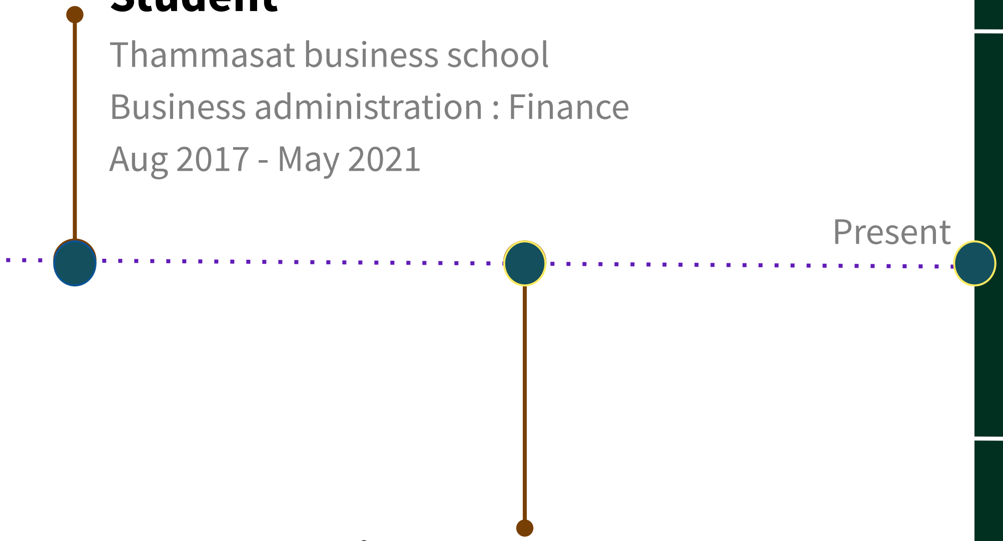
Naphon Santisukwongchot

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

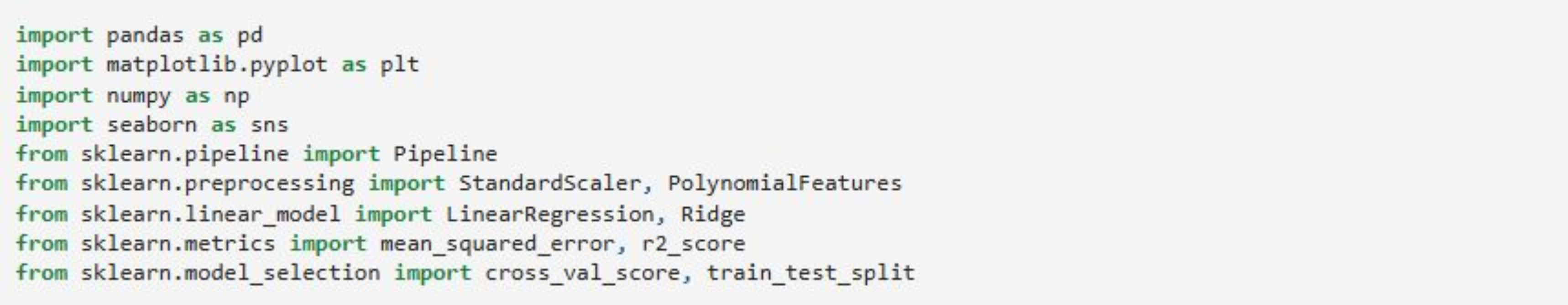
- ◊ Attention to Detail
- ◊ Collaboration
- ◊ Problem Solving
- ◊ Regression , Classification, Clustering
- ◊ Business Acumen
- ◊ Critical Thinking
- ◊ IELTS 6

Insurance Cost Analysis (1)

Importing library

Import frameworks

pandas, numpy :	Data manipulation
matplotlib, seaborn :	Data visualization
Sklearn :	Machine learning



```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score, train_test_split
```

