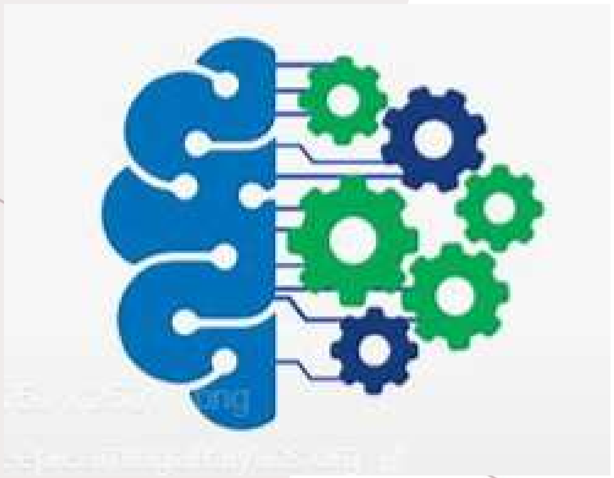
CAPSTONE PROJECT 4 – MACHINE LEARNING

BY LINDAWATI HENDRAWAN

PERSONAL MEDICAL COST PREDICTIONS

DATA SOURCE & INFORMATION

- https://www.kaggle.com/mirichoi0218/insurance
- Data of patients of Health insurance company at USA
- Content:
  - age: age of primary beneficiary
  - sex: insurance contractor gender, female, male
  - bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
  - children: Number of children covered by health insurance / Number of dependents
  - smoker: Smoking
  - region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
  - charges: Individual medical costs billed by health insurance



OBJECTIVES

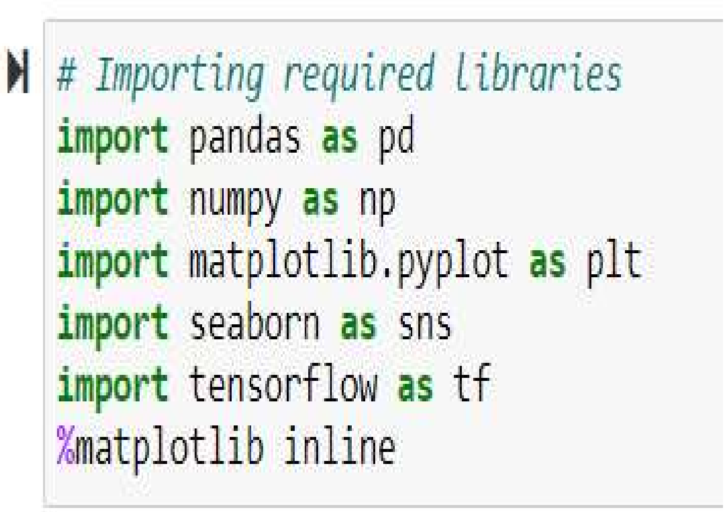
Build a Machine Learning Model to :

- To analyse the individual medical cost billed by Health Insurance
- To identify the individual with the risk of having higher medical cost
- To identify the factors that influence the increasing of medical cost
- To predict the medical cost

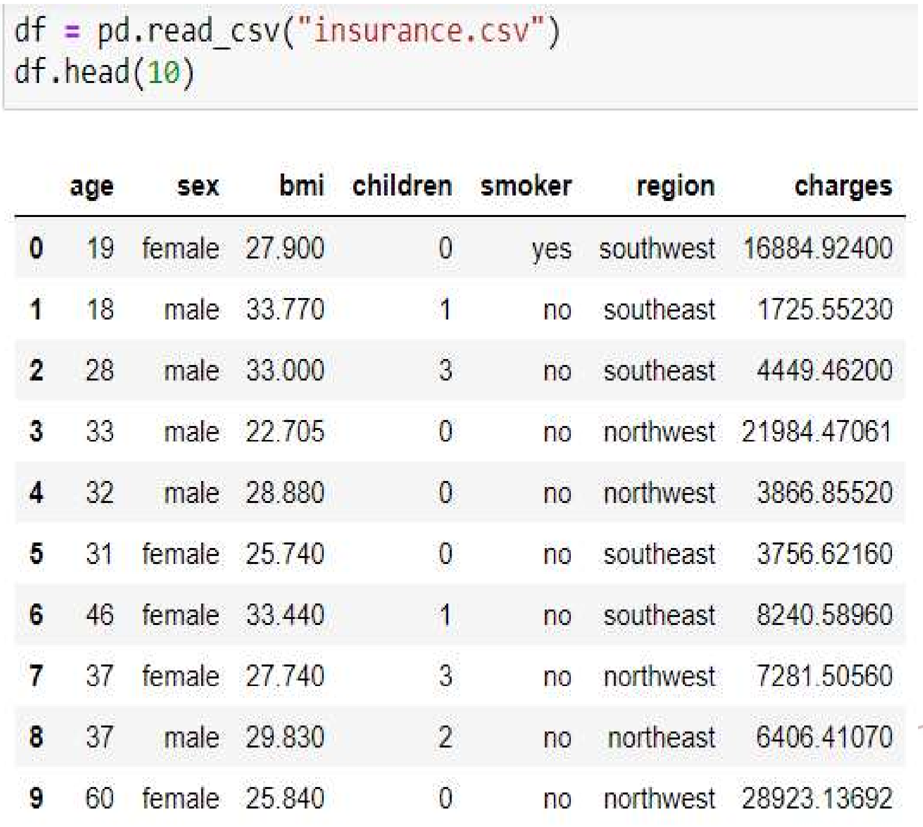
PROCESS WORKFLOWS

EDA AND DATA PREPARATION

- Import the required libraries
- Load data to Jupiter notebook



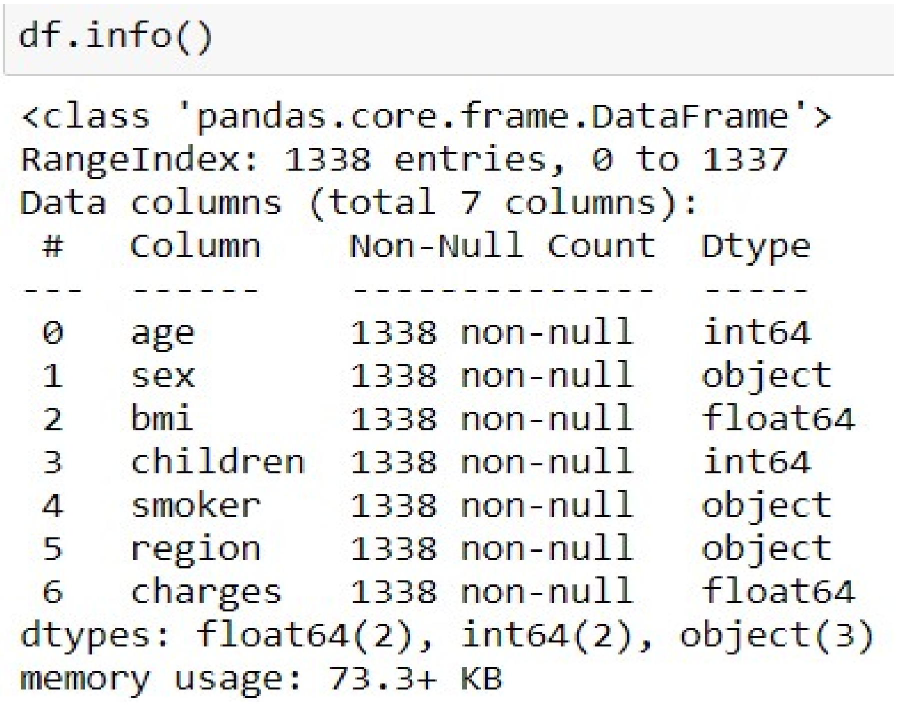
```
# Importing required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
%matplotlib inline
```



```
df = pd.read_csv("insurance.csv")
df.head(10)
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
5	31	female	25.740	0	no	southeast	3756.62160
6	46	female	33.440	1	no	southeast	8240.58960
7	37	female	27.740	3	no	northwest	7281.50560
8	37	male	29.830	2	no	northeast	6406.41070
9	60	female	25.840	0	no	northwest	28923.13692

EDA AND DATA PREPARATION

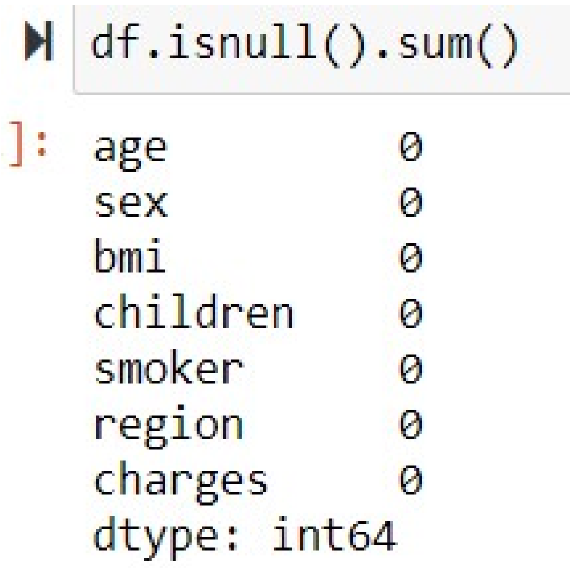


