission_attempt_ver001.csv", index=False)
```

- It's important to set index = False. You can try and see what would happen.

10 Question 5: Show your screenshot in the notebook.

I propose two ways of attaching picture to Markdown. 1. `<img src='xxxx' width=700>` replcae xxxx with your open dropbox share link. 2. `![image info](xxxx.png)` save the screenshot to the same folder, and change the name xxxx into yours.



- You may see Error on my first submission, because I included redundant index in the data file.

11 All done! Let me or ChatGPT know if you have any questions.