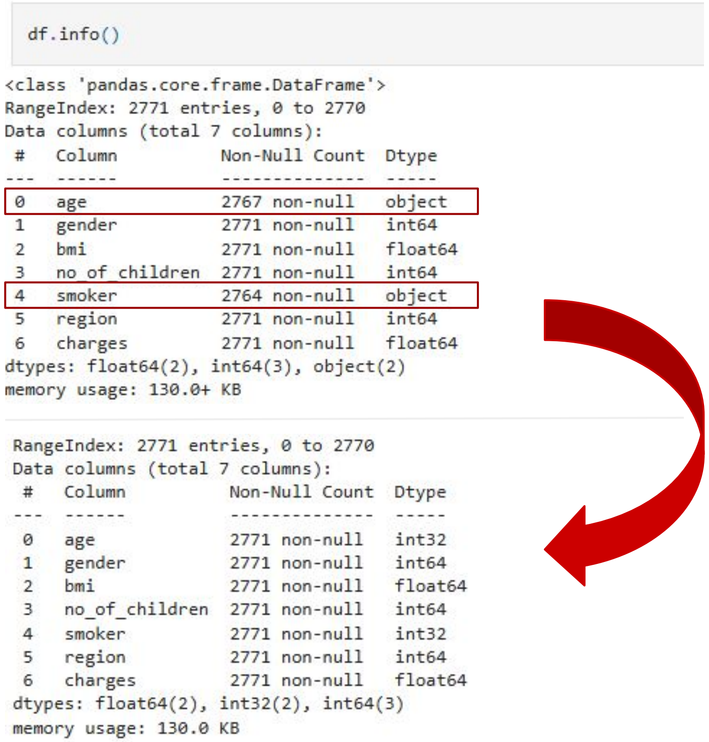
Insurance Cost Analysis (2)

Data wrangling

- df.info() : identify 'Null' in columns.
- There are missing values in age and smoker column

Handle missing data

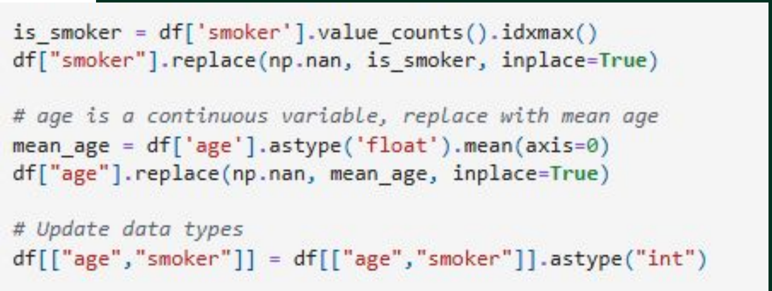
- Continuous attributes (age), replace with **mean**
- Categorical attributes (smoker), replace with **mode**
- Update data types



```
df.info()
```

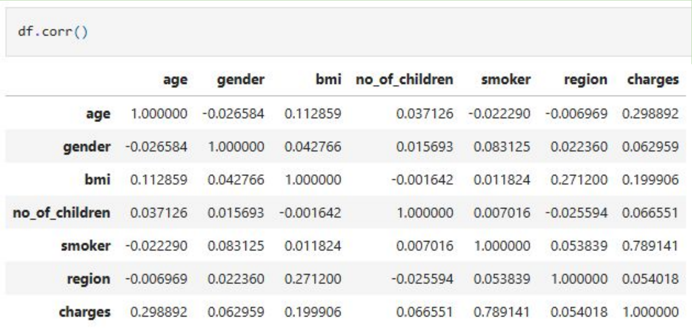
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2771 entries, 0 to 2770
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             2767 non-null   object 
 1   gender          2771 non-null   int64  
 2   bmi             2771 non-null   float64
 3   no_of_children  2771 non-null   int64  
 4   smoker          2764 non-null   object 
 5   region          2771 non-null   int64  
 6   charges         2771 non-null   float64
dtypes: float64(2), int64(3), object(2)
memory usage: 130.0+ KB
```

```
RangeIndex: 2771 entries, 0 to 2770
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             2771 non-null   int32  
 1   gender          2771 non-null   int64  
 2   bmi             2771 non-null   float64
 3   no_of_children  2771 non-null   int64  
 4   smoker          2771 non-null   int32  
 5   region          2771 non-null   int64  
 6   charges         2771 non-null   float64
dtypes: float64(2), int32(2), int64(3)
memory usage: 130.0 KB
```



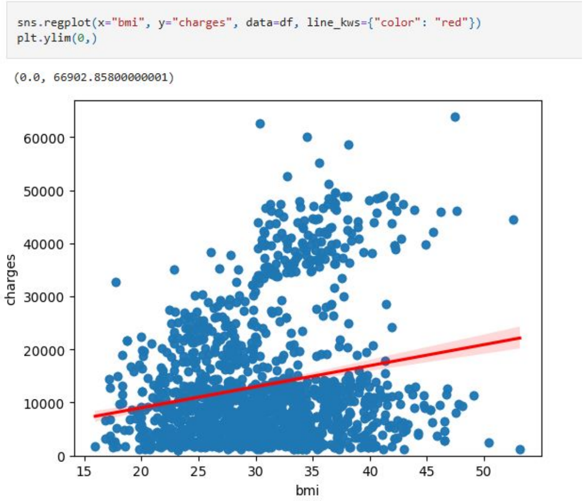
```
is_smoker = df['smoker'].value_counts().idxmax()
df["smoker"].replace(np.nan, is_smoker, inplace=True)