```
df.info()
```

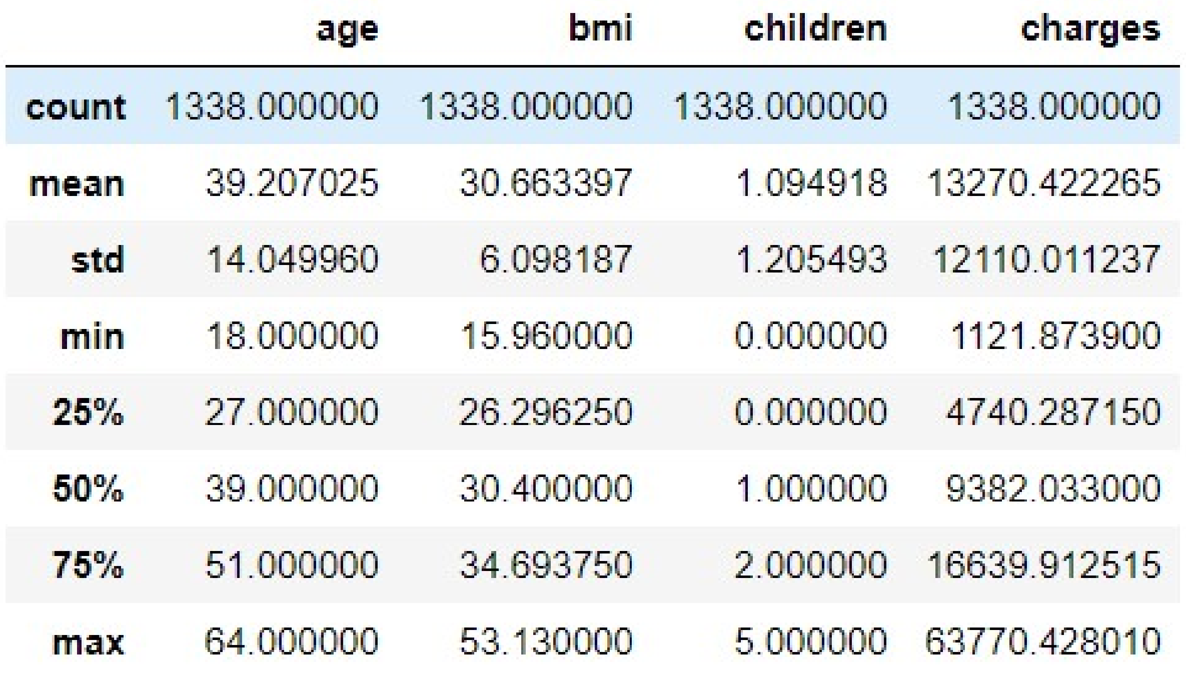
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

EDA AND DATA PREPARATION

. Data Cleaning and Data Description



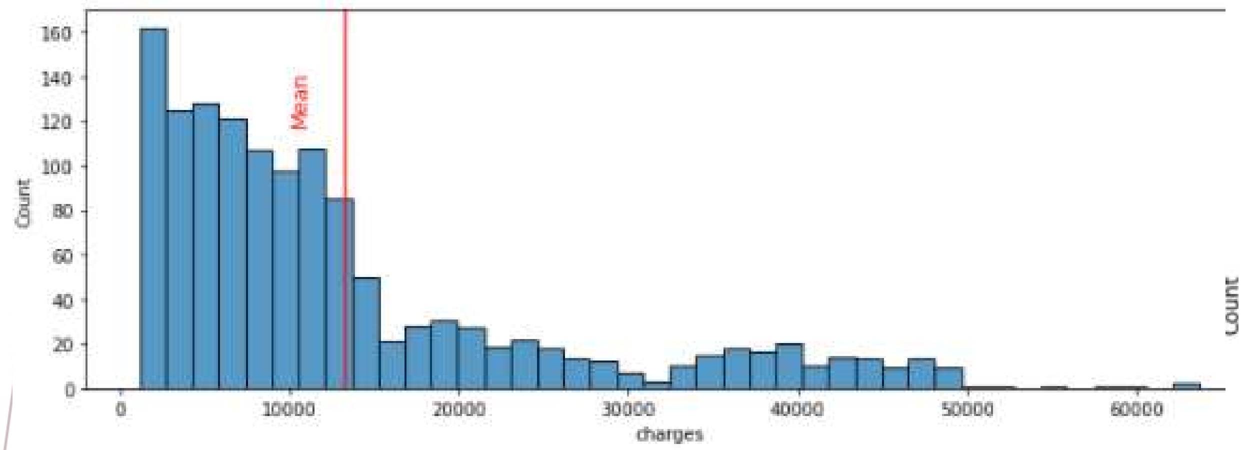
```
df.isnull().sum()

