Dr. Ambedkar Institute of Technology for Handicapped, Kanpur - 208024, U.P.

AITH

न दैन्यम् न पलायनम्

BREAST CANCER DETECTION: USING MACHINE LEARNING

GUIDED BY

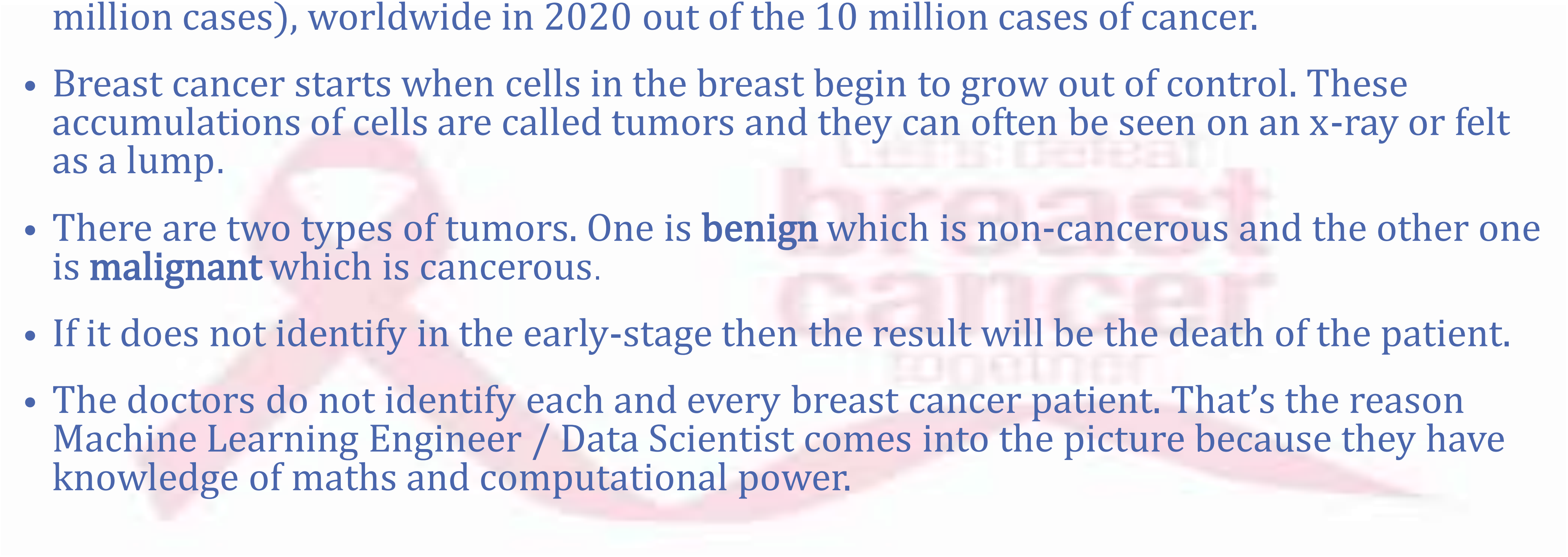
SRI NATH DWIVEDI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

ASHUTOSH SINGH (19)
SWAPNIL SIDDHARTHA (58)
UMANG RAJ CHAURASIA (63)

