A Mini Project Report on

“Cancer Detection”

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Computer Engineering

by

Nilesh Parab (D221452)
Ananya Yadav (D221427)
Riya Singh (D221421)
Vivek Tiwari(D221422)

Under the Guidance of

Prof. Uma Ade

Department of Computer Engineering
Watumull Institute of Electronics Engineering and Computer Technology
Ulhasnagar

UNIVERSITY OF MUMBAI

Academic Year 2022-2023

Approval Sheet

This Mini Project Report entitled ***"Cancer Detection"*** Submitted by ***"Nilesh Parab" (D221452), "Ananya Yadav" (D221427), "Riya Singh" (D221421), "Vivek Tiwari" (D221422)*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Computer Engineering*** from ***University of Mumbai***.

Under The Guidance of

Prof. Uma Ade

Prof. Dhanajay Raut

Head Department of Computer Engineering

Place: Ulhasnagar

Date: 11.11.2022

CERTIFICATE

This is to certify that the mini project entitled ***"Cancer Detection"*** Submitted by ***"Nilesh Parab" (D221452), "Ananya Yadav" (D221427), "Riya Singh " (D221421), "Vivek Tiwari" (D221422)*** or the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer Engineering***, to the University of Mumbai, is a bonafide work carried out during academic year 2022-2023.

Guide Name & Signature

Examiners:

1.

2.

Prof. Dhanajay Raut

Head Department of Computer Engineering

Principal

Place: Ulhasnagar

Date:11.11.2022

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Nilesh Parab (D221452)
Ananya Yadav (D221427)
Riya Singh (D221421)
Vivek Tiwari(D221422)

Date: 11.11.2022

Abstract

Cancer has been described as a diverse illness with a wide range of subgroups. Early cancer diagnosis and prognosis are essential for clinical patient treatment, which has become a requirement in cancer research. Numerous research teams from the biomedical and bioinformatics fields have studied the use of machine learning (ML) techniques due to the significance of categorizing cancer patients into high or low risk groups. These methods have been applied in an effort to simulate the development and management of malignant diseases. Furthermore, their significance is demonstrated by the fact that ML algorithms can recognize important features in complicated datasets. Even though it is evident that the use of ML methods can improve our understanding of cancer progression, an appropriate level of validation is needed in order for these methods to be considered in the everyday clinical practice.

The predictive models discussed here are based on various supervised ML techniques as well as on different input features and data samples. Given the growing trend on the application of ML methods in cancer research, we present here the most recent publications that employ these techniques as an aim to model cancer risk or patient outcomes.