]: age         0
   sex         0
   bmi         0
   children    0
   smoker      0
   region      0
   charges     0
   dtype: int64
```



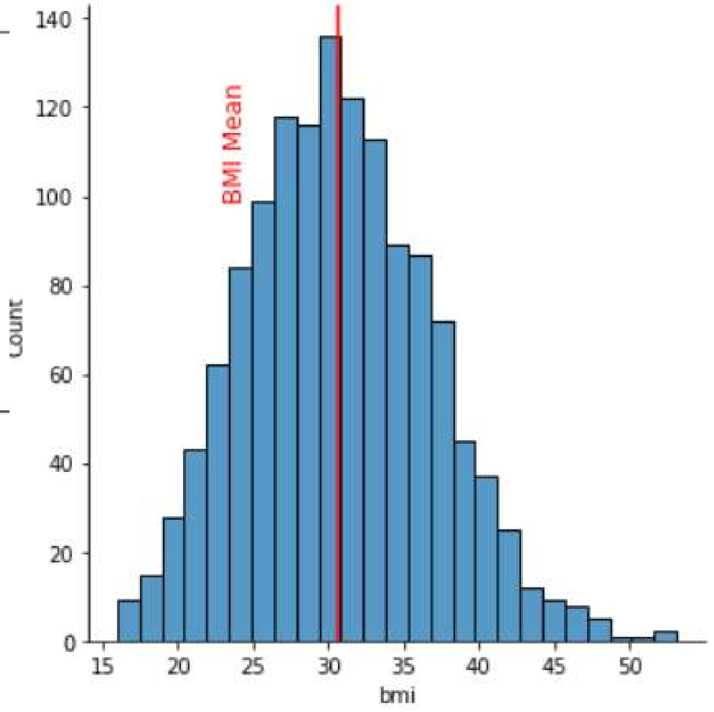
	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

DATA DISTRIBUTIONS

Distribution by charges



Distribution by bmi

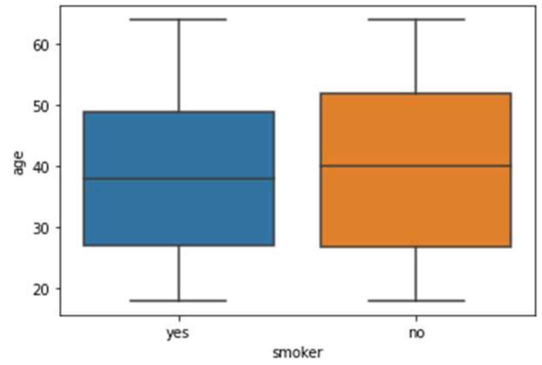


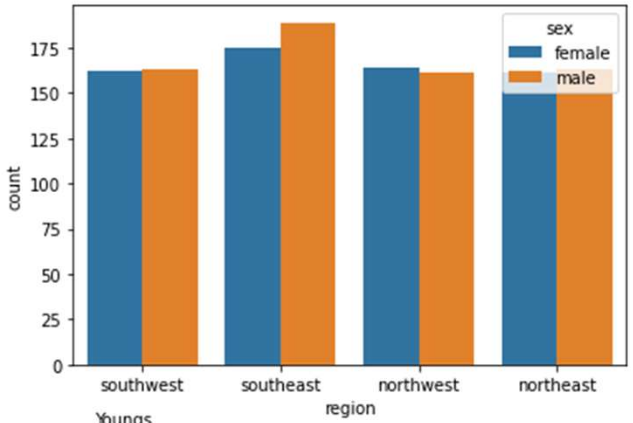
L1

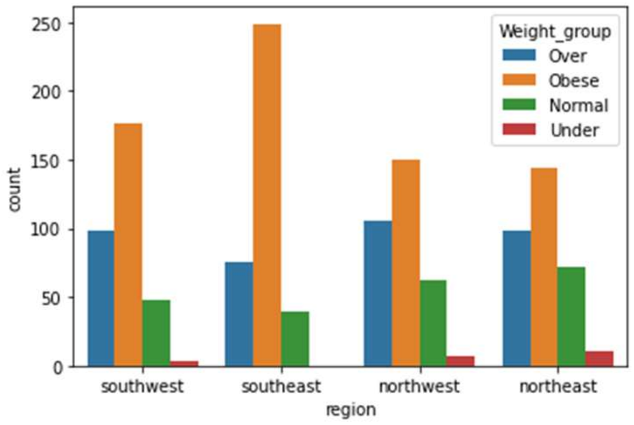
Slide 7

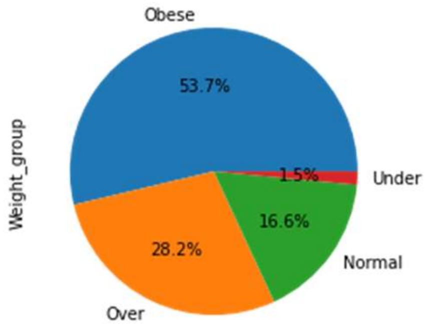
L1	Linda, 3/5/2021

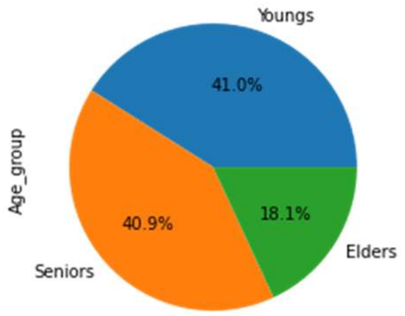
DATA DISTRIBUTIONS