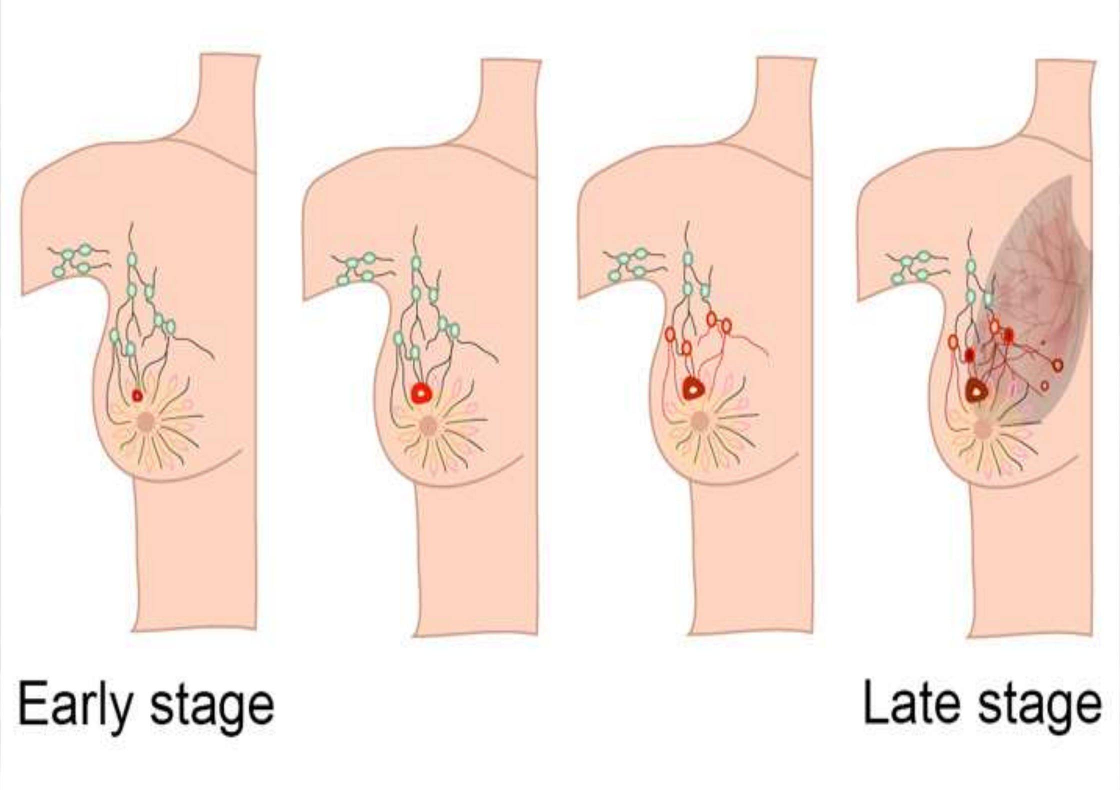
BREAST CANCER: AN OVERVIEW



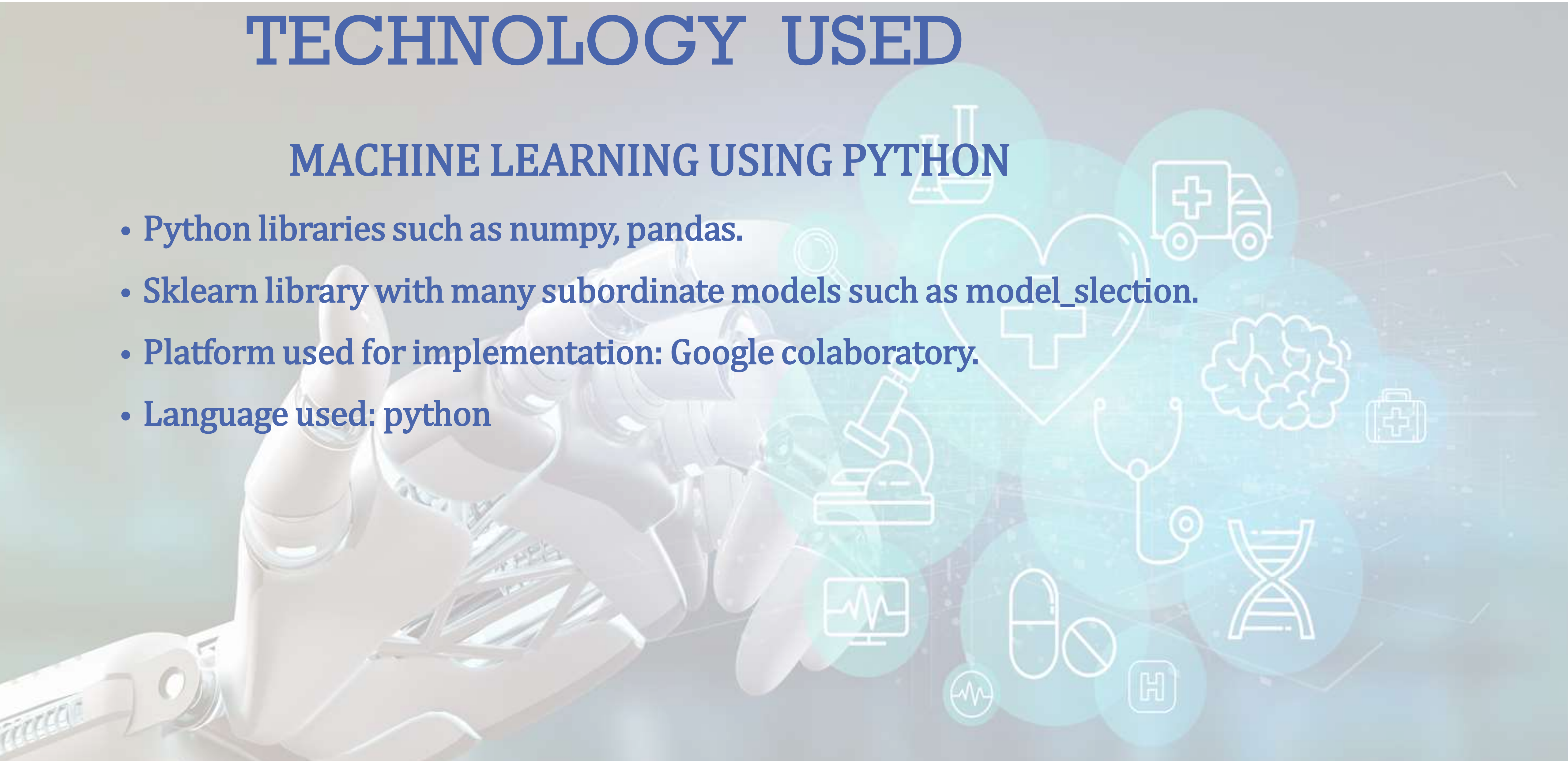
- Breast cancer is major cause of death in women around the world. According to WHO (World Health Organization), breast cancer accounted for maximum deaths (2.26 million cases), worldwide in 2020 out of the 10 million cases of cancer.
- Breast cancer starts when cells in the breast begin to grow out of control. These accumulations of cells are called tumors and they can often be seen on an x-ray or felt as a lump.
- There are two types of tumors. One is **benign** which is non-cancerous and the other one is **malignant** which is cancerous.
- If it does not identify in the early-stage then the result will be the death of the patient.
- The doctors do not identify each and every breast cancer patient. That's the reason Machine Learning Engineer / Data Scientist comes into the picture because they have knowledge of maths and computational power.

PROBLEM STATEMENT

- Predicting if the cancer diagnosis is benign or malignant based on several observations/features i.e its radius, area, smoothness, texture etc.
- Using Machine learning technologies to predict the tumors.