CONTENTS

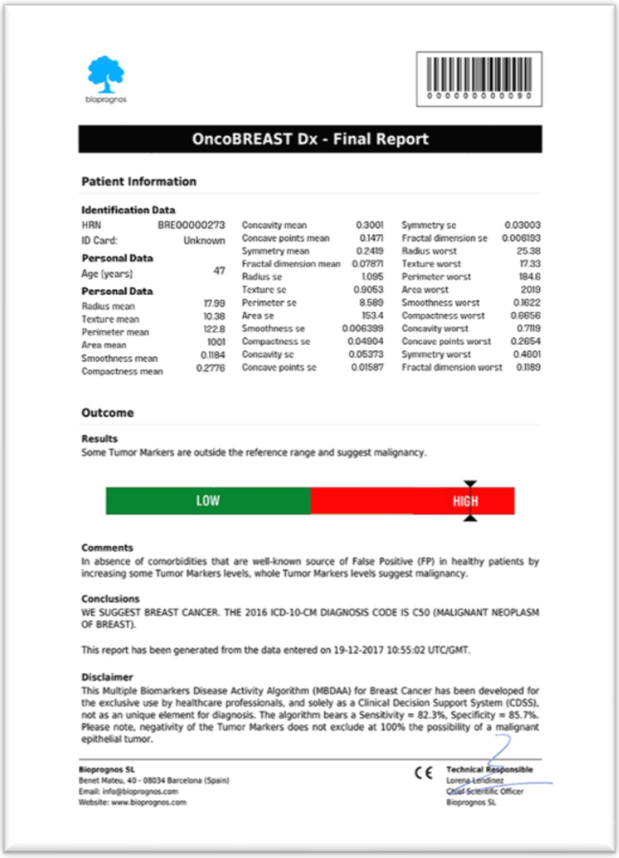
Chapter 1: Introduction

Over the past few decades, there has been continuous development in cancer research. To identify certain cancer types before symptoms, emerge, researchers have employed a number of strategies, such as early-stage screening. They have also developed novel techniques for the early prognosis of cancer therapy. Thanks to the advancement of modern medical technology, massive amounts of data on cancer have been acquired and are now accessible to the scientific community. Making an accurate forecast of a disease's trajectory, however, is one of the most exciting and challenging issues facing doctors. As a result, scientists working on medical research increasingly frequently use ML approaches. Even though these datasets are complex, these approaches can identify patterns and correlations among them.

According to statistics from around the world, breast cancer (BC) is one of the most prevalent malignancies in women and accounts for a considerable portion of new cancer cases and cancer-related fatalities in today's society. Early BC diagnosis might encourage patients to receive prompt clinical care, which can dramatically improve their prognosis and likelihood of survival. Patients may avoid receiving therapies they do not need if benign tumors are classified more accurately. Therefore, substantial research is being done on how to correctly diagnose BC and classify people into groups that are malignant or benign. Machine learning (ML) is widely acknowledged as the preferred technology in BC pattern categorization and forecast modeling due to its distinct benefits in identifying essential characteristics from complex BC datasets. Data may be effectively categorized using methods like data mining and classification. Particularly in the medical industry, where those techniques are frequently utilized to make judgments through diagnosis and analysis.

Currently, a technician operating the CT scan scanner at a clinic reports the cancer stage, but this information may be inaccurate because the technician may not be qualified to predict cancer stage. Our objective is to develop a model that can accurately predict cancer malignancy based on qualities without the requirement for qualifying.

Chapter 2: Problem Statement



OncoBREAST Dx - Final Report

Patient Information

Identification Data

HRN	BRE00000273	Concavity mean	0.3001	Symmetry se	0.03003
ID Card:	Unknown	Concave points mean	0.1471	Fractal dimension se	0.006193
Personal Data		Symmetry mean	0.2419	Radius worst	25.38
Age (years)	47	Fractal dimension mean	0.07871	Texture worst	17.33
		Radius se	1.095	Perimeter worst	184.6
Personal Data		Texture se	0.9053	Area worst	2019
Radius mean	17.99	Perimeter se	8.589	Smoothness worst	0.1622
Texture mean	10.38	Area se	153.4	Compactness worst	0.6656
Perimeter mean	122.8	Smoothness se	0.006399	Concavity worst	0.7119
Area mean	1001	Compactness se	0.04904	Concave points worst	0.2654
Smoothness mean	0.1184	Concavity se	0.05373	Symmetry worst	0.4601
Compactness mean	0.2776	Concave points se	0.01587	Fractal dimension worst	0.1189

Outcome

Results

Some Tumor Markers are outside the reference range and suggest malignancy.

LOW HIGH

Comments

In absence of comorbidities that are well-known source of False Positive (FP) in healthy patients by increasing some Tumor Markers levels, whole Tumor Markers levels suggest malignancy.

Conclusions

WE SUGGEST BREAST CANCER. THE 2016 ICD-10-CM DIAGNOSIS CODE IS C50 (MALIGNANT NEOPLASM OF BREAST).

This report has been generated from the data entered on 19-12-2017 10:55:02 UTC/GMT.

Disclaimer

This Multiple Biomarkers Disease Activity Algorithm (MBDAA) for Breast Cancer has been developed for the exclusive use by healthcare professionals, and solely as a Clinical Decision Support System (CDSS), not as an unique element for diagnosis. The algorithm bears a Sensitivity = 82.3%, Specificity = 85.7%. Please note, negativity of the Tumor Markers does not exclude at 100% the possibility of a malignant epithelial tumor.

