Importing Libraries:

```
import pymongo
import pandas as pd
```

```
import pymongo
import pandas as pd

# Define the MongoDB connection details
mongo_uri = 'mongodb://localhost:27017'

# Create a MongoDB client and connect to the database
client = pymongo.MongoClient(mongo_uri)
db = client.ML  # "ML" is the database name

# Select the collection
collection = db.medical_insurance  # "medical_insurance" is the collection name

# Retrieve data from the collection
data = list(collection.find({}))

# Load the data into a Pandas DataFrame
df = pd.DataFrame(data)

# Close the MongoDB connection
client.close()

# Display the first few rows of the DataFrame
df.head()
```

	_id	age	sex	bmi	children	smoker	region	charges
0	6549dc9aa9df4772c57e5a88	19	female	27.900	0	yes	southwest	16884.92400
1	6549dc9aa9df4772c57e5a89	18	male	33.770	1	no	southeast	1725.55230
2	6549dc9aa9df4772c57e5a8a	28	male	33.000	3	no	southeast	4449.46200
3	6549dc9aa9df4772c57e5a8b	33	male	22.705	0	no	northwest	21984.47061
4	6549dc9aa9df4772c57e5a8c	32	male	28.880	0	no	northwest	3866.85520

```
client=pymongo.MongoClient('mongodb://localhost:27017')
client
```

```
     MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True)
```

```
import json
```

```
import pandas as pd
import numpy as np

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import mean_squared_error,mean_absolute_error, r2_score
```

```
import matplotlib.pyplot as plt
import seaborn as sns

import pickle
import json
```

```
# Loading Train dataset:
train_data = df.drop('_id',axis=1)
```

```
# Shape of dataset:
train_data.shape
```

```
(1338, 7)
```

```
# Cheacking for NaN Values (Missing Values):
train_data.isnull().sum()
```

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

```
# Insights of dataset:
train_data.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
```

```
# Description of dataset:
train_data.describe()
```

Double-click (or enter) to edit

	age	bmi	children	charges
mean	39.207025	30.663397	1.094918	13270.422265

```
encoder = LabelEncoder()
labels = encoder.fit_transform(train_data.sex)
```

```
train_data['sex'] = labels
```

50%	39.000000	30.400000	1.000000	9382.033000

```
train_data.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	yes	southwest	16884.92400
1	18	1	33.770	1	no	southeast	1725.55230
2	28	1	33.000	3	no	southeast	4449.46200
3	33	1	22.705	0	no	northwest	21984.47061
4	32	1	28.880	0	no	northwest	3866.85520

```
labels = encoder.fit_transform(train_data.region)
```

```
train_data['region'] = labels
train_data.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	yes	3	16884.92400
1	18	1	33.770	1	no	2	1725.55230
2	28	1	33.000	3	no	2	4449.46200
3	33	1	22.705	0	no	1	21984.47061
4	32	1	28.880	0	no	1	3866.85520

```
labels = encoder.fit_transform(train_data.smoker)
```

```
train_data['smoker'] = labels
train_data.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520

```
train_data.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   int32  
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   int32  
 5   region    1338 non-null   int32  
 6   charges   1338 non-null   float64
dtypes: float64(2), int32(3), int64(2)
memory usage: 57.6 KB
```

Train Test split

```
df = train_data.select_dtypes(exclude=object)
x = train_data.drop('charges',axis = 1)
y = train_data['charges']
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=34)
```

```
x_train
```

	age	sex	bmi	children	smoker	region
414	19	0	35.150	0	0	1
1279	25	0	26.790	2	0	1
647	40	0	23.370	3	0	0
764	45	0	25.175	2	0	0
1133	52	0	18.335	0	0	1
...	...	...	...	...	...	...
453	20	1	29.735	0	0	1
324	29	1	27.200	0	0	3
1109	45	1	20.350	3	0	2
490	19	0	32.900	0	0	3
1146	60	1	32.800	0	1	3

1070 rows × 6 columns

```
x_train.columns
```

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region'], dtype='object')
```

```
df
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520
...	...	...	...	...	...	...	...
1333	50	1	30.970	3	0	1	10600.54830
1334	18	0	31.920	0	0	0	2205.98080

Model Training

```
model = LinearRegression()
model.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Testing Data Evaluation
y_pred = model.predict(x_test)

mse = mean_squared_error(y_test, y_pred)
print("MSE :",mse)

rmse = np.sqrt(mse)
print("RMSE :",rmse)

mae = mean_absolute_error(y_test, y_pred)
print("MAE :",mae)

r2 = r2_score(y_test, y_pred)
print('R-Squared :',r2)
```

```
MSE : 41271154.57832547
RMSE : 6424.26295992976
MAE : 4410.013263731577
R-Squared : 0.7461578203319277
```

```
# Training Data Evaluation

y_pred_train = model.predict(x_train)
mse = mean_squared_error(y_train, y_pred_train)
print("MSE :",mse)

rmse = np.sqrt(mse)
print("RMSE :",rmse)

mae = mean_absolute_error(y_train, y_pred_train)
print("MAE :",mae)

r2 = r2_score(y_train, y_pred_train)
print('R-Squared :',r2)
```

```
     MSE : 35365859.39407278
     RMSE : 5946.920160391661
     MAE : 4094.433690405064
     R-Squared : 0.7514552383513079
```

```
filename = 'medical_insurance_cost_predictor.pkl'
pickle.dump(model, open(filename,'wb'))
```