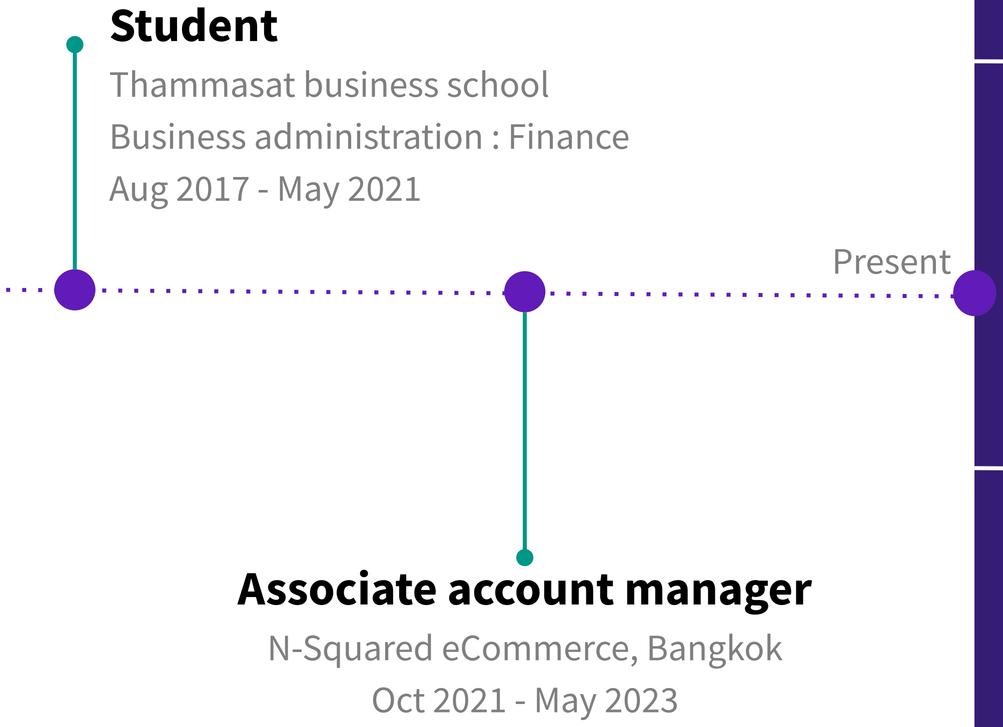
Naphon Santisukwongchot

Profile summary

Seeking a career transition into data science. Excellent understanding and proficiency of platforms for effective data analysis, including Excel, Python, R, and SQL. Strong communication, organizational and analytical skills.



Student

Thammasat business school

Business administration : Finance

Aug 2017 - May 2021

Present

Associate account manager

N-Squared eCommerce, Bangkok

Oct 2021 - May 2023

Technical strengths

Business Intelligence :	Looker, Power BI, Tableau
Data Analysis :	Pandas, NumPy
Data Visualization :	Matplotlib, Seaborn
Machine Learning :	Scikit-Learn
Microsoft Office :	Excel, PowerPoint, Word
Programming :	Python, R, SQL

Skills

◊ Attention to Detail

◊ Collaboration

◊ Problem Solving

◊ Business Acumen

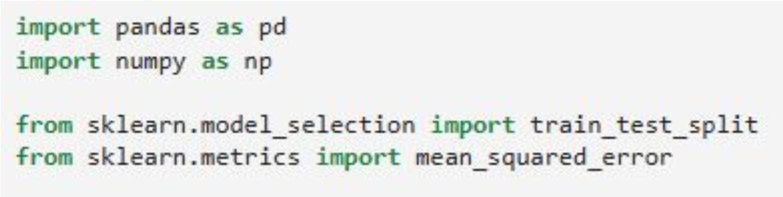
◊ Critical Thinking

◊ IELTS 6

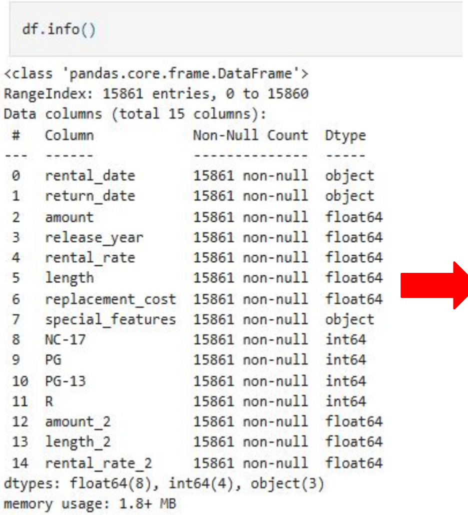
◊ Regression , Classification, Clustering