Categorize the BMMI:

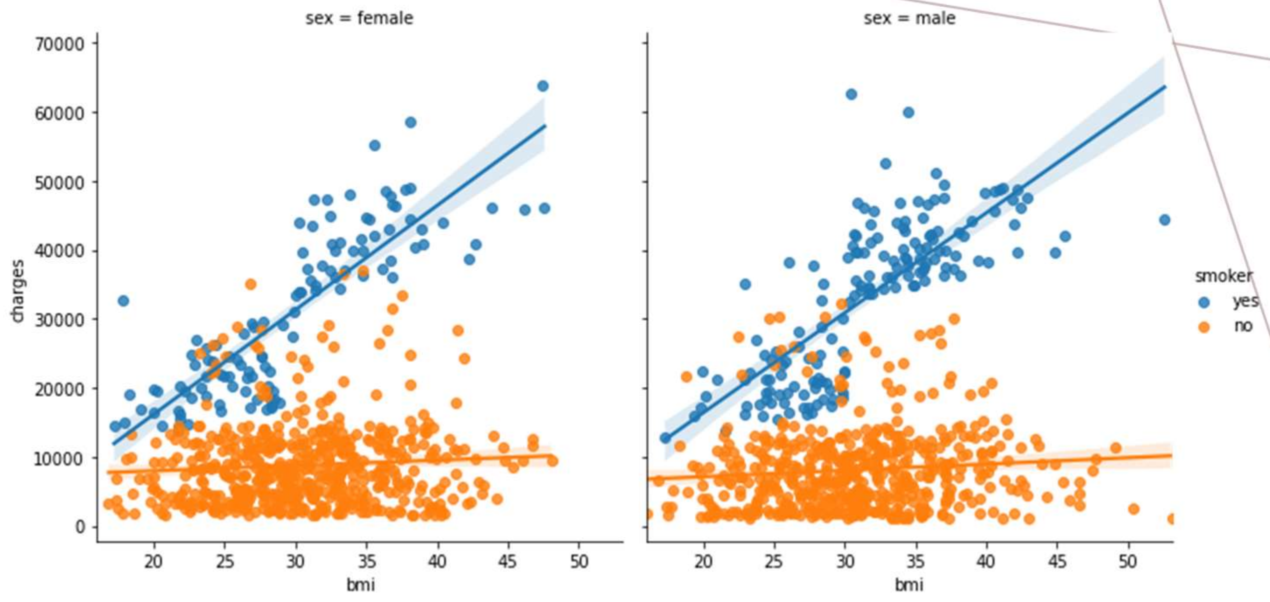
Under Weight: BMI < 18.5
Normal Weight: 18.5 =< BMI < 24.9
Overweight: 25 =< BMI < 29.9
Obese: BMI > 30

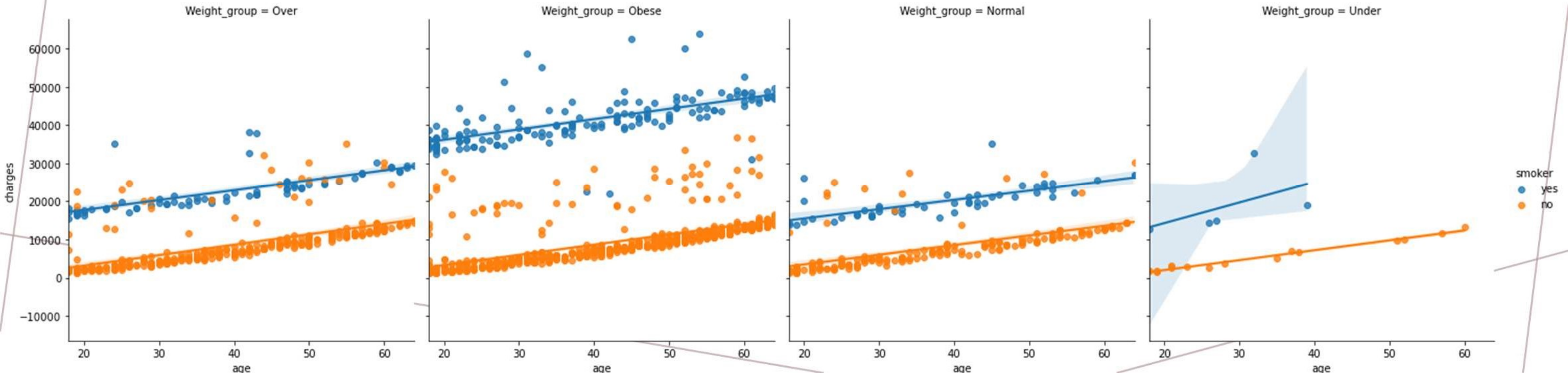
Categorize Age:

Age group: Teen : age <18 years
Youngs: 18<= age<35
Senior: 35 <= age <55
Elders : age >=55

DATA DISTRIBUTIONS

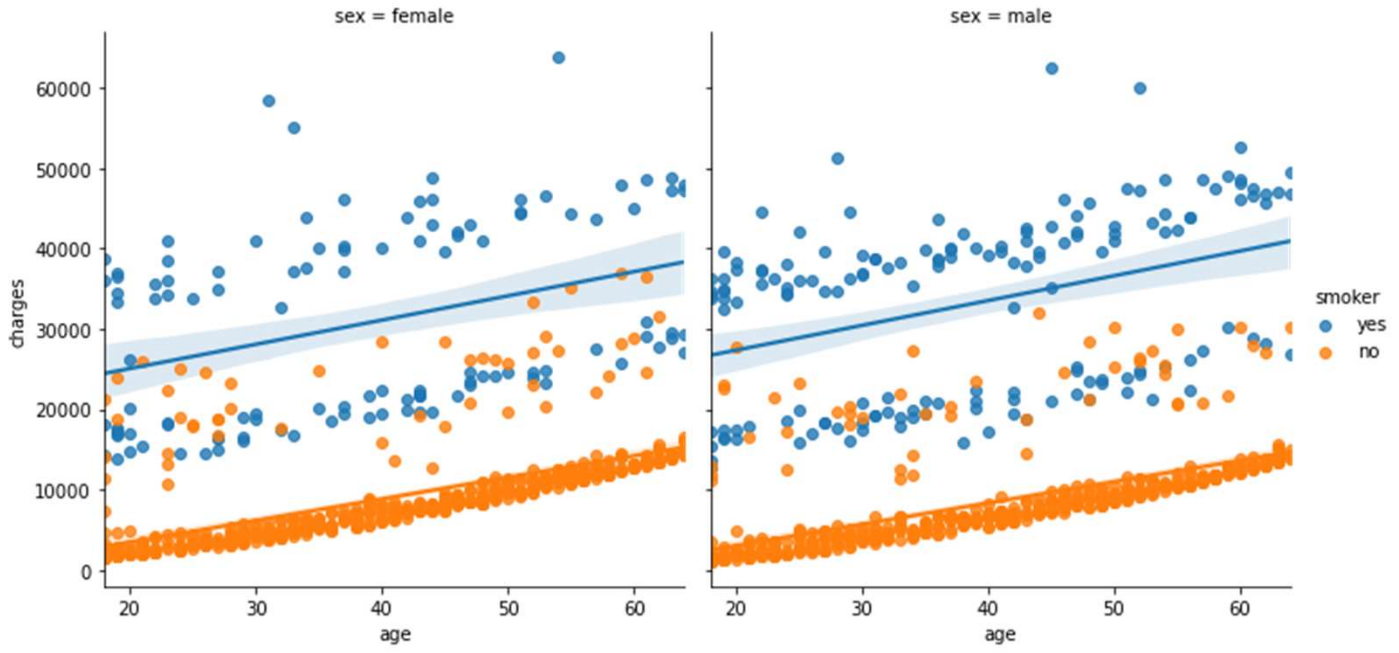
- People with higher bmi and smoker will have higher medical cost
- Weak positive correlation between bmi and charges of non smoker
- Obese woman and smoker has higher medical cost than man





