Cofficient Logistic Regression

```
import numpy

from sklearn import linear_model

import matplotlib.pyplot as plt

X = numpy.array([3.78, 2.44, 2.09, 0.14, 1.72, 1.65, 4.92, 4.37, 4.96, 4.52, 3.69, 5.88]).reshape(-1,1)

y = numpy.array([0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1])

plt.plot(X,y)

logr = linear_model.LogisticRegression()

logr.fit(X,y)

log_odds = logr.coef_

odds = numpy.exp(log_odds)

print(odds)
```

Probability Logistic Regression

```
import numpy

from sklearn import linear_model

import matplotlib.pyplot as plt

X = numpy.array([3.78, 2.44, 2.09, 0.14, 1.72, 1.65, 4.92, 4.37, 4.96, 4.52, 3.69, 5.88]).reshape(-1,1)

y = numpy.array([0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1])

plt.plot(X,y)

logr = linear_model.LogisticRegression()

logr.fit(X,y)

def logit2prob(logr, X):

  log_odds = logr.coef_ * X + logr.intercept_

  odds = numpy.exp(log_odds)

  probability = odds / (1 + odds)

return(probability)

print(logit2prob(logr, X))
```

Decision Tree

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
data = {
   'Age' : [36,42,23,52,43,44,66,35,52,35,24,18,45],
   'Experience' : [10,12,4,4,21,14,3,14,13,5,3,3,9],
   'Rank' : [9,4,6,4,8,5,7,9,7,9,5,7,9],
   'Nationality' :['UK','USA','N','USA','USA','UK','N','UK','N','N','USA','UK','UK'],
   'Go' : ['NO','NO','NO','NO','YES','NO','YES','YES','YES','YES','NO','YES','YES']
}
df = pd.DataFrame(data)
print(df)
label_encoder = LabelEncoder()
df['Nationality'] = label_encoder.fit_transform(df['Nationality'])
df['Go'] =label_encoder.fit_transform(df['Go'])
x = df[['Age','Experience','Rank','Nationality']]
y = df['Go']
model = DecisionTreeClassifier()
model.fit(x,y)
input_data = pd.DataFrame([[40, 10, 7, 1]], columns=['Age', 'Experience', 'Rank', 'Nationality'])
prediction = model.predict(input_data)
prediction_label = label_encoder.inverse_transform(prediction)
print("Should the 40 years old american comedian go to the show?"+prediction_label[0])
```

Random Forest

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

import warnings

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.tree import plot_tree

df = pd.read_csv('/content/Random forest1.csv')

print(df)

label_encoder = LabelEncoder()

df['Position'] = label_encoder.fit_transform(df['Position'])

x = df[['Position','Level']]

y = df['Salary']

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=42)

rf_regressor = RandomForestRegressor(n_estimators=100,random_state=42)

rf_regressor.fit(x_train,y_train)

y_pred = rf_regressor.predict(x_test)

print("Prediction:",y_pred)

from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y_test,y_pred)

r2 = r2_score(y_test,y_pred)
```

```
print(f"Mean_squared_Error:{mse}")

print(f"R_squared:{r2}")

tree = rf_regressor.estimators_[0]

plt.figure(figsize=(20,10))

plot_tree(tree,filled=True,feature_names=['Position','Level'],rounded=True, precision=2)

plt.title("Visuallization of single Decision Tree from Random forest Model")

plt.show()
```