Project Predicting Movie Rental Durations (1)

A DVD rental company needs your help! They want to figure out how many days a customer will rent a DVD for based on some features. They want you to try out some regression models which will help predict the number of days a customer will rent a DVD. **The company wants a model which yields a MSE** of 3 or less on a test set. The model you make will help the company become more efficient inventory planning.

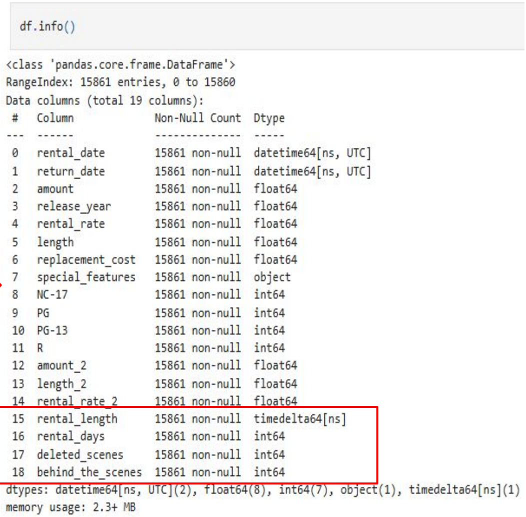


```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15861 entries, 0 to 15860
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   rental_date       15861 non-null  object
 1   return_date       15861 non-null  object
 2   amount            15861 non-null  float64
 3   release_year      15861 non-null  float64
 4   rental_rate       15861 non-null  float64
 5   length            15861 non-null  float64
 6   replacement_cost  15861 non-null  float64
 7   special_features  15861 non-null  object
 8   NC-17             15861 non-null  int64
 9   PG                15861 non-null  int64
 10  PG-13             15861 non-null  int64
 11  R                 15861 non-null  int64
 12  amount_2          15861 non-null  float64
 13  length_2          15861 non-null  float64
 14  rental_rate_2     15861 non-null  float64
dtypes: float64(8), int64(4), object(3)
memory usage: 1.8+ MB
```



