SQE LAB PROJECT

Submitted by:
AQSA TABASSUM (02)
SHAMSA KANWAL (028)
SADIA AZAM (039)

Project Title: Breast Cancer Class

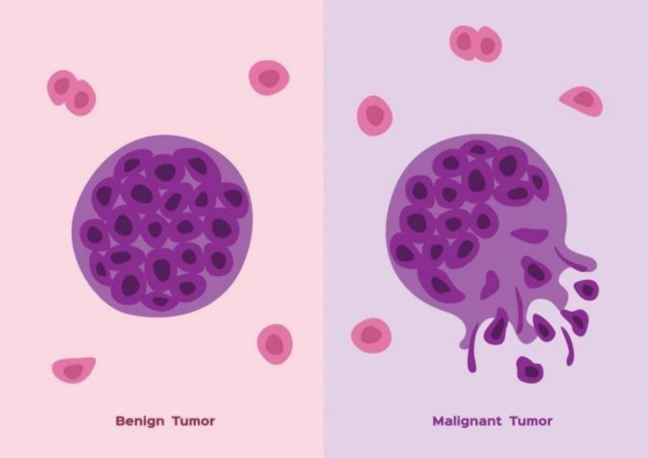
Table of Content Page No

1. TUMOR:

Human body is made up of cells, tissues and organs etc. All the cells in our body divides and grows and sometimes what happens is some of cells in our body may divides repeatedly without some control. In that case it forms an abnormal mask and abnormal tissues those abnormal tissues are referred as tumor.

2. TYPES OF TUMOR:

1. Benign
2. Malignant



BENIGN TUMORS:

- ✞ Benign tumors are those which do not move to the other parts of body
- ✞ They are not as much harmful called as non-cancerous tumors ✞ Slow growing

MALIGNANT TUMORS:

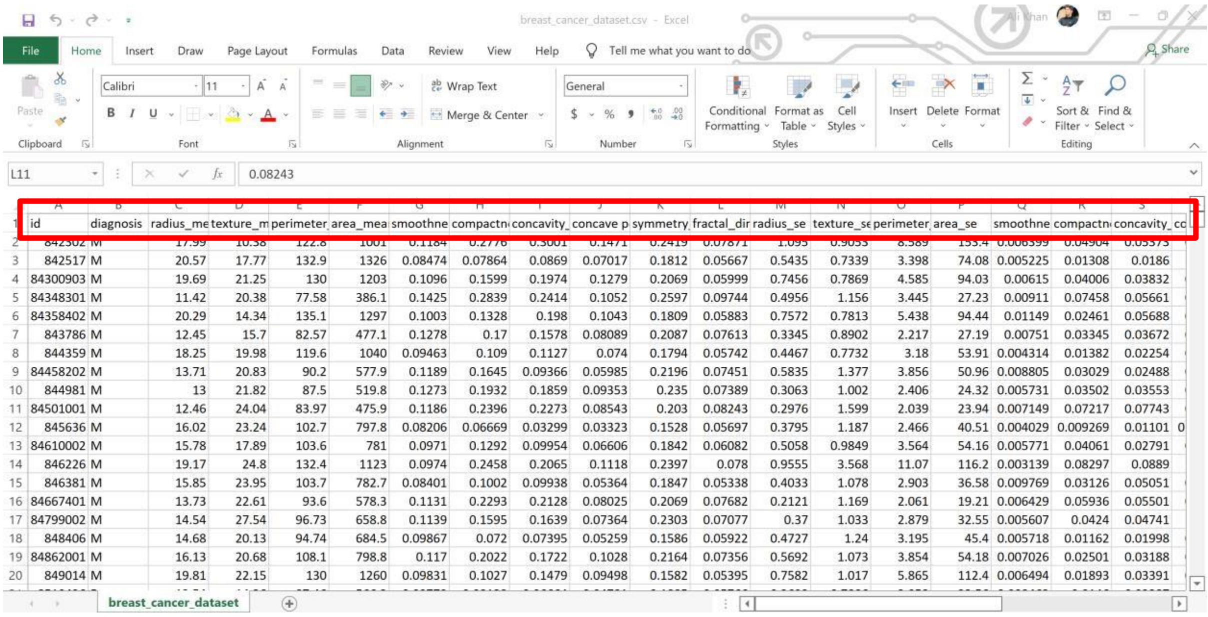
- ✞ Malignant tumors are those which have capability to move to the other parts of the body
- ✞ They are dangerous and called as cancerous tumors
- ✞ Fast growing

3. BREAST CANCER CLASSIFICATION:

This algorithm will classify the tumors as malignant or benign. That's why it is known as Breast Cancer Classification. Logistic Regression model will be trained on the dataset and then it will be tested on the new data.

