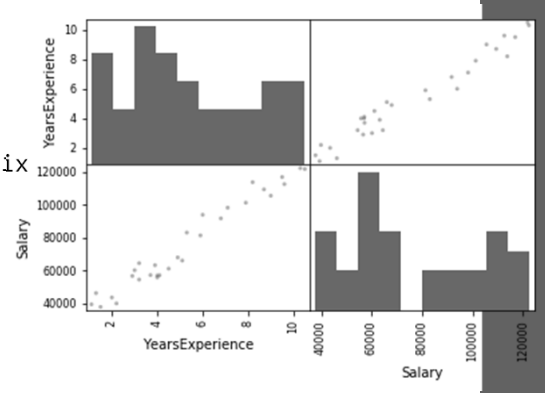
REGRESSION

Mohan M J

Simple Linear Regression

```
import pandas as pd
import numpy as np
myData = pd.read_csv('Salary_Data.csv')
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as myplot
scatter_matrix(myData)
myplot.show()
myplot.scatter(X,Y)
myplot.show()
np.corrcoef(myData.YearsExperience, myData.Salary)
```




```
[ 1. , 0.97824162],
[ 0.97824162, 1. ]
```

Contd..

```
X = myData.iloc[:,:-1].values

Y = myData.iloc[:,1].values

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size= 0.3)

from sklearn.linear_model import LinearRegression

myModel = LinearRegression()

myModel.fit(X_train,Y_train)

y_pred = myModel.predict(X_test)
```

Contd..

```
myplot.scatter(X_train,Y_train, color='r')

myplot.plot(X_train,myModel.predict(X_train), color='b')

myplot.title('Salary vs Exp')

myplot.xlabel('Years of exp')

myplot.ylabel('Salary')

myplot.show()
```

Contd..

```
myplot.scatter(X_test,Y_test, color='r')
myplot.plot(X_train,myModel.predict(X_train), color='b')
myplot.title('Salary vs Exp')
myplot.xlabel('Years of exp')
myplot.ylabel('Salary')
myplot.show()
```

Multiple Linear Regression

? Profit (INR) for a few startups in the current financial year is given in the dataset (Sartups.csv)

Operational spends for the company such as R&D Spends, Administration Expenses, Marketing Spends are given

City where the company is established is also given

Find out the company performance w.r.t. the operational expenses and location. Asses the companies

Build a model for maximizing profit for the Venture Capitalist

Contd..

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as myPlot
import pandas as pd
# Importing the dataset
myData = pd.read_csv('Startup_India.csv')
myData
X = myData.iloc[:, :-1].values
X
y = myData.iloc[:, 4].values
y
```

Contd..

```
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
X[:,3] = labelencoder.fit_transform(X[:,3])
X
oneHotEncoder = OneHotEncoder(categorical_features = [3])
X = oneHotEncoder.fit_transform(X).toarray()
print(X)
#Avoiding Dummy variable trap
X=X[:,1:]
X
```

Contd..

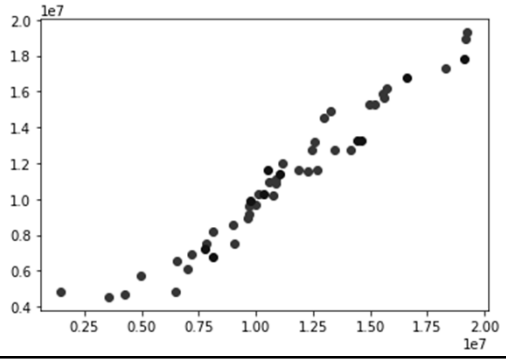
```
# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 0)


# Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)


# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

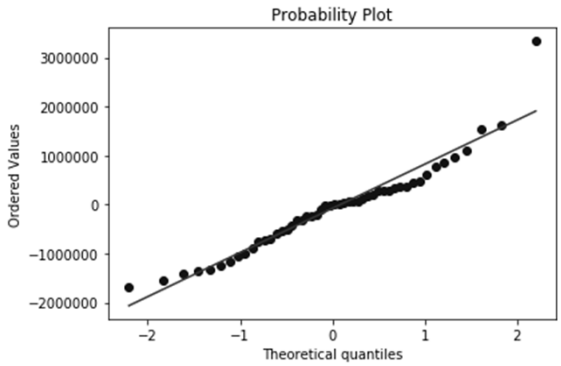
Contd..

```
#plt.figure(1, figsize=(10, 10),)
plt.scatter(y_train,regressor.predict(X_train), color='r')
plt.scatter(y_test,regressor.predict(X_test), color='b')
plt.show()
```



Contd..

```
from scipy import stats as mystats
res = regressor.predict(X)-y
mystats.probplot(res,plot=plt)
plt.show()
```



Contd..

```
# Building the optimal model using Backward Elimination
import statsmodels.formula.api as sm
X = np.append(arr = np.ones((50, 1)).astype(int), values = X, axis = 1)
X_opt = X[:, [0, 1, 2, 3, 4, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()

X_opt = X[:, [0, 1, 3, 4, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()
```

Contd..

```
X_opt = X[:, [0, 3, 4, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()

X_opt = X[:, [0, 3, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()

X_opt = X[:, [0, 3]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()
```

THANKS