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15861 entries, 0 to 15860
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   rental_date       15861 non-null  datetime64[ns, UTC]
 1   return_date       15861 non-null  datetime64[ns, UTC]
 2   amount            15861 non-null  float64
 3   release_year      15861 non-null  float64
 4   rental_rate       15861 non-null  float64
 5   length            15861 non-null  float64
 6   replacement_cost  15861 non-null  float64
 7   special_features  15861 non-null  object
 8   NC-17             15861 non-null  int64
 9   PG                15861 non-null  int64
 10  PG-13             15861 non-null  int64
 11  R                 15861 non-null  int64
 12  amount_2          15861 non-null  float64
 13  length_2          15861 non-null  float64
 14  rental_rate_2     15861 non-null  float64
 15  rental_length     15861 non-null  timedelta64[ns]
 16  rental_days       15861 non-null  int64
 17  deleted_scenes    15861 non-null  int64
 18  behind_the_scenes 15861 non-null  int64
dtypes: datetime64[ns, UTC](2), float64(8), int64(7), object(1), timedelta64[ns](1)
memory usage: 2.3+ MB
```

Exploratory data analysis

◊ Import frameworks and csv file

◊ Perform EDA : df.head(), df.info(), df.describe

◊ Set a target variable

- Add rental_length column
- Add rental_days column : Target

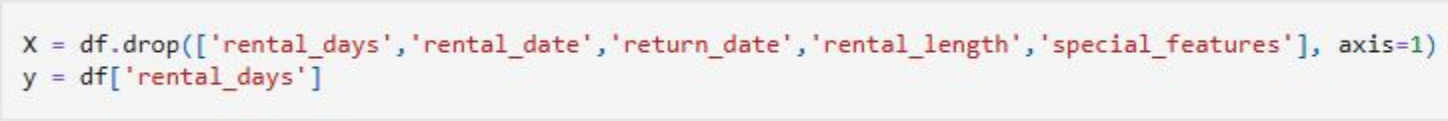
◊ Categorize special features into one hot encoder

- Add deleted_scenes column : Feature
- Add behind_the_scenes column : Feature

Project Predicting Movie Rental Durations (2)

Feature Selection

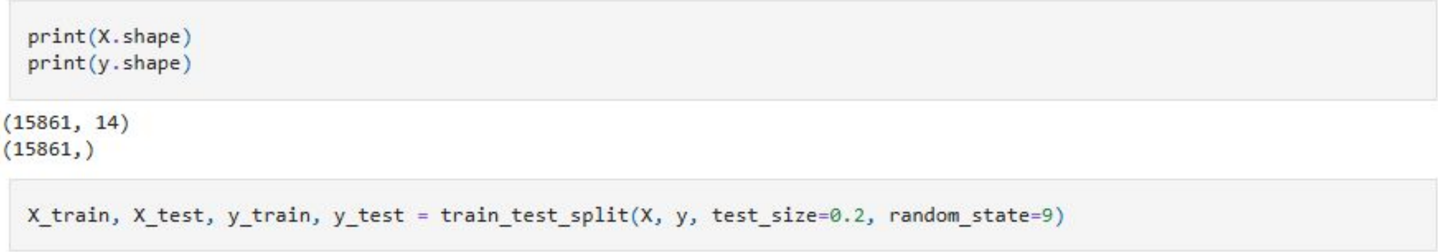
- ◊ removing irrelevant features
  - - Assign relevant features into X
- ◊ Assign rental_days (target) into Y



```
X = df.drop(['rental_days','rental_date','return_date','rental_length','special_features'], axis=1)
y = df['rental_days']
```

Data implementation

- ◊ Checking data set dimension
- ◊ Perform train test split



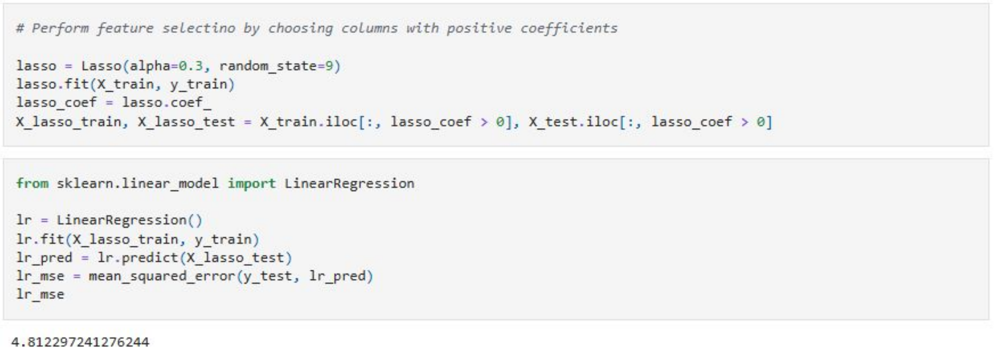
```
print(X.shape)
print(y.shape)
```

```
(15861, 14)
(15861,)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=9)
```

Project Predicting Movie Rental Durations (3)

Linear (lasso)

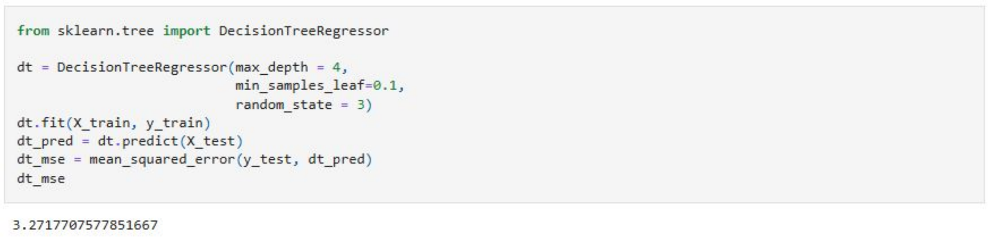