# age is a continuous variable, replace with mean age
mean_age = df['age'].astype('float').mean(axis=0)
df["age"].replace(np.nan, mean_age, inplace=True)

# Update data types
df[["age","smoker"]] = df[["age","smoker"]].astype("int")
```

Insurance Cost Analysis (3)



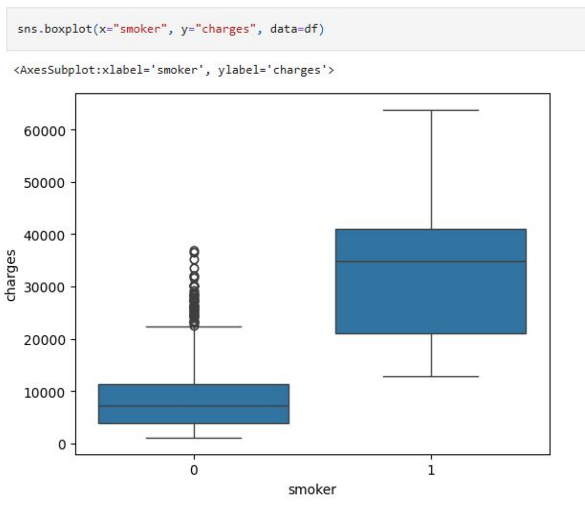
```
df.corr()
```

	age	gender	bmi	no_of_children	smoker	region	charges
age	1.000000	-0.026584	0.112859	0.037126	-0.022290	-0.006969	0.298892
gender	-0.026584	1.000000	0.042766	0.015693	0.083125	0.022360	0.062959
bmi	0.112859	0.042766	1.000000	-0.001642	0.011824	0.271200	0.199906
no_of_children	0.037126	0.015693	-0.001642	1.000000	0.007016	-0.025594	0.066551
smoker	-0.022290	0.083125	0.011824	0.007016	1.000000	0.053839	0.789141
region	-0.006969	0.022360	0.271200	-0.025594	0.053839	1.000000	0.054018
charges	0.298892	0.062959	0.199906	0.066551	0.789141	0.054018	1.000000



```
sns.regplot(x="bmi", y="charges", data=df, line_kws={"color": "red"})
plt.ylim(0,)
```

```
(0.0, 66902.85800000001)
```



```
sns.boxplot(x="smoker", y="charges", data=df)
```

```
<AxesSubplot:xlabel='smoker', ylabel='charges'>
```

Exploratory data analysis

◊ df.corr() : compute correlation across columns

◊ sns.regplot : visualize the relationship between ‘bmi’ and ‘charges’

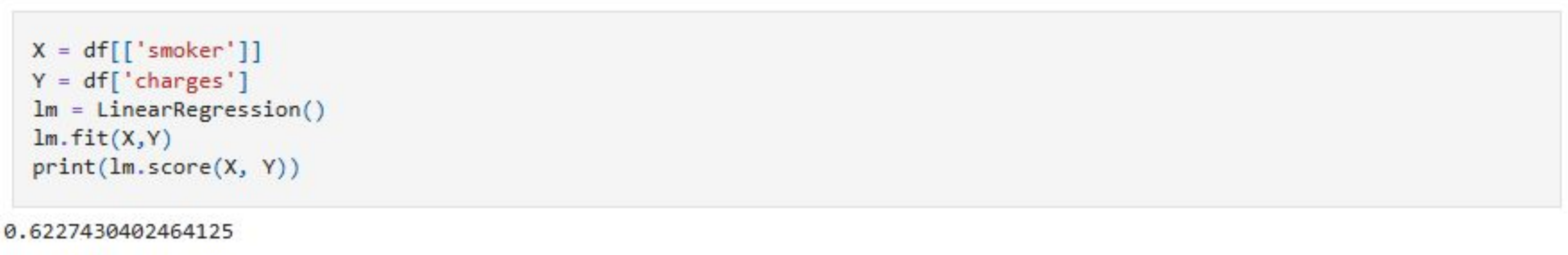
◊ sns.boxplot : visualize the distribution, spread, and outliers in smoker category

Insurance Cost Analysis (4)

Conducting model

◊ Conduct linear regression model

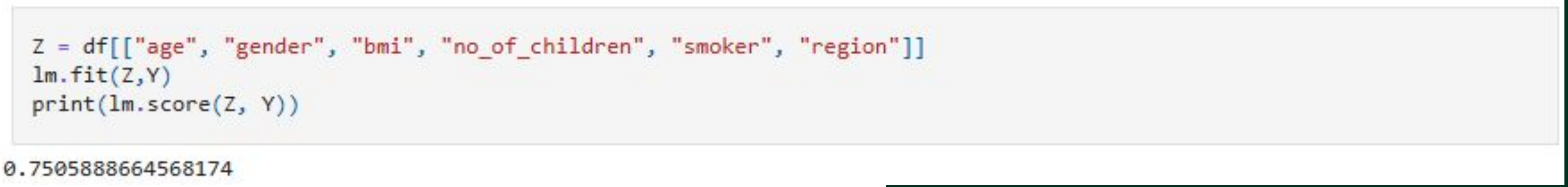
◊ Using **only** **‘smoker’** to predict **‘charges’** : R2 = 0.62