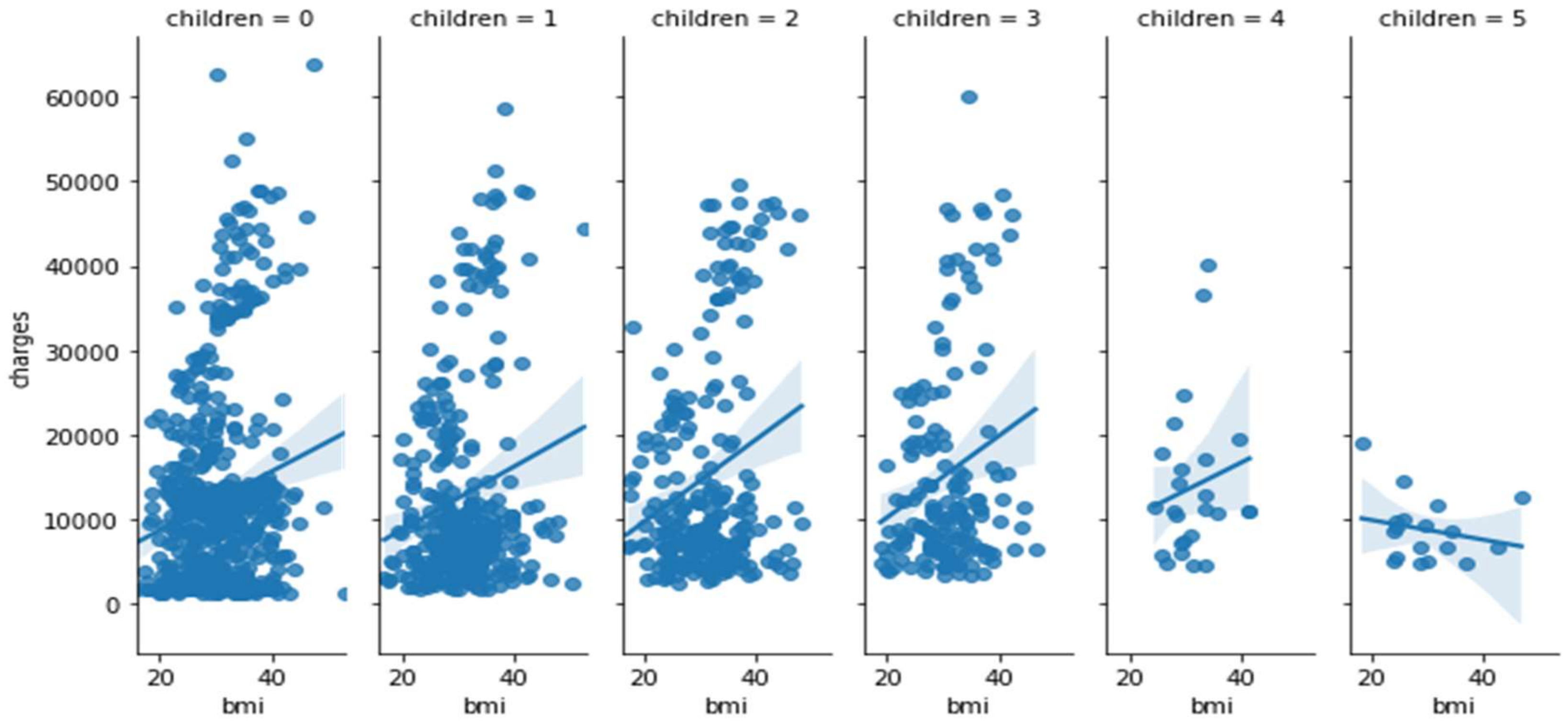
DATA DISTRIBUTIONS

- Smoker has double medical cost than non smoker
- Medical cost increases with age



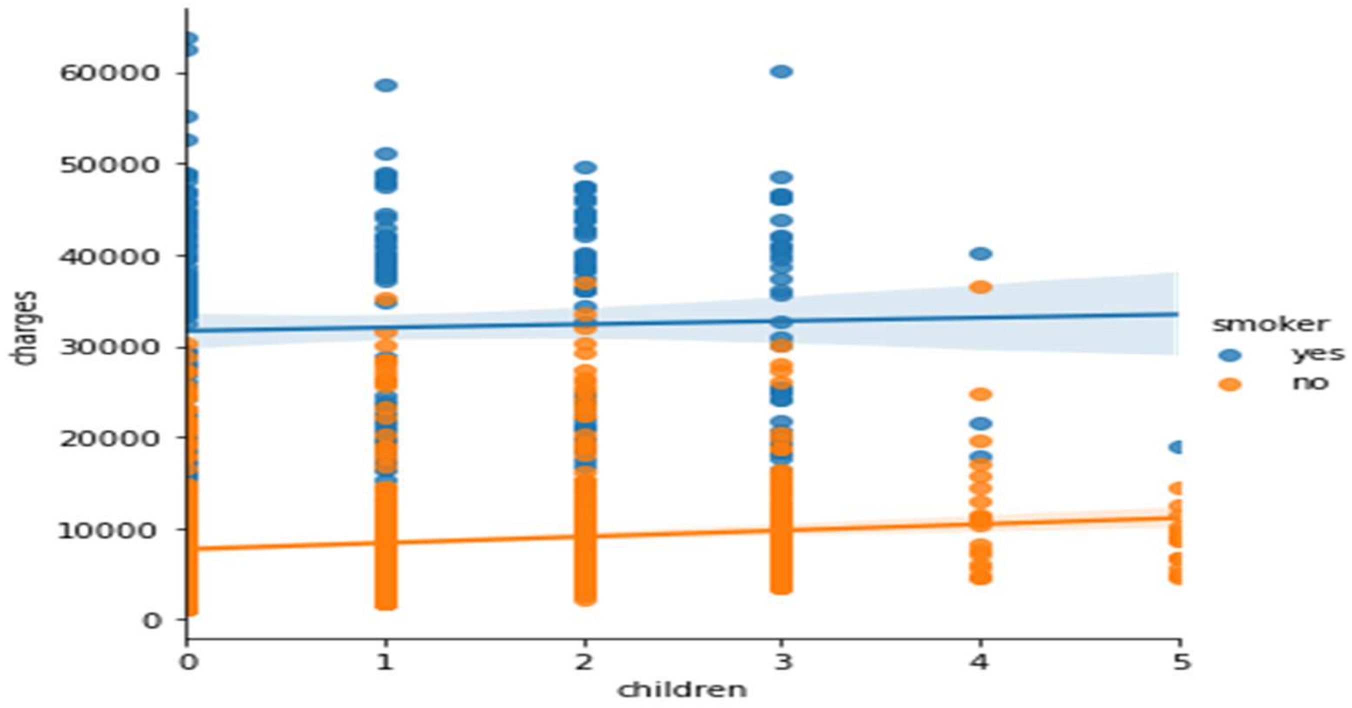
DATA DISTRIBUTIONS

- Family with more children is healthier and happier?

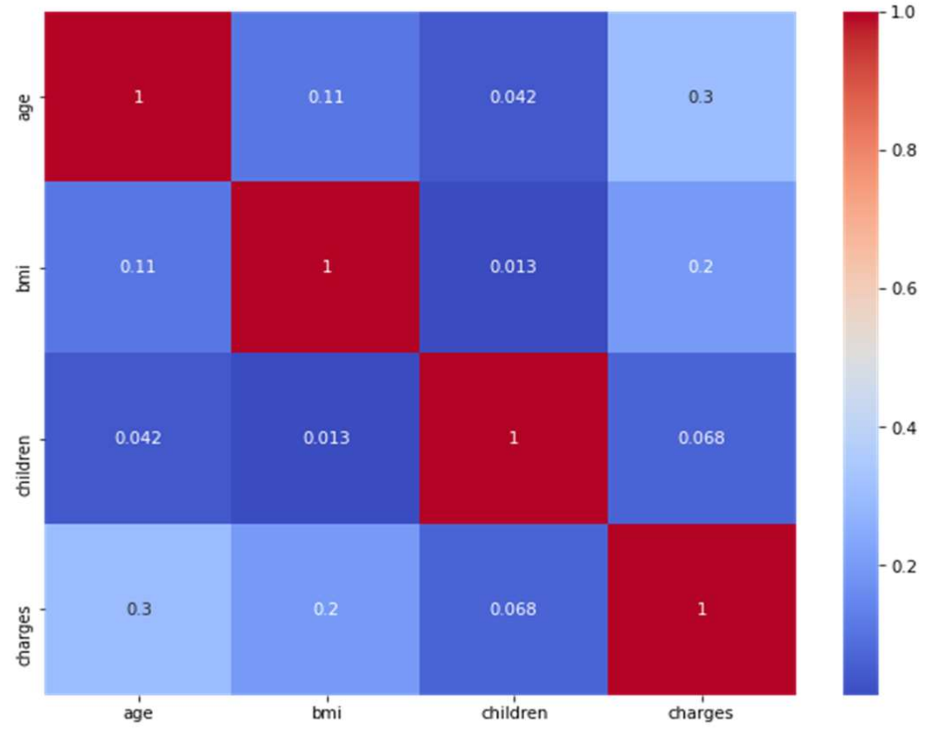


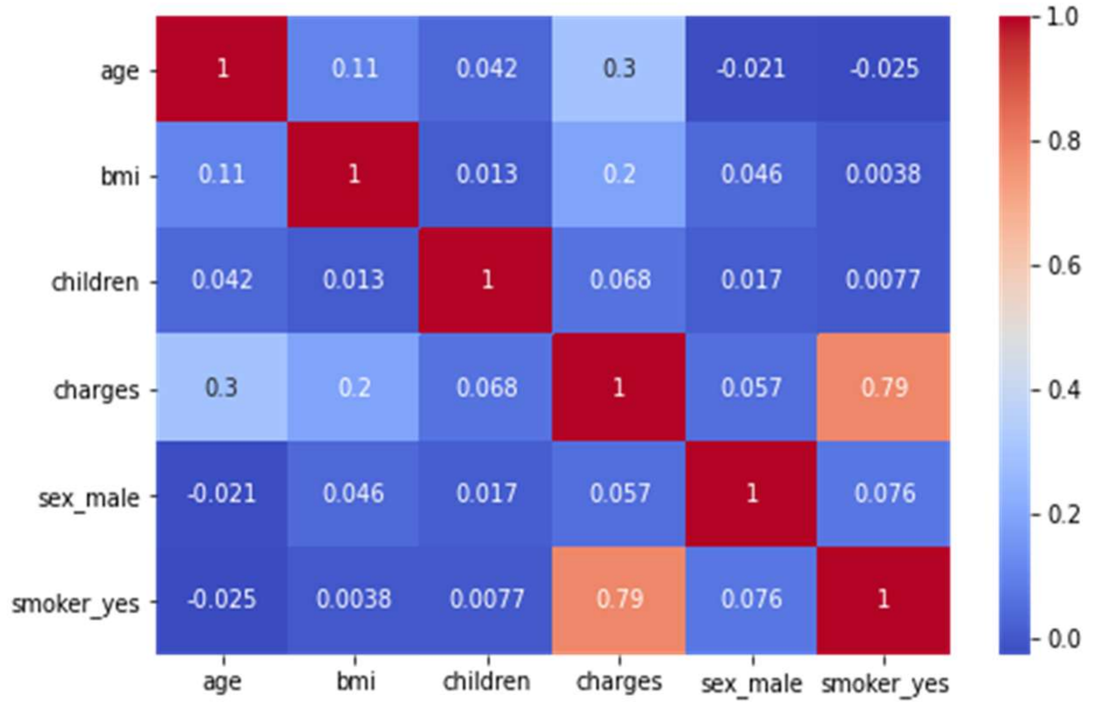
DATA DISTRIBUTIONS

- Family with more children is mostly non smoker

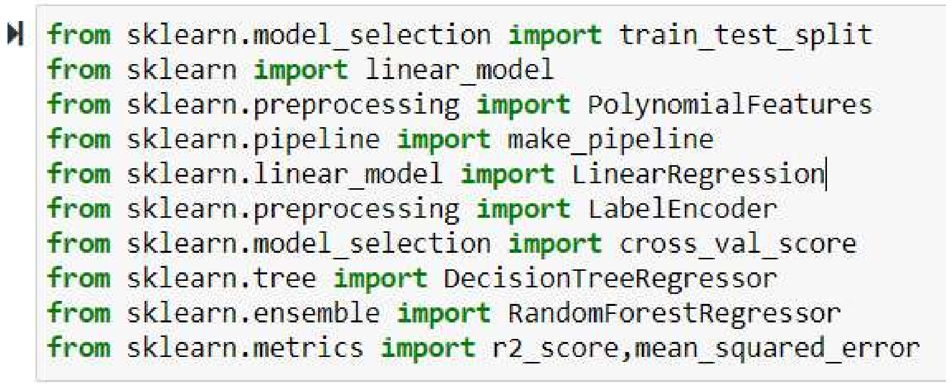


CORRELATIONS MATRIX

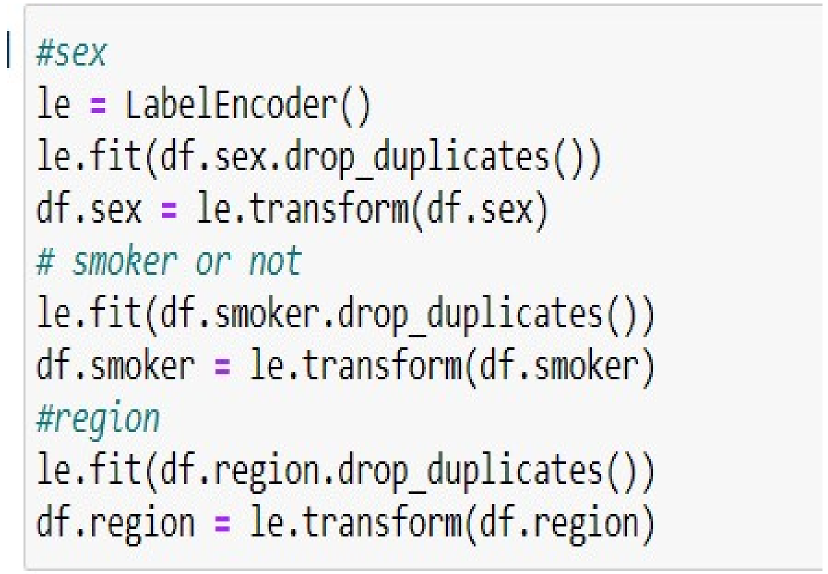




DATA PREPROCESSING

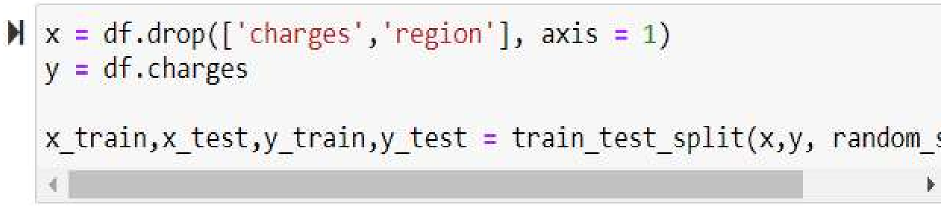