Bioprognos SL
Benet Mateu, 40 - 08034 Barcelona (Spain)
Email: info@bioprognos.com
Website: www.bioprognos.com

Technical Responsible
Lorena Lendínez
Chief Scientific Officer
Bioprognos SL

- Early diagnosis of cancer focuses on detecting symptomatic patients as early as possible so they have the best chance for successful treatment.

- The technical member simply identified the data's patience stages.

- The technical person is unaware about cancer sickness, cancer malignancies, or any other issues associated with them.

- Therefore, patience sometimes addresses the subject of death as well as some other disease linked problems.

- Currently, the machine operator is forecasting the findings of the CT scan reports.

- Machine operators are not very skilled in identifying the stage of malignancy.

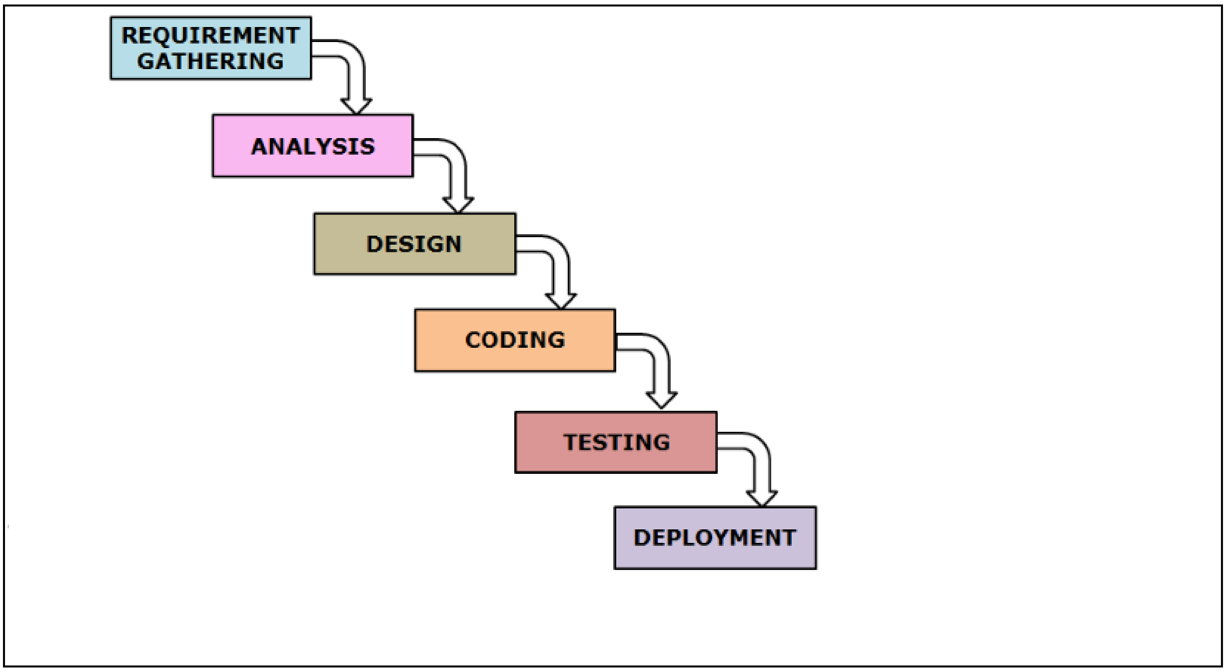
- In this situation, there is a strong probability that the outcome will be incorrect.

Chapter 3: Implementation and Approach

We started with choosing the tools first, in terms of how we'll build the web applications I started doingfeasibility study about that we found python will work well form us in terms of strong predictions andanalyzing. The data used in this project was downloaded from UCI We started finding some of the frontend development tools which can help us to give good look and feel for project, We discovered some of the tools like Google Collab, Jupyter Notebook etc. We decided to go with Python. Python is across-functional, maximally interpreted language that has lots of advantages to offer. The object- oriented programming language is commonly used to streamline large complex data sets. Over and above, having a dynamic semantics plus unmeasured capacities of *RAD(rapid applicationdevelopment)*, Python is heavily utilized to script as well. We started deciding modules like analyzing the highest rate of crime taking place in which area. Now we had to choose my software development approach which will be better for my idea, I decided to choose waterfall model.