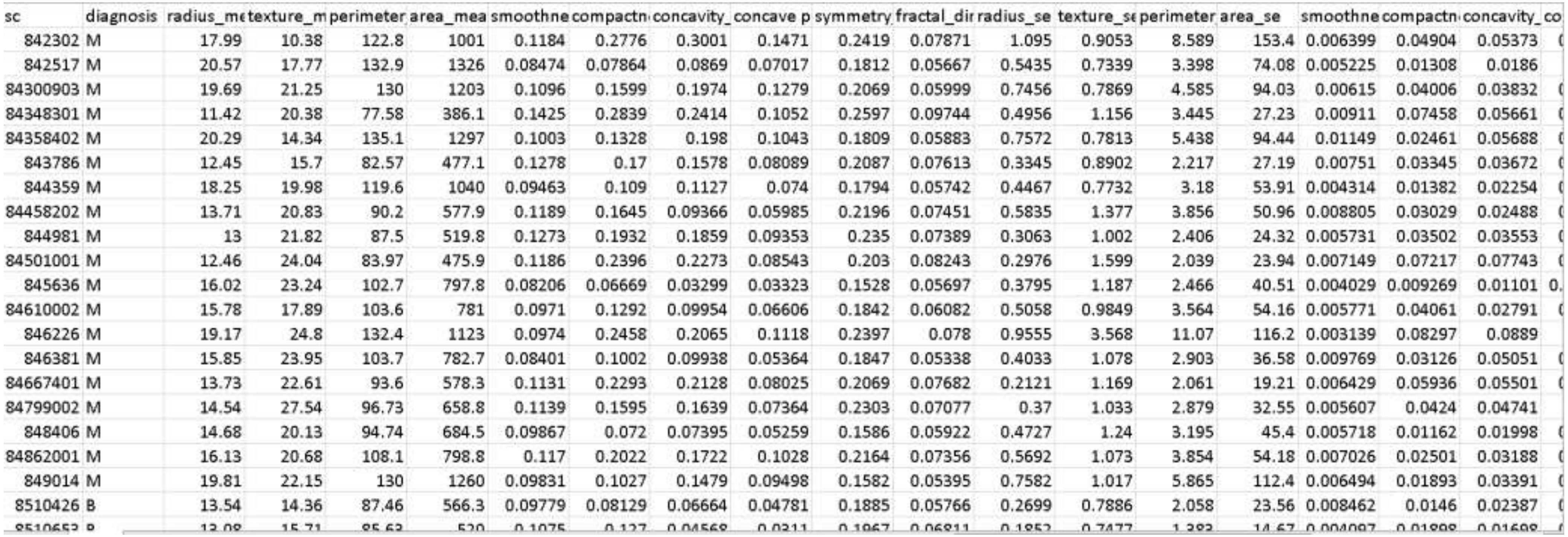
TECHNOLOGY USED



MACHINE LEARNING USING PYTHON

- Python libraries such as numpy, pandas.
- Sklearn library with many subordinate models such as model_slection.
- Platform used for implementation: Google colaboratory.
- Language used: python

DATASET USED

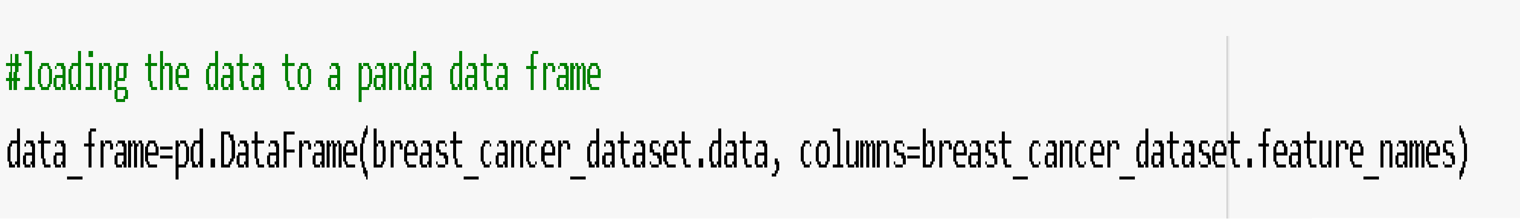


sc	diagnosis	radius_me	texture_m	perimeter	area_mea	smoothne	compactn	concavity_	concave p	symmetry	fractal_dir	radius_se	texture_se	perimeter	area_se	smoothne	compactn	concavity_	co
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.006399	0.04904	0.05373	0
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	
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	0
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	0
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	0
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	0
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	0
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	0
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	0
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	0
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	0.
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	0
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	
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	0
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	0
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	
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	0
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	0
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	0
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.008462	0.0146	0.02387	0
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.004097	0.01898	0.01698	0