```
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score,mean_squared_error
```



```
#sex
le = LabelEncoder()
le.fit(df.sex.drop_duplicates())
df.sex = le.transform(df.sex)
# smoker or not
le.fit(df.smoker.drop_duplicates())
df.smoker = le.transform(df.smoker)
#region
le.fit(df.region.drop_duplicates())
df.region = le.transform(df.region)
```

Train Test split

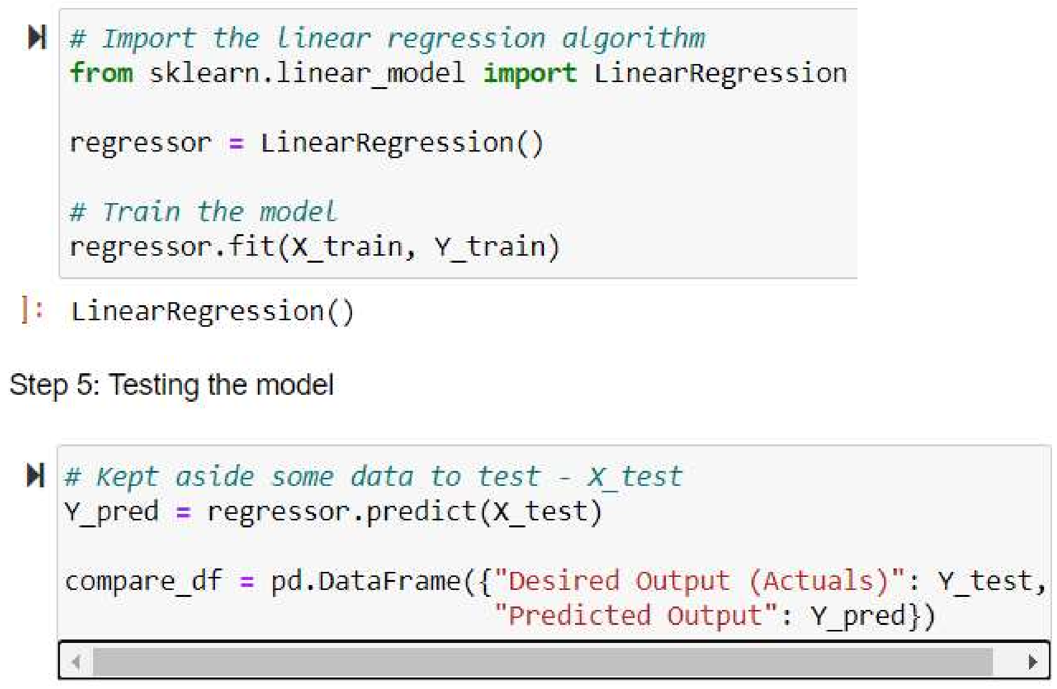