Waterfall Approach:

Development activities are performed in order, with possibly minor overlap, but with little or no iteration between activities. User needs are determined, requirements are defined, and the full system is designed, built, and tested for ultimate delivery at one point in time. A document driven approach best suited for highly precedence systems with stable requirements. The waterfall model is often also referred to as the linear and sequential model, for the flow of activities in this model are rather linear and sequential as the name suggests. In this model, the software development activities move to the next phase only after the activities in the current phase are over. However, like is the case with a waterfall, one cannot return to the previous stage



Chapter 4: Technology Used

Pandas:

Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named NumPy, which provides support for multi-dimensional arrays.

NumPy

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning

Matplotlib

In Machine learning, it helps to understand the huge amount of data through different visualizations.

scikit learn

It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

Chapter 5: Predicting Cancer Stage

5.1 American Cancer Society

5.1.1 CT scan for Cancer

A CT scan (also known as a computed tomography scan, CAT scan, and spiral or helical CT) can help doctors find cancer and show things like a tumor's shape and size. CT scans are most often an outpatient procedure. The scan is painless and takes about 10 to 30 minutes.

5.1.2 What does a CT scan show?

CT scans show a slice, or cross-section, of the body. The image shows your bones, organs, and soft tissues more clearly than standard x-rays.

CT scans can show a tumor's shape, size, and location. They can even show the blood vessels that feed the tumor – all without having to cut into the patient.

Doctors often use CT scans to help them guide a needle to remove a small piece of tissue. This is called a CT-guided biopsy. CT scans can also be used to guide needles into tumors for some types of cancer treatments, such as radiofrequency ablation (RFA), which uses heat to destroy a tumor.

By comparing CT scans done over time, doctors can see how a tumor is responding to treatment or find out if the cancer has come back after treatment.

5.1.3 How does a CT scan work?

In a way, CT scans are like standard x-ray tests. But an x-ray test aims a broad beam of radiation from only one angle. A CT scan uses a pencil-thin beam to create a series of pictures taken from different angles. The information from each angle is fed into a computer, which then creates a black and white picture that shows a slice of a certain area of the body – much like looking at a single slice from a loaf of bread.

Special contrast materials can be used to get a clearer picture. These can be swallowed as a liquid, put into a vein, or put into the intestines through the rectum as an enema.

By layering CT image slices on top of each other, the machine can create a 3-dimensional (3-D) view. The 3-D image can be rotated on a computer screen to look at different angles.

Doctors are now taking CT technology one step further in a technique called virtual endoscopy. They can look at the inside surfaces of organs such as the lungs (virtual bronchoscopy) or colon (virtual colonoscopy or CT colonography) without actually having to put scopes into the body. The 3-D CT images are arranged to create a black and white view on the computer screen. This looks a lot like it would if they were doing an actual endoscopy.

5.2 Cancer.Net

Staging is a way to describe a cancer. The cancer's stage tells you where a cancer is located and its size, how far it has grown into nearby tissues, and if it has spread to nearby lymph nodes or other parts of the body. Before starting any cancer treatment, doctors may use physical exams, imaging scans, and other tests to determine a cancer's stage. Staging may not be completed until all the tests are finished.

5.2.1 Why does cancer stage matter?

Staging helps your doctor plan the best treatment. This may include choosing a type of surgery and whether or not to use chemotherapy or radiation therapy. Knowing the cancer stage lets your entire health care team talk about your diagnosis in the same way.

