

Module 1 Quiz

(Mayur Brijwani)

Question 1

127.0.0.1:8888/notebooks/homework/homework_module1.ipynb

jupyter

homework_module1 Last Checkpoint: 20 minutes ago

File Edit View Run Kernel Settings Help

Markdown ▾

```
[1]: import pandas as pd
import seaborn as sns # for visualization
import matplotlib.pyplot as plt # for visualization
from sklearn.feature_extraction import DictVectorizer # to create vector from dict
from sklearn.linear_model import LinearRegression # model
from sklearn.metrics import mean_squared_error # for calculating diff in actual and predicted
```

Q1. Downloading the data

Download the data for January and February 2023.

Read the data for January. How many columns are there?

- 15
- 17
- 18
- 19

Answer - 19

```
[17]: #Reading jan yellow taxi data


df = pd.read_parquet('../data/yellow_tripdata_2023-01.parquet')

#Number of columns


len(df.columns)

[17]: 19
```

Question 2

 jupyter homework_module1 Last Checkpoint: 21 minutes ago

File Edit View Run Kernel Settings Help

 Code

▼ **Q2. Computing duration**

What's the standard deviation of the trips duration in January?

- 32.59
- 42.59
- 52.59
- 62.59

Answer - 42.59

```
[18]: # Creating new column duration for seeing total time
df['duration'] = df.tpep_dropoff_datetime - df.tpep_pickup_datetime

#Converting new column into minutes for simplicity
df.duration = df.duration.apply(lambda td: td.total_seconds() / 60)

# Calculating Standard deviation
df.duration.std()
```

```
[18]: 42.594351241920904
```

Question 3

127.0.0.1:8888/notebooks/homework/homework_module1.ipynb

jupyter homework_module1 Last Checkpoint: 6 minutes ago

File Edit View Run Kernel Settings Help

JupyterLab Python 3 (ipyker

Q3. Dropping outliers

Next, we need to check the distribution of the duration variable. There are some outliers. Let's remove them and keep only the records where the duration was between 1 and 60 minutes (inclusive).

What fraction of the records left after you dropped the outliers?

- 90%
- 92%
- 95%
- 98%

Answer - 98%

```
[11]: # Dividing record between 1 and 60 inclusive with total original records and multiplying it with 100 for percentage of records Left

len(df[(df.duration >= 1) & (df.duration <= 60)])/len(df) * 100

[11]: 98.1220282212598
```

Question 4

Q4. One-hot encoding

Let's apply one-hot encoding to the pickup and dropoff location IDs. We'll use only these two features for our model.

Turn the dataframe into a list of dictionaries (remember to re-cast the ids to strings - otherwise it will label encode them)

Fit a dictionary vectorizer

Get a feature matrix from it

What's the dimensionality of this matrix (number of columns)?

- 2
- 155
- 345
- 515
- 715

Answer - 515

```
[13]: # Creating Categories
categorical = ['PULocationID', 'DOLocationID']

# Converting categories data type to string
df[categorical] = df[categorical].astype(str)

# Turning the dataframe into a list of dictionaries
train_dicts = df[categorical].to_dict(orient='records')

# Initializing dictionary vectorizer
dv = DictVectorizer()


# Fitting a dictionary vectorizer
X_train = dv.fit_transform(train_dicts)

# Extracting feature matrix
X_train.shape
```


```
[13]: (3009173, 515)
```

Question 5

127.0.0.1:8888/notebooks/homework/homework_module1.ipynb

 **jupyter** homework_module1 Last Checkpoint: 8 minutes ago

File Edit View Run Kernel Settings Help

 Code

Q5. Training a model

Now let's use the feature matrix from the previous step to train a model.

- Train a plain linear regression model with default parameters, where duration is the response variable
- Calculate the RMSE of the model on the training data

What's the RMSE on train?

- 3.64
- 7.64
- 11.64
- 16.64

Answer - 7.64

```
[14]: # Setting the target
target = 'duration'
y_train = df[target].values

# Training a plain linear regression model
lr = LinearRegression()
lr.fit(X_train, y_train)

# Predicting on training data
y_pred = lr.predict(X_train)

# Calculating the RMSE of the model on the training data
mean_squared_error(y_train, y_pred, squared=False)
```

[14]: 7.649261927665777

Question 6

books/homework/homework_module1.ipynb

Jupyter homework_module1 Last Checkpoint: 12 minutes ago

File Edit View Run Kernel Settings Help

📄 + 🔍 📄 ▶ ⏏ ⏪ ⏩ Code ▾

Q6. Evaluating the model

Now let's apply this model to the validation dataset (February 2023).

What's the RMSE on validation?

- 3.81
- 7.81
- 11.81
- 16.81

Answer - 7.81

```
[16]: df = pd.read_parquet('../data/yellow_tripdata_2023-02.parquet')

# Creating new column duration for seeing total time
df['duration'] = df.tpep_dropoff_datetime - df.tpep_pickup_datetime

# Converting new column into minutes for simplicity
df.duration = df.duration.apply(lambda td: td.total_seconds() / 60)
df = df[(df.duration >= 1) & (df.duration <= 60)]

# Creating Categories
categorical = ['PULocationID', 'DOLocationID']

# Converting categories data type to string
df[categorical] = df[categorical].astype(str)

# Turning the dataframe into a list of dictionaries
val_dicts = df[categorical].to_dict(orient='records')

# Transforming a dictionary vectorizer
X_val = dv.transform(val_dicts)

target = 'duration'
y_val = df[target].values

# Predicting on validation data

y_pred = lr.predict(X_val)

mean_squared_error(y_val, y_pred, squared=False)
```

[16]: 7.811817548344513