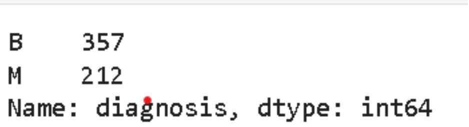
Source: Kaggle

EXPLORING AND PREPARING THE DATA

- Importing the data Importing the dataset in the google colaboratory

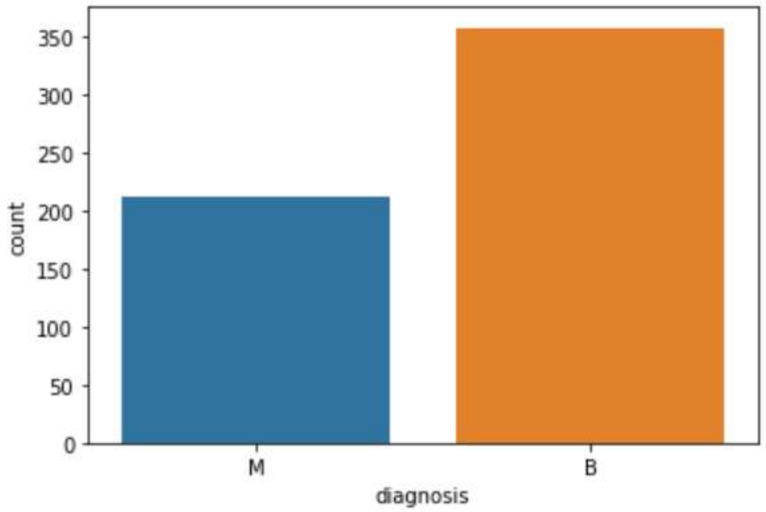


```
#loading the data to a panda data frame
data_frame=pd.DataFrame(breast_cancer_dataset.data, columns=breast_cancer_dataset.feature_names)
```