5.2.2 Doctors can also use staging to:

- Understand the chance that the cancer will come back or spread after the original treatment.
- Help forecast the prognosis, which is the chance of recovery
- Help determine which cancer clinical trials may be open to you.
- See how well a treatment worked
- Compare how well new treatments work among large groups of people with the same diagnosis

Chapter 6: Description of Dataset

6.1 Name: Diagnostic Wisconsin Breast Cancer Database

6.2 Source:

6.2.1 Creators:

1. Dr. William H. Wolberg, General Surgery Dept.

University of Wisconsin, Clinical Sciences Center

Madison, WI 53792

2. W. Nick Street, Computer Sciences Dept.

University of Wisconsin, 1210 West Dayton St., Madison, WI 53706

3. Olvi L. Mangasarian, Computer Sciences Dept.

University of Wisconsin, 1210 West Dayton St., Madison, WI 53706

6.2.2 Donor: Nick Street

6.2.3 Date: November 1995

6.3 Data Set Information:

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. A few of the images can be found at [Web Link]

Separating plane described above was obtained using Multi surface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

The actual linear program used to obtain the separating plane in the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:
ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

6.4 Attribute Information:

1) ID number
2) Diagnosis (M = malignant, B = benign)
3-32)
Ten real-valued features are computed for each cell nucleus:

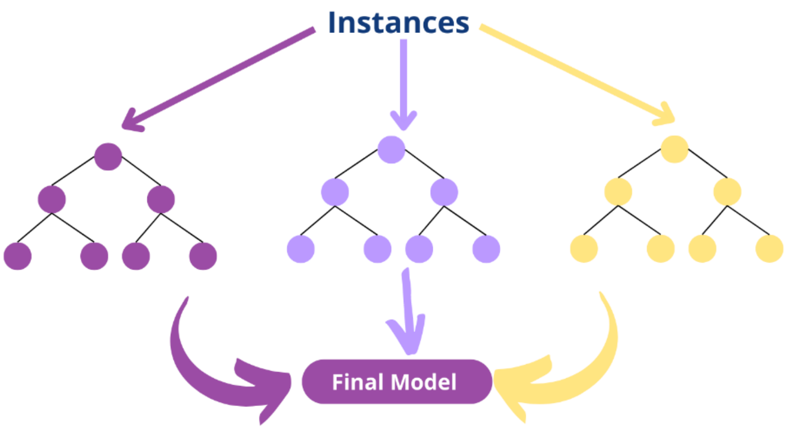
a) radius (mean of distances from centre to points on the perimeter)
b) texture (standard deviation of grey-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