4. DATASET:

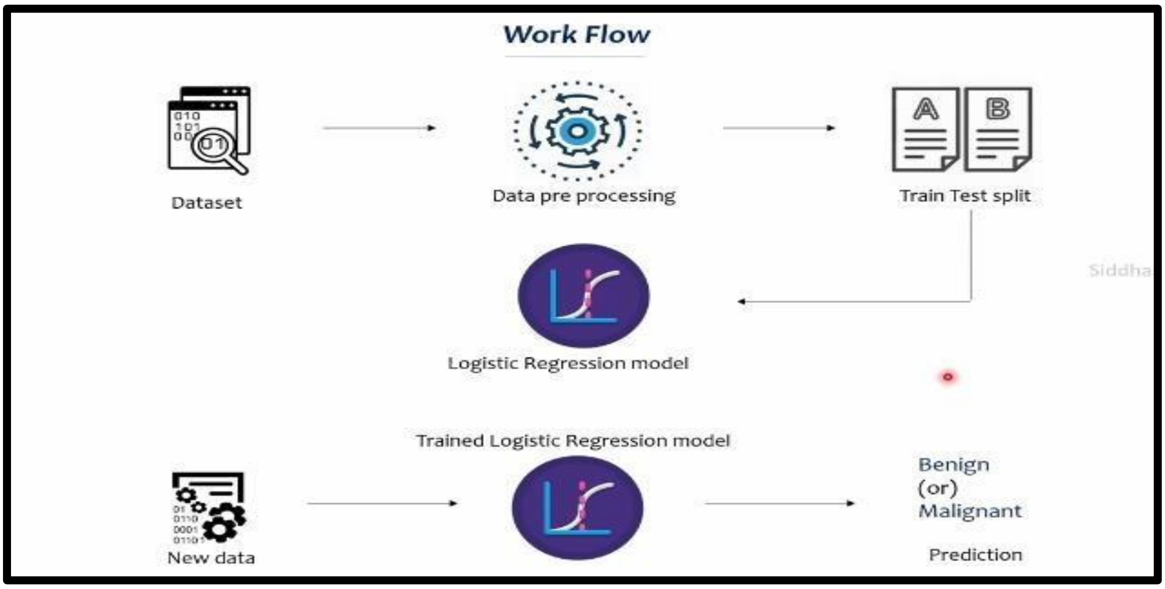
Fine needle aspiration: It is a type of biopsy procedure. In fine needle aspiration, a thin needle is inserted into an area of abnormal appearing tissue or body fluid. As with other types of biopsies, the sample collected during fine needle aspiration can help make a diagnosis or rule out conditions such as cancer. The data we use has been derived from this particular test called Fine needle aspiration. This is a standard procedure.



L11 | 0.08243

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	id	diagnosis	radius_me	texture_m	perimeter_	area_mea	smoothne	compactn	concavity_	concave p	symmetry_	fractal_dir	radius_se	texture_se	perimeter_	area_se	smoothne	compactn	concavity_
2	842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.006399	0.04904	0.05373
3	842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225	0.01308	0.0186
4	84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832
5	84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661
6	84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688
7	843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751	0.03345	0.03672
8	844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.004314	0.01382	0.02254
9	84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.008805	0.03029	0.02488
10	844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.005731	0.03502	0.03553
11	84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.007149	0.07217	0.07743
12	845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.004029	0.009269	0.01101
13	84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.005771	0.04061	0.02791
14	846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.003139	0.08297	0.0889
15	846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.009769	0.03126	0.05051
16	84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.006429	0.05936	0.05501
17	84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.005607	0.0424	0.04741
18	848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.005718	0.01162	0.01998
19	84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.007026	0.02501	0.03188
20	849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.006494	0.01893	0.03391

5. WORKFLOW:

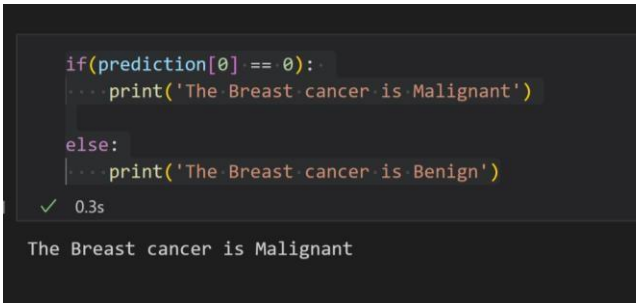

Work Flow
Dataset
Data pre processing
Train Test split
Siddha
Logistic Regression model
Trained Logistic Regression model
New data
Benign
(or)
Malignant
Prediction

