Predictive Analytics: Regression

Steps in Modelling

After collecting the data from various sources and defining the problem statement. The following steps need to be followed to create a machine-learning model:

1. **Split the dataset into a feature matrix and target vector.**
2. **Split the dataset (feature matrix and target vector) into training and testing datasets.**
3. **Choose an appropriate model based on the problem statement.**
4. **Choose the hyperparameters for training the model.**
5. **Fit the model on the training dataset.**
6. **Apply the model to the testing dataset.**
7. **Evaluate the performance of the trained model.**

You will learn about all these steps in detail in the following videos.

Problem Statement

You are working with a consulting firm and your manager has given you the task to create a machine-learning model which can predict the salary of an employee given the years of experience.

1. Split the dataset into a feature matrix and target vector

Firstly, you will import the dataset.

```
In [4]:
1 dataset = pd.read_csv('Salary_Data.csv')
2 dataset.head()
```

Out[4]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

This dataset contains two columns namely "YearsExperience" and "Salary". For use case "YearsExperience" is the feature matrix and the "Salary" is the target vector.

```
In [8]:
1 X = dataset.iloc[:, :-1].values
2 y = dataset.iloc[:, -1].values
3
4 print(X[:5])
5 print(y[:5])
```

```
[[1.1]
 [1.3]
 [1.5]
 [2. ]
 [2.2]]
[39343. 46205. 37731. 43525. 39891.]
```

2. Splitting the dataset into the Training set and Test set

After splitting the dataset into a feature matrix and target vector, the next step is to split these into training and testing datasets.

```
In [9]:
1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)
```

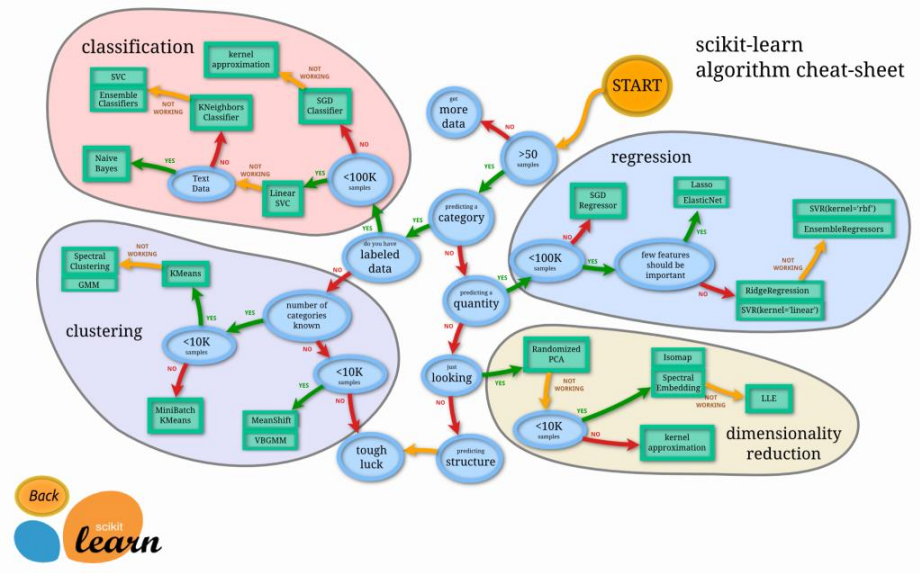
```
In [10]:
1 X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
Out[10]: ((20, 1), (10, 1), (20,), (10,))
```

The dataset is now split into the training and the testing set using the train_test_split library of sklearn.

3. Choose an appropriate model based on the problem statement

According to the problem statement, the model needs to predict the salary of an employee based on years of experience. This is continuous output which falls under the category of regression. Also, the model only contains one feature which can be trained using the “Simple Linear Regression” model.



The sklearn website also contains a well-structured map for choosing the right estimator. You can click on this link to browse through the map which is provided as an image above.

4. Choose the hyperparameters for training the model

You can create a Linear Regression model using the sklearn library and can also change the hyperparameters for optimizing the model performance.

In [19]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression(fit_intercept=True)
```

You are going to learn a lot about hyperparameters and hyperparameter tuning in the following videos.

5. Fit the model on the training dataset

After initializing the model you need to fit the model using the training dataset.

```
In [20]:  1  regressor.fit(X_train, y_train)
Out[20]: LinearRegression()
```

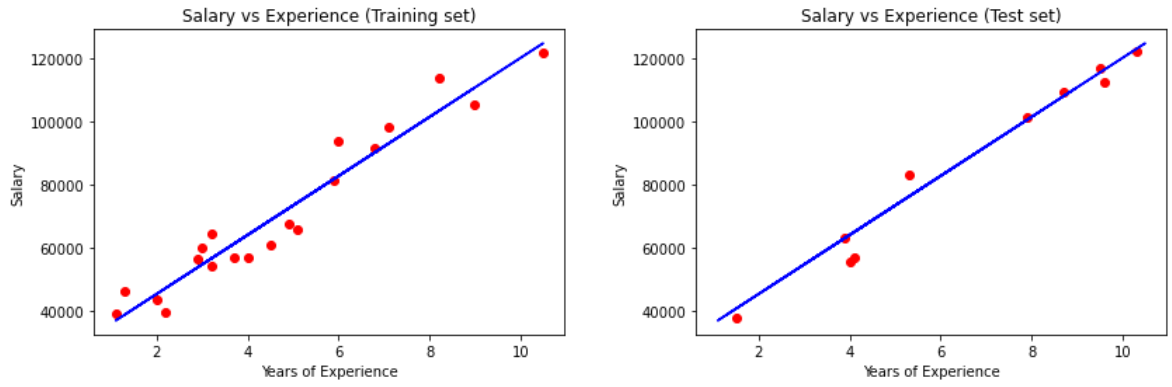
6. Apply the model to the testing dataset

After training the model on the training dataset you can predict the output of the test dataset.

```
In [16]:  1  y_pred = regressor.predict(X_test)
```

7. Evaluate the performance of the trained model

After predicting the output you can visualize the output and then evaluate the output on various parameters. Let's visualize the output of the model on the training and testing dataset.



You will learn various metrics that can be helpful in evaluating models in future videos.