```
x = df.drop(['charges','region'], axis = 1)
y = df.charges

x_train,x_test,y_train,y_test = train_test_split(x,y, random_s
```

DATA MODELLING AND EVALUATION

- Multiple Linear Regression
- Polynomial Regression
- Decision Tree Regression
- Random Forest Regressor

TRAINING AND TESTING THE MODEL

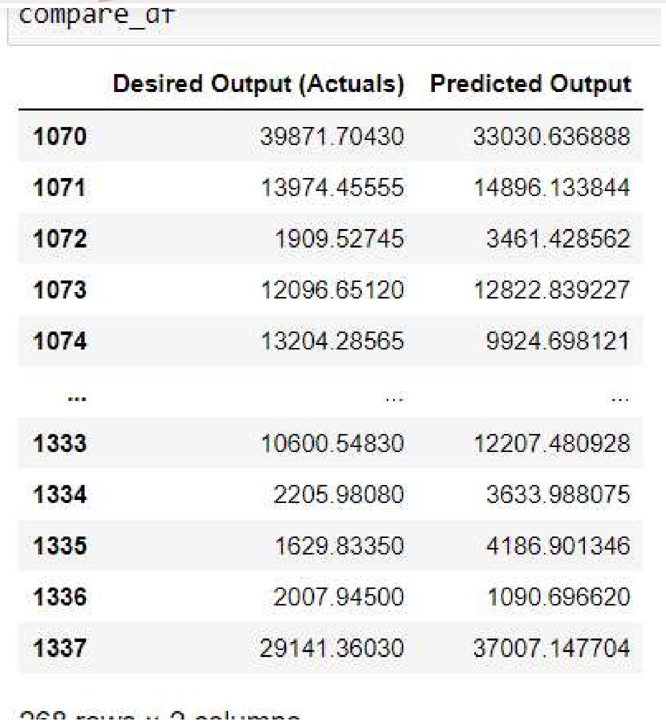


```
# Import the linear regression algorithm
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()

# Train the model
regressor.fit(X_train, Y_train)
```

```
]: LinearRegression()
```

Step 5: Testing the model

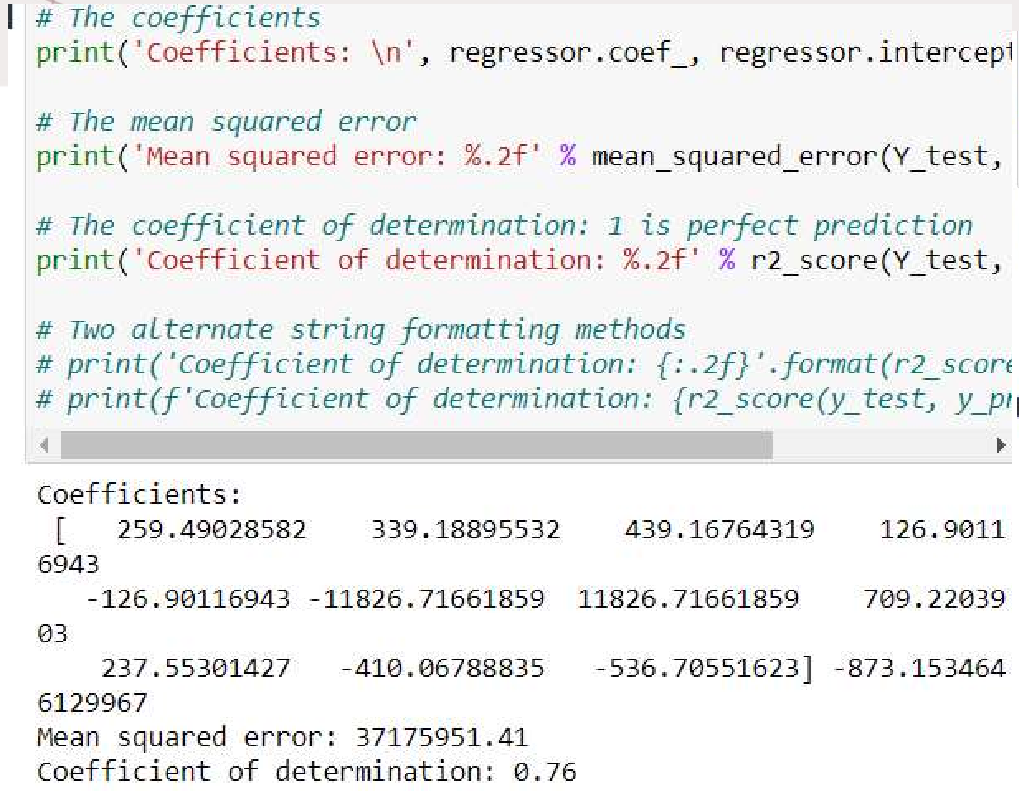
```
# Kept aside some data to test - X_test
Y_pred = regressor.predict(X_test)

compare_df = pd.DataFrame({"Desired Output (Actuals)": Y_test,
                          "Predicted Output": Y_pred})
```



```
compare_df
```

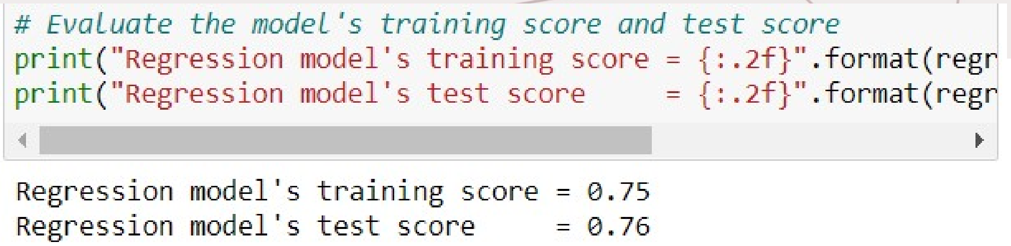
	Desired Output (Actuals)	Predicted Output
1070	39871.70430	33030.636888
1071	13974.45555	14896.133844
1072	1909.52745	3461.428562
1073	12096.65120	12822.839227
1074	13204.28565	9924.698121
...	...	...
1333	10600.54830	12207.480928
1334	2205.98080	3633.988075
1335	1629.83350	4186.901346
1336	2007.94500	1090.696620
1337	29141.36030	37007.147704

EVALUATE THE MODEL



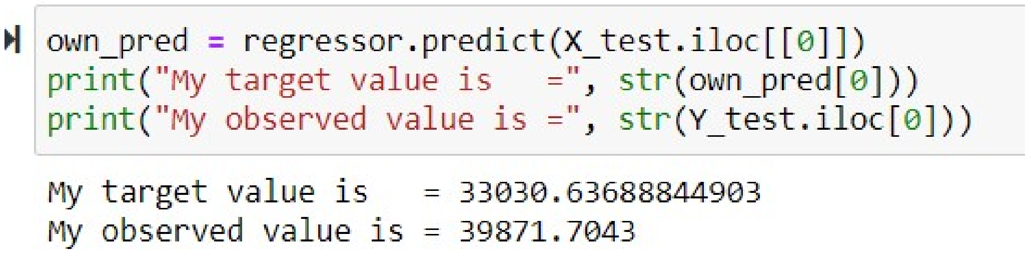
```
# The coefficients
print('Coefficients: \n', regressor.coef_, regressor.intercept

