Linear Regression

```
1 - It is a Supervised Learning predictive model used for finding the linear
    relationship between dependent variable(Y) and independent variable(s)(X)
2 - Linear Regression is used when output variable has continues values.
3 - Equaion of linear regression : Y = mx + c

Two Types:
 1. Simple Linear Regression(Only one independent Variable)
 2. Multiple Linear Regreesion (Two or more independent Variables)
```

Required Liabraries for Linear Regression

In [ ]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_scores
```

Assumption of Linear Regression

1 - Linearity

```
1-Indipendent variable(X) and Dependent variable(Y) should be linear to each other.
2-i.e. relationship between x and y is linear.
```

2 - Independence

```
All observations(Indipendent variable) are indipendent to each other.
```

3 - No Multicolinearity

```
There is no strong co-relation between indipendent variables.
```

4 - Normality

```
1 - Residuals should have normally distribution.
2 - Normal distribution also known as Gaussian Distribution.
3 - Gaussian distribution is a probablity Distribution that is
    symmetric about the mean and it shouuld be in bell shape curve.
```

5 - Homoscedasticity

```
1 - Homo means same & scedasticity means variance , so it means same variance.
2 - If the variance in the residual error is constant regardless of the dependent
    variable(x) then it is homoscedasticity.
```

Project Steps

In [ ]:

```
1. Problem Statement
2. Data Gathering(JSON, CSV, Excel, PDF, Images, Videos, Text, etc)
3. Exploratory Data Analysis(Pandas, Matplotlib, Seaborn) (60 %)
4. Feature Engineering (Scaling, Handling Outliers, Encoding, Log Transform, Binning)
5. Feature Selection (Required Feature to train the model)
6. Model Building (LR, Logistic Regression, DT, RF, etc)
7. Model Evaluation (MSE, AMSE, Classification Report, Confusion matrix, etc)
8. Model Deployment (AWS, GCP, Azure, Heroku)
```

Coefficient of Correlation(R)

```
1 -It is also called as Karl Pearson Correlation Coefficient/ R-value.
2 -It gives the correlation between IV(X1) and IV(X2).
3 -It gives correlation between IV(X) and DV(Y).
4 -It gives the correlation between all the variables present in the Dataset.
5 -Range of R-value is -1 to +1

R > 0.7 or R < -0.7 ---> Good predictors or strongly correlated.
R = -0.3 to 0.3     ---> Bad predictors.
R = 0               ---> No relation

Rxy = Covariance / Std dev of(x,y)
```

Covariance

```
It is measure of association between X and Y.
1 - If Y increase with increasing X then it is +ve covariance.
2 - If Y decrease with incresing X then it is -ve covariance.
3 - If there is no linear tendency for Y with change to X then it is zero(0)
```

Gradient Descent Algorithm (GD)

In [ ]:

```
1. Use to find best fit line.
2. Best value of m and c.
3. To find m and c, GD uses partial derivatives which is use reduce cost function(MSE).
4. Cost function = Mean squared Error(MSE)
5. It uses the X_train and Y_train.
```

Best Fit Line(BFL)

In [ ]:

```
1. It passes to maximum number of datapoints
2. Line which has lowest Sum of Errors (MSE)
3. Gradient Descent Algorithm is used to find BFL
3. Algorithm finds one BFL from infinite number of posssibilities.
4. BFL always passes from Xmean and Ymean.

```

Mean Square Error(MSE)

In [ ]:

```
1. It is define as the average of square of difference between Yactual and Ypredicted.
2. MSE = Σ(Yact - Ypred)/N
```

Cost Function (j)

In [ ]:

```
1. Cost function for 'm'
Dj/Dm = -2/N Σ(Yact - Ypred) * Xact

2. Cost function for 'c'
Dj/Dc = -2/N Σ(Yact - Ypred)
```

Model Evaluation

Prediction

```
y_pred = linear_model.predict(x_test) >> we get predicted values of dependant
         variable Y
```

1. SSE(sum of square error)

```
1. Called as Residual or Schochastic Error
2. Residual Error = (y-actual - y-pred)
3. Sum of Square Error :
   SSE = Σ(y-actual - y-pred)^2
```

2. SSR(Sum of Square Due to Regression)

```
1. Called as Regression or Deterministic Error
2. Regression Error = (y-pred  - ymean)
3. Sum of Square Due to Regression :
   SSR = Σ(y-pred  - ymean)^2
```

3. SST(Sum of Squares Total)

```
1. Square difference between Dependent variable()Y and its mean.
2. SST = Σ(y-actual - y-mean)^2
3. SST = SSE + SSR
```

4. mean_squared_error

```
mse = mean_squared_error(y_test,y_pred)
print('Mean Squared Error is :',mse)
```

5. Root Mean Squared Error

```
rmse = np.sqrt(mse)
print('Root Mean Squared Error is :',rmse)
```

6. Mean Absolute Error

```
mae = mean_absolute_error(y_test,y_pred)
print('Mean absolute Error is :',mae)
```

7. r2_score (Coefficient of determination)

In [ ]:

```
1. It is to check goodness of best fit line
2. For Good Correlated features R2 will increase more
3. For Bad Predictors r2 will very low (0)
4. R2 can be negative
5. R2 = 1 - (SSE/SST)
```

Disadvantage of r2_score

In [ ]:

```
1. R2 will never decrease
2. When we add more features R2 score will increase
   (For Correlated and non correlated features)
3. ex. if R value is low for x3 then again r2 will increase

for Testing Data  : r2_score(y_test,y_pred)
for Training Data : r2_score(y_train,y_pred_train)
```

8. Adjusted r2_Score

In [ ]:

```
1. When we add more features R2 score will increase ,this is disadvantage of
   R2_score so we use Adjusted R2 score.
2. There is no built in function for adjusted R2 score, we need to calculte it.

Adjusted r2_score = R2 - [(k-1/n-k)*(1-R2)^2]
k - no. of parameters
n - total no. od size
```

Advantages of Linear Regression

In [ ]:

```
1. Perform exceptionally well for linearly seperated data
2. Easy to implement
3. It can handle overfitting
```

Disdvantages of Linear Regression

In [ ]:

```
1. Model Fails,If the relation between independent variables and dependent variab
2. If the independent variables are correlated, then it may affect performance
3. Impact of missing Values / Sensitive to missing Values
4. Impact of Outliers / Sensitive to outliers
```

Applications

In [ ]:

```
1. Price
2. Population
3. Age
4. Any Contineous Data
```