Capstone project

Dataset I selected in breast cancer dataset. Because 1 of 8 women is diagnosed with breast cancer in their lifetime. So I wished to create an AI model to predict that.

Domain : Machine Learning

Learning : Supervised Learning

Regressor or Classification : Classification

```
In [1]:
import pandas as pd

df = pd.read_csv("breast-cancer.csv")
```

```
In [2]:
df.shape

Out[2]: (569, 32)
```

Dataset has 569 rows and 32 columns. Id column is not required for us so I removed it from the Dataset.

```
df.drop(columns=["id"],inplace=True)
```

So my column count now becomes 31.

Our output column is 'diagnosis_M' and all other columns are our input columns.

```
In [8]:
df['diagnosis_M'].value_counts()

Out[8]: diagnosis_M
0    357
1    212
Name: count, dtype: int64
```

```
In [15]:
indep = df[['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
       'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst']]
dep = df['diagnosis_M']
```

- After this I applied univariate analysis to this dataset for identifying any outliers is present or not.
- Since our input columns are 30, it is little difficult for the user to provide all the values for the input variables. So we applied **feature selection** and **dimentionality reduction** to get the lesser variable count required to create a model to predict breast cancer.

```
Using estimator: LogisticRegression()

Selected Features (indices or names): Index(['radius_mean', 'texture_se', 'radius_worst', 'concavity_worst', 'concave points_w
orst'], dtype='object')
```

	Logistic	SVMl	SVMnl	KNN	Navie	Decision	Random
Logistic	0.958042	0.951049	0.944056	0.937063	0.93007	0.944056	0.951049
SVC	0.909091	0.916084	0.923077	0.902098	0.86014	0.86014	0.958042
Random	0.951049	0.958042	0.965035	0.937063	0.951049	0.937063	0.958042
DecisionTree	0.972028	0.972028	0.958042	0.979021	0.944056	0.923077	0.965035

We are getting LogisticRegression model with **95.8% with 5 input columns**.

So now our input and output variable becomes,

```
In [19]:
indep = df[['radius_mean', 'texture_se', 'radius_worst', 'concavity_worst', 'concave points_worst']]
dep = df['diagnosis_M']
```

We are splitting our df into train and test set and the applied Standard Scaler to it and we created Logistic Regressor model and we got 93.7% **f1_score** and 99.0% **roc_auc score**.

In [24]:
```
from sklearn.metrics import f1_score

f1_macro=f1_score(y_test,grid_pred,average='weighted')
f1_macro
```

Out[24]: 0.9370387881309454

In [25]:
```
from sklearn.metrics import roc_auc_score

roc_auc_score(y_test,grid.predict_proba(x_test)[:,1])
```

Out[25]: 0.9903567984570877

- Since we applied Standard Scaler preprocessing to our input variables, we saved our final model and standard scaler also to load these two in our deployment phase.
- We loaded the saved model and saved preprocessing technique in our deployment phase, got the inputs and applied preprocessing to the input and passed those to the model for prediction of Breast cancer Malignant tumor or not.

In [1]:
```
import pickle

loaded_model = pickle.load(open("finalized_model_LogisticRegression.sav", 'rb'))
scaler = pickle.load(open("scaler.pkl", 'rb'))

input_data = [[20.3, 1.2, 28.5, 0.45, 0.23]]

scaled_input = scaler.transform(input_data)

result = loaded_model.predict(scaled_input)
print(result)
```

```
[1]
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

In [2]:
```
input_data = [[12.1, 0.4, 13.3, 0.02, 0.01]]

scaled_input = scaler.transform(input_data)

result1 = loaded_model.predict(scaled_input)
print(result1)
```

```
[0]
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```