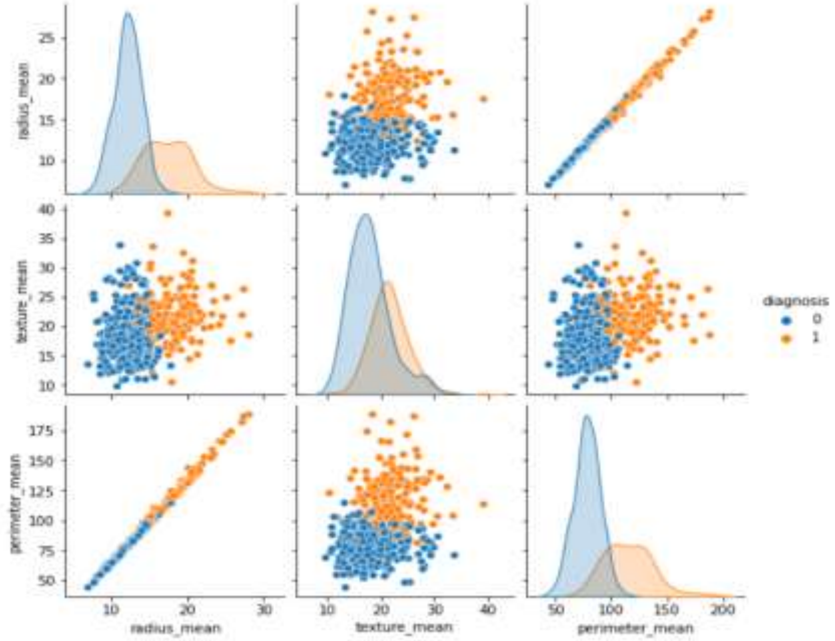


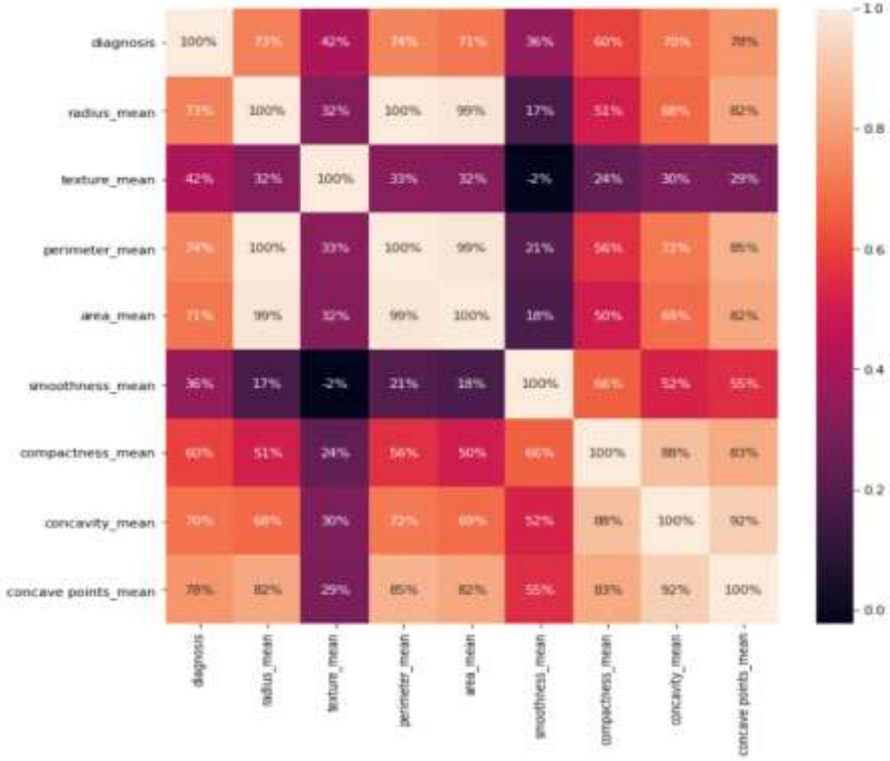
```
B    357
M    212
Name: diagnosis, dtype: int64
```

- This graph indicate that 357 masses are benign and 212 malignant



VISUALIZATION

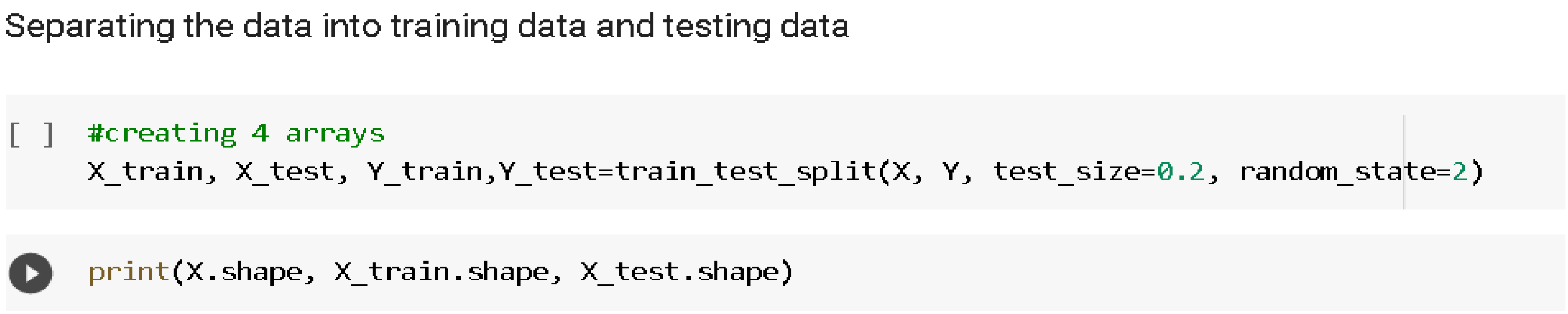




A vector x of numeric values, and for each value of x, subtracts the minimum value in x and divides by the range of values in x.

DATA PREPARATION

- Separating the dataset into training and testing dataset
- 80% of the dataset is used to train the model and remaining to test the model.