6. PYTHON CODE:

```
import numpy as np import pandas as pd import
sklearn.datasets from sklearn.model_selection
import train_test_split from sklearn.linear_model
import LogisticRegression from sklearn.metrics
import accuracy_score
#loading data from sklearn breast_cancer_dataset =
sklearn.datasets.load_breast_cancer()
print(breast_cancer_dataset) #loading data to a data frame
data_frame = pd.DataFrame(breast_cancer_dataset.data, columns
= breast_cancer_dataset.feature_names) #print the first 5 rows
of the dataframe data_frame.head()
#adding the target column to the data frame data_frame['label']
= breast_cancer_dataset.target
#print last 5 rows of the dataframe data_frame.tail()
#number of rows and columns in the dataset
data_frame.shape
#getting some information about the data
data_frame.info()
#checking for missing values data_frame.isnull().sum()
#statistical measures about the data
data_frame.describe()
#checking the distribution of target
variable data_frame['label'].value_counts
#1-->benign, 0-->malignant
data_frame.groupby('label').mean()
#Separating the features and target
X = data_frame.drop(columns = 'label', axis =
1) Y = data_frame['label'] print(Y)
#Spilitting the data into training data and testing data
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,
test_size=0.2, random_state=2) print(X.shape, X_train.shape,
X_test.shape)
#Model training
#Logistic Regression
model = LogisticRegression()
#training the logistic regression model using training data
model.fit(X_train, Y_train)
#MODEL EVALUATION
#ACCURACY SCORE
#accuracy on training data
```

```
X_train_prediction = model.predict(X_train) training_data_accuracy =
accuracy_score(Y_train,  X_train_prediction) print('Accuracy on
training data = ', training_data_accuracy)
#accuracy on test data
X_test_prediction = model.predict(X_test) test_data_accuracy =
accuracy_score(Y_test,  X_test_prediction) print('Accuracy on
test data = ', test_data_accuracy)
#BUILDING A PREDICTIVE SYSTEM
input_data =




)

#change the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)
#reshape the numpy array as we are predicting for one
datapoint input_data_reshape =
input_data_as_numpy_array.reshape(1,-1) prediction =
model.predict(input_data_reshape) print(prediction)
if(prediction[0] == 0):
    print('The Breast cancer is Malignant')
 else:
    print('The Breast cancer is Benign')
```

```
(18.25,19.98,119.6,1040,0.09463,0.109,0.1127,0.074,0.1794,0.05742,0.4467,0.773
2,3.18,53.91,0.004314,0.01382,0.02254,0.01039,0.01369,0.002179,22.88,27.66,153
.2,1606,0.1442,0.2576,0.3784,0.1932,0.3063,0.08368
```



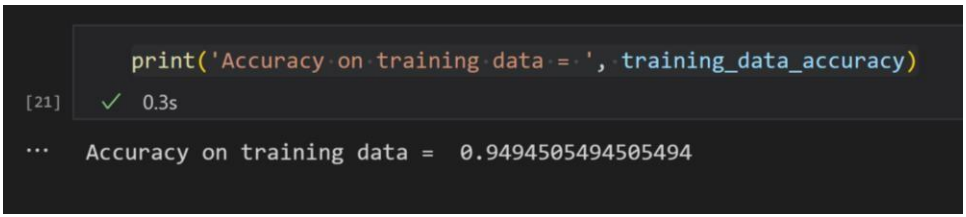
```
if(prediction[0] == 0):
    print('The Breast cancer is Malignant')

else:
    print('The Breast cancer is Benign')
```

✓ 0.3s

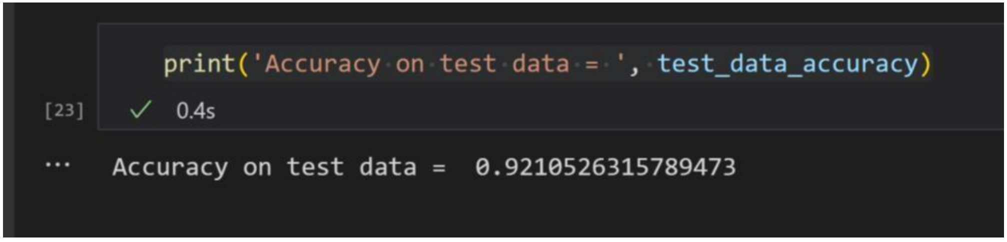
```
The Breast cancer is Malignant
```

7. CALCULATION BY ALGORITHM

- **Accuracy**



```
print('Accuracy on training data = ', training_data_accuracy)
[21] ✓ 0.3s

··· Accuracy on training data =  0.9494505494505494
```



```
print('Accuracy on test data = ', test_data_accuracy)
[23] ✓ 0.4s

··· Accuracy on test data =  0.9210526315789473
```

8. ABOUT LIBRARIES:

- **import numpy as np** (Used to make numpy arrays)
- **import pandas as pd** (Used to create pandas dataframe, which are helpful to analyze the process data in more structured way)
- **import sklearn.datasets**(Used to import the breast cancer data)
- **from sklearn.model_selection import train_test_split**(Splits the data into training and testing part)
- **from sklearn.linear_model import LogisticRegression**(Logistic regression is used because we have binary decision)
- **from sklearn.metrics import accuracy_score**(Used to evaluate our model i.e. how many correct predictions our model is making)