# The mean squared error
print('Mean squared error: %.2f' % mean_squared_error(Y_test,

# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: %.2f' % r2_score(Y_test,

# Two alternate string formatting methods
# print('Coefficient of determination: {:.2f}'.format(r2_score
# print(f'Coefficient of determination: {r2_score(y_test, y_pr
```

```
Coefficients:
 [   259.49028582    339.18895532    439.16764319    126.9011
6943
   -126.90116943 -11826.71661859  11826.71661859    709.22039
03
    237.55301427   -410.06788835   -536.70551623] -873.153464
6129967
Mean squared error: 37175951.41
Coefficient of determination: 0.76
```



```
# Evaluate the model's training score and test score
print("Regression model's training score = {:.2f}".format(regr
print("Regression model's test score     = {:.2f}".format(regr
```

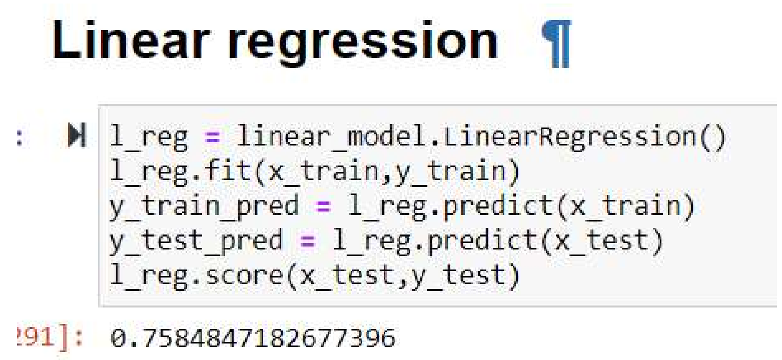
```
Regression model's training score = 0.75
Regression model's test score     = 0.76
```



```
own_pred = regressor.predict(X_test.iloc[[0]])
print("My target value is   =", str(own_pred[0]))
print("My observed value is =", str(Y_test.iloc[0]))
```

```
My target value is   = 33030.63688844903
My observed value is = 39871.7043
```

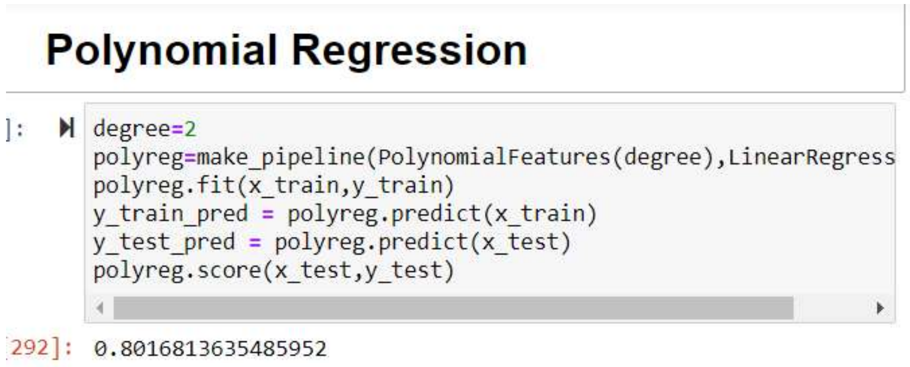
MODEL EVALUATIONS



Linear regression ¶

```
l_reg = linear_model.LinearRegression()
l_reg.fit(x_train,y_train)
y_train_pred = l_reg.predict(x_train)
y_test_pred = l_reg.predict(x_test)
l_reg.score(x_test,y_test)
```

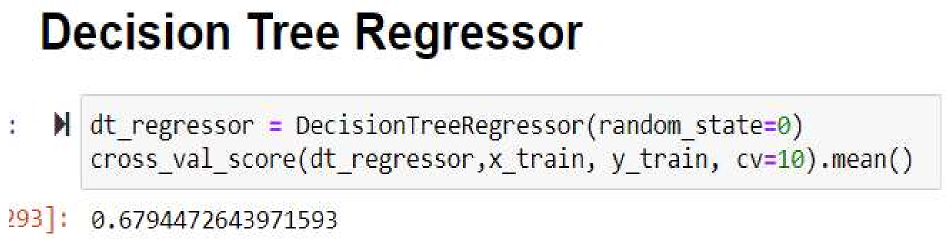
```
91]: 0.7584847182677396
```



Polynomial Regression

```
degree=2
polyreg=make_pipeline(PolynomialFeatures(degree),LinearRegress
polyreg.fit(x_train,y_train)
y_train_pred = polyreg.predict(x_train)
y_test_pred = polyreg.predict(x_test)
polyreg.score(x_test,y_test)
```

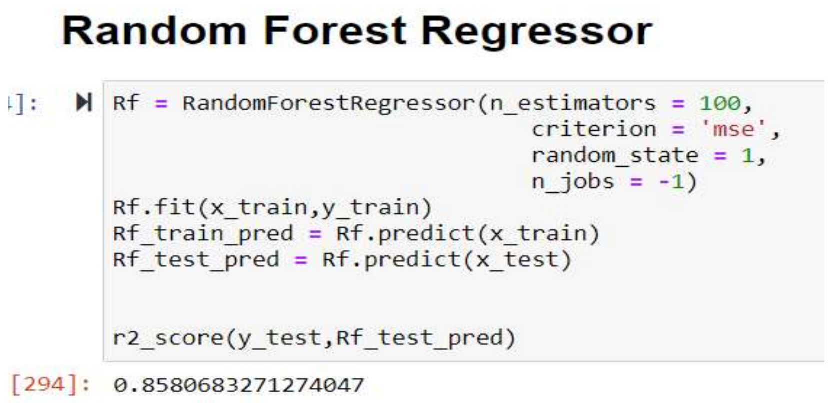
```
292]: 0.8016813635485952
```



Decision Tree Regressor

```
dt_regressor = DecisionTreeRegressor(random_state=0)
cross_val_score(dt_regressor,x_train, y_train, cv=10).mean()
```

```
93]: 0.6794472643971593
```



Random Forest Regressor

```
Rf = RandomForestRegressor(n_estimators = 100,
                           criterion = 'mse',
                           random_state = 1,
                           n_jobs = -1)
Rf.fit(x_train,y_train)
Rf_train_pred = Rf.predict(x_train)
Rf_test_pred = Rf.predict(x_test)


r2_score(y_test,Rf_test_pred)
```

```
[294]: 0.8580683271274047
```

FUTURE OPPORTUNITIES

- More data to use with another model evaluation
- Remove the outlier
- Other variables to analyse such as alcohol consumption, high blood pressure, work environment and stress

CONCLUSIONS

- Smoker is unhealthy and has higher risk to get serious disease
- Polynomial Regression and Random Forest Regressor has higher score prediction result

- Obesity has impact to have bad health and increase the medical cost
- Encourage to have healthy life styles by daily exercises, healthy diet, work life balance and happy life
- Higher medical insurance premium charges according to the age, healthy diet, and healthy life styles

Q & A
THANK YOU