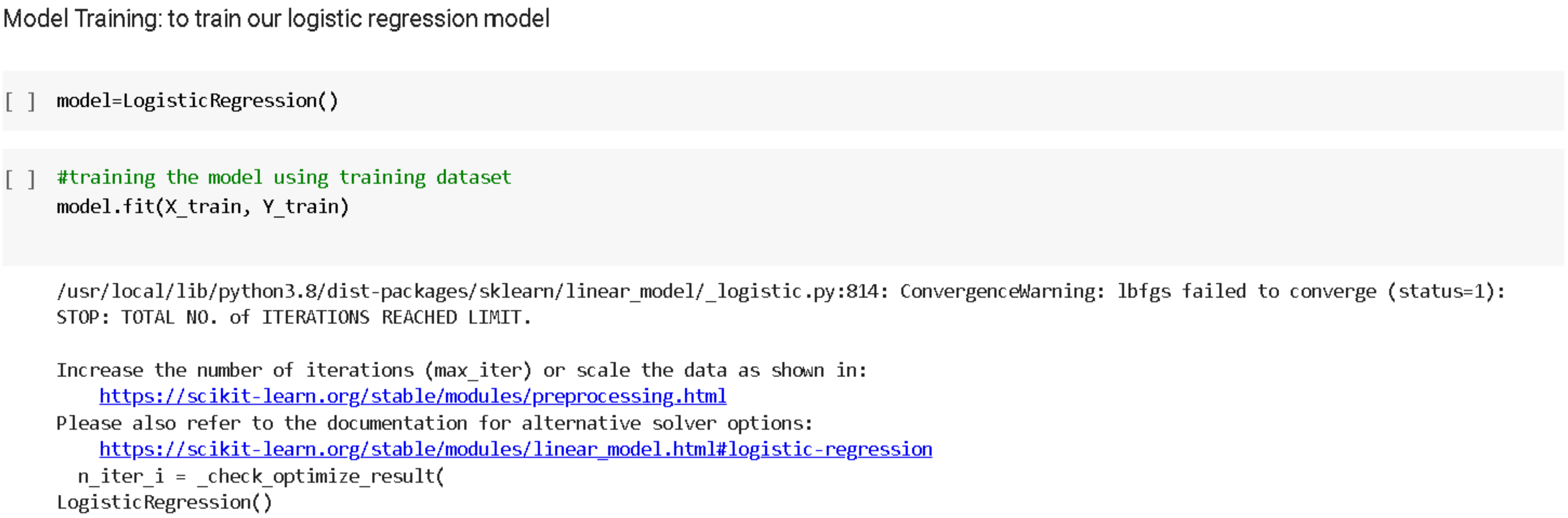
Separating the data into training data and testing data

```
[ ]  #creating 4 arrays
     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)
```

TRAINING THE MODEL

- Training the model on the data



Model Training: to train our logistic regression model

```
[ ]  model=LogisticRegression()
```

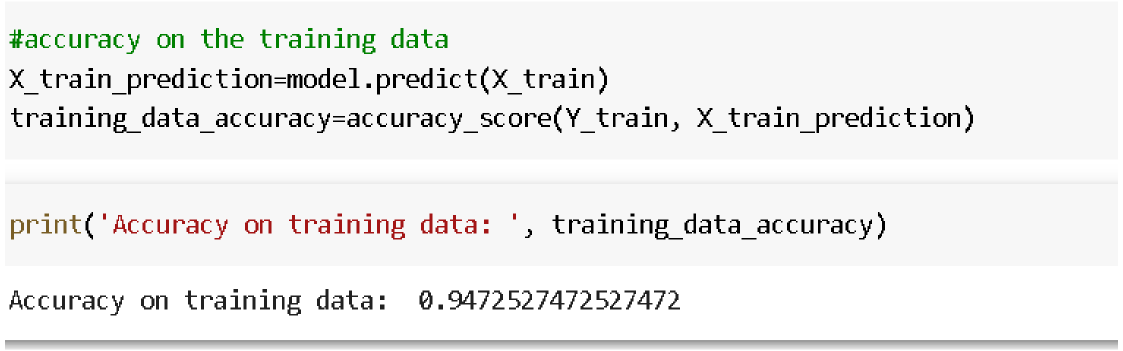
```
[ ]  #training the model using training dataset
     model.fit(X_train, Y_train)
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/linear_model/_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
LogisticRegression()
```