```
X = df[['smoker']]
Y = df['charges']
lm = LinearRegression()
lm.fit(X,Y)
print(lm.score(X, Y))
```

```
0.6227430402464125
```

◊ Using **‘All features’** to predict **‘charges’** : R2 = 0.75



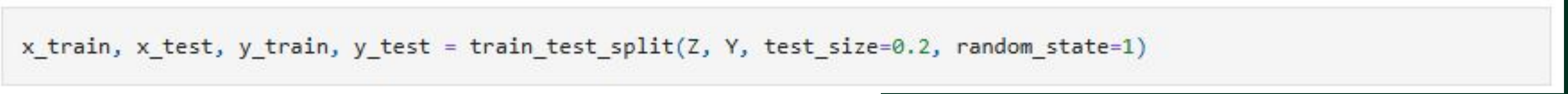
```
Z = df[["age", "gender", "bmi", "no_of_children", "smoker", "region"]]
lm.fit(Z,Y)
print(lm.score(Z, Y))
```

```
0.7505888664568174
```

Insurance Cost Analysis (5)

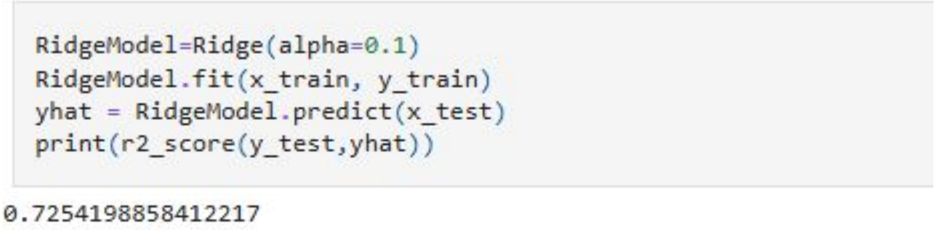
Model refinement

◊ Split the data into training and test set, 20%



```
x_train, x_test, y_train, y_test = train_test_split(Z, Y, test_size=0.2, random_state=1)
```

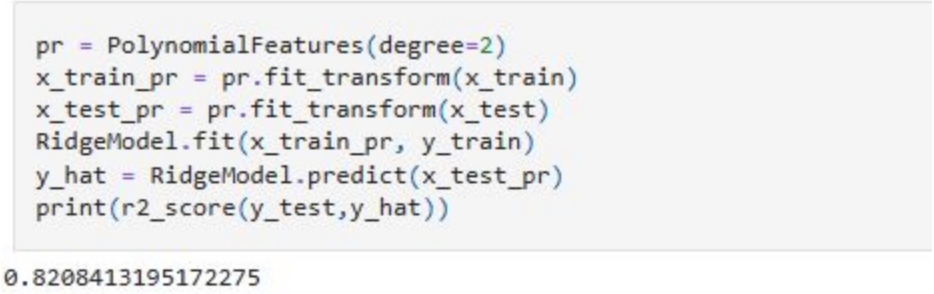
◊ Conduct **ridge regression** with **alpha = 0.1** : R2 = 0.73



```
RidgeModel=Ridge(alpha=0.1)
RidgeModel.fit(x_train, y_train)
yhat = RidgeModel.predict(x_test)
print(r2_score(y_test,yhat))
```

```
0.7254198858412217
```

◊ Perform **polynomial transformation** with **degree = 2** : R2 = 0.82



```
pr = PolynomialFeatures(degree=2)
x_train_pr = pr.fit_transform(x_train)
x_test_pr = pr.fit_transform(x_test)
RidgeModel.fit(x_train_pr, y_train)
y_hat = RidgeModel.predict(x_test_pr)
print(r2_score(y_test,y_hat))
```

```
0.8208413195172275
```

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

Data Source

https://drive.google.com/file/d/1YbSAGYHV0sVyGRTY3eYdqn9nX-S3GCjQ/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