```
# Perform feature selectino by choosing columns with positive coefficients

lasso = Lasso(alpha=0.3, random_state=9)
lasso.fit(X_train, y_train)
lasso_coef = lasso.coef_
X_lasso_train, X_lasso_test = X_train.iloc[:, lasso_coef > 0], X_test.iloc[:, lasso_coef > 0]
```

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X_lasso_train, y_train)
lr_pred = lr.predict(X_lasso_test)
lr_mse = mean_squared_error(y_test, lr_pred)
lr_mse
```

```
4.812297241276244
```

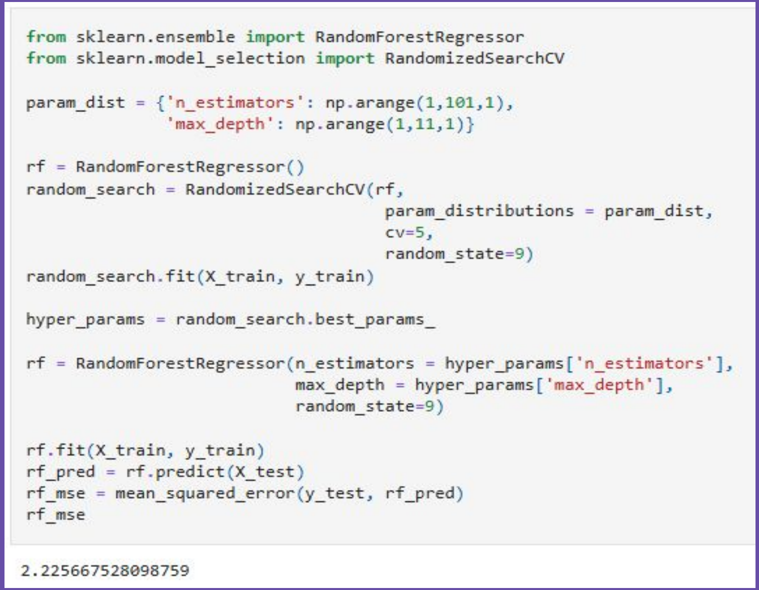
Decision tree



```
from sklearn.tree import DecisionTreeRegressor

dt = DecisionTreeRegressor(max_depth = 4,
                           min_samples_leaf=0.1,
                           random_state = 3)
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)
dt_mse = mean_squared_error(y_test, dt_pred)
dt_mse
```

```
3.2717707577851667
```

Random forest



```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import RandomizedSearchCV

param_dist = {'n_estimators': np.arange(1,101,1),
              'max_depth': np.arange(1,11,1)}

rf = RandomForestRegressor()
random_search = RandomizedSearchCV(rf,
                                   param_distributions = param_dist,
                                   cv=5,
                                   random_state=9)
random_search.fit(X_train, y_train)

hyper_params = random_search.best_params_

rf = RandomForestRegressor(n_estimators = hyper_params['n_estimators'],
                           max_depth = hyper_params['max_depth'],
                           random_state=9)

rf.fit(X_train, y_train)
rf_pred = rf.predict(X_test)
rf_mse = mean_squared_error(y_test, rf_pred)
rf_mse
```

```
2.225667528098759
```

MSE calculation

◊ Perform machine learning

- Linear (lasso) :	MSE = 4.812
- Decision tree :	MSE = 3.271
- Random Forest :	MSE = 2.225 ☺

Contact

Naphon Santisukwongchot
emoney_euro@hotmail.com
(+66)89 738 3632

https://www.linkedin.com/in/naphon1999/
https://github.com/naphon1999
https://www.datacamp.com/portfolio/naphon1999
https://drive.google.com/drive/folders/1-3x_-Xmho03z5u3PA6VKZi2-nY90oixK?usp=sharing

Portfolio reference

https://drive.google.com/file/d/1nS9qUg9F65z3MXSeZQoGUH-U2ZeuNgZq/view?usp=drive_link

Certifications & Developments

Data Science Bootcamp 10 :	DataRockie
Data Analyst in SQL & Python :	DataCamp
Google Advanced Data Analytics :	Google
IBM Data Science:	IBM
Machine Learning :	DeepLearning.AI

Work achievement

◊ Achieve campaign sales target
◊ Completely release aging stock