ACCURACY SCORE

To check the accuracy of the training data

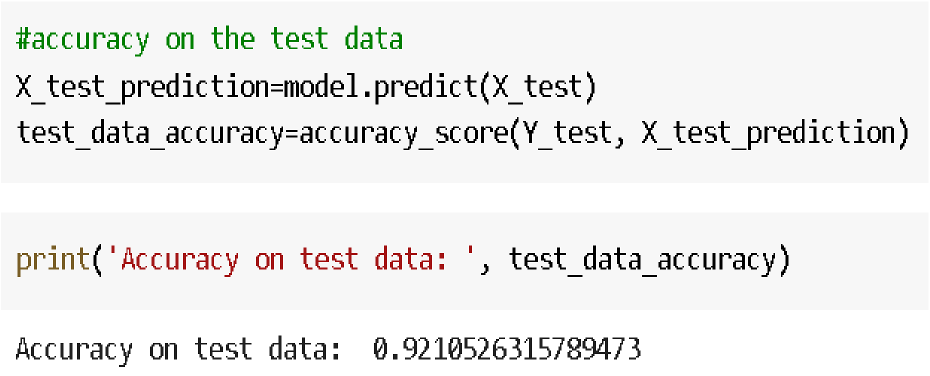


```
#accuracy on the training data
X_train_prediction=model.predict(X_train)
training_data_accuracy=accuracy_score(Y_train, X_train_prediction)
```

```
print('Accuracy on training data: ', training_data_accuracy)
```

```
Accuracy on training data:  0.9472527472527472
```

To check the accuracy of the test data



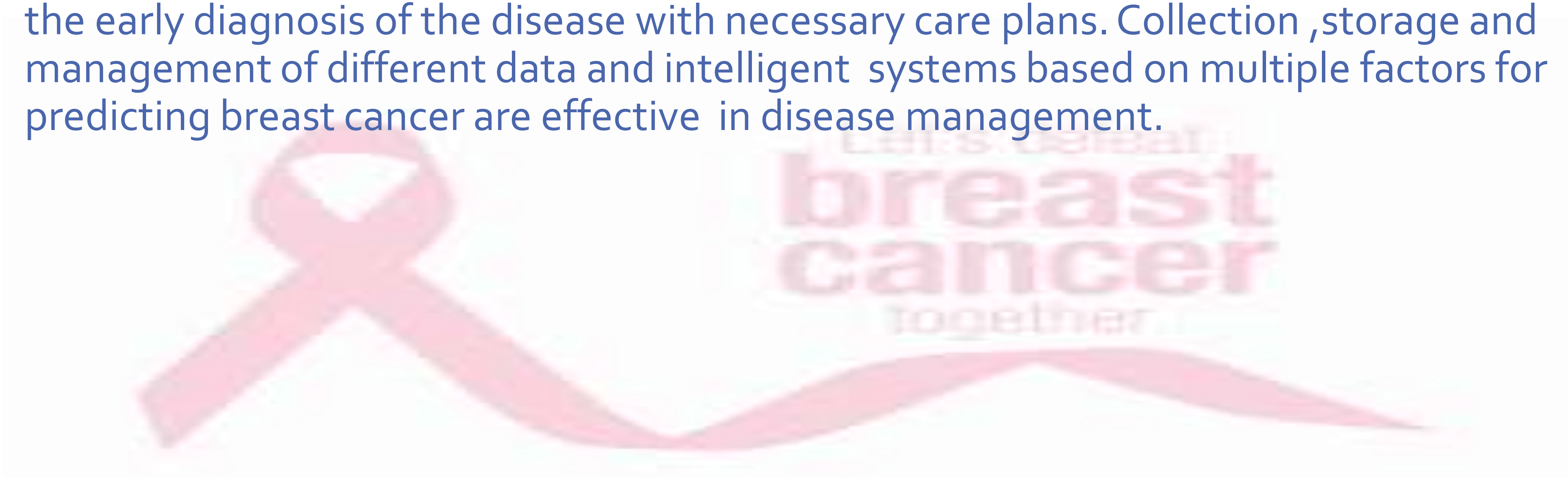
```
#accuracy on the test data
X_test_prediction=model.predict(X_test)
test_data_accuracy=accuracy_score(Y_test, X_test_prediction)
```

```
print('Accuracy on test data: ', test_data_accuracy)
```

```
Accuracy on test data:  0.9210526315789473
```

CONCLUSION

Combining multiple risk factors in modelling for breast cancer prediction could help the early diagnosis of the disease with necessary care plans. Collection ,storage and management of different data and intelligent systems based on multiple factors for predicting breast cancer are effective in disease management.



THANK YOU