6.5 Dataset:

1	id	diagnosis	radius_m	texture_m	perimeter	area_mea	smoothne	compactr	concavity	concave p	symmetry	fractal_d	radius_se	texture_s	perimeter	area_se	smoothne	compactr	concavity	concave p	symmetry	fractal_d	radius_w	texture_w	perimeter	area_wor	smoothne	compactr	concavity	concave p	symmetry	fractal_dimension_worst
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	0.0064	0.04904	0.05373	0.01587	0.03003	0.00619	25.38	17.33	184.6	2019	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189
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	0.00523	0.01308	0.0186	0.0134	0.01389	0.00353	24.99	23.41	158.8	1956	0.1238	0.1866	0.2416	0.186	0.275	0.08902
4	8.4E+07	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	0.02058	0.0225	0.00457	23.57	25.53	152.5	1709	0.1444	0.4245	0.4504	0.243	0.3613	0.08758
5	8.4E+07	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	0.01867	0.05963	0.00921	14.91	26.5	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.173
6	8.4E+07	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	0.01885	0.01756	0.00512	22.54	16.67	152.2	1575	0.1374	0.205	0.4	0.1625	0.2364	0.07678
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	0.01137	0.02165	0.00508	15.47	23.75	103.4	741.6	0.1791	0.5249	0.5355	0.1741	0.3985	0.1244
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	0.00431	0.01382	0.02254	0.01039	0.01369	0.00218	22.88	27.66	153.2	1606	0.1442	0.2576	0.3784	0.1932	0.3063	0.08368
9	8.4E+07	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	0.00881	0.03029	0.02488	0.01448	0.01486	0.00541	17.06	28.14	110.6	897	0.1654	0.3682	0.2678	0.1556	0.3196	0.1151
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	0.00573	0.03502	0.03553	0.01226	0.02143	0.00375	15.49	30.73	106.2	739.3	0.1703	0.5401	0.539	0.206	0.4378	0.1072
11	8.5E+07	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	0.00715	0.07217	0.07743	0.01432	0.01789	0.01008	15.09	40.68	97.65	711.4	0.1853	1.058	1.105	0.221	0.4366	0.2075
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	0.00403	0.00927	0.01101	0.00759	0.0146	0.00304	19.19	33.88	123.8	1150	0.1181	0.1551	0.1459	0.09975	0.2948	0.08452
13	8.5E+07	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	0.00577	0.04061	0.02791	0.01282	0.02008	0.00414	20.42	27.28	136.5	1299	0.1396	0.5609	0.3965	0.181	0.3792	0.1048
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	0.00314	0.08297	0.0889	0.0409	0.04484	0.01284	20.96	29.94	151.7	1332	0.1037	0.3903	0.3639	0.1767	0.3176	0.1023
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	0.00977	0.03126	0.05051	0.01992	0.02981	0.003	16.84	27.66	112	876.5	0.1131	0.1924	0.2322	0.1119	0.2809	0.06287
16	8.5E+07	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	0.00643	0.05936	0.05501	0.01628	0.01961	0.00809	15.03	32.01	108.8	697.7	0.1651	0.7725	0.6943	0.2208	0.3596	0.1431
17	8.5E+07	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	0.00561	0.0424	0.04741	0.0109	0.01857	0.00547	17.46	37.13	124.1	943.2	0.1678	0.6577	0.7026	0.1712	0.4218	0.1341
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	0.00572	0.01162	0.01998	0.01109	0.0141	0.00209	19.07	30.88	123.4	1138	0.1464	0.1871	0.2914	0.1609	0.3029	0.08216
19	8.5E+07	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	0.00703	0.02501	0.03188	0.01297	0.01689	0.00414	20.96	31.48	136.8	1315	0.1789	0.4233	0.4784	0.2073	0.3706	0.1142
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	0.00649	0.01893	0.03391	0.01521	0.01356	0.002	27.32	30.88	186.8	2398	0.1512	0.315	0.5372	0.2388	0.2768	0.07615
21	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	0.00846	0.0146	0.02387	0.01315	0.0198	0.0023	15.11	19.26	99.7	711.2	0.144	0.1773	0.239	0.1288	0.2977	0.07259
22	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	0.0041	0.01898	0.01698	0.00649	0.01678	0.00243	14.5	20.49	96.09	630.5	0.1312	0.2776	0.189	0.07283	0.3184	0.08183
23	8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.00961	0.01432	0.01985	0.01421	0.02027	0.00297	10.23	15.66	65.13	314.9	0.1324	0.1148	0.08867	0.06227	0.245	0.07773
24	8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.00679	0.05328	0.06446	0.02252	0.03672	0.00439	18.07	19.08	125.1	980.9	0.139	0.5954	0.6305	0.2393	0.4667	0.09946
25	851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.00473	0.01259	0.01715	0.01038	0.01083	0.00199	29.17	35.59	188	2615	0.1401	0.26	0.3155	0.2009	0.2822	0.07526
26	852552	M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455	102.6	0.00605	0.01882	0.02741	0.0113	0.01468	0.0028	26.46	31.56	177	2215	0.1805	0.3578	0.4695	0.2095	0.3613	0.09564
27	852631	M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276	111.4	0.00803	0.03799	0.03732	0.02397	0.02308	0.00744	22.25	21.4	152.4	1461	0.1545	0.3949	0.3853	0.255	0.4066	0.1059
28	852763	M	14.58	21.53	97.41	644.8	0.1054	0.1868	0.1425	0.08783	0.2252	0.06924	0.2545	0.9832	2.11	21.05	0.00445	0.03055	0.02681	0.01352	0.01454	0.00371	17.62	33.21	122.4	896.9	0.1525	0.6643	0.5539	0.2701	0.4264	0.1275
29	852781	M	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.00422	21.31	27.26	139.9	1403	0.1338	0.2117	0.3446	0.149	0.2341	0.07421
30	852973	M	15.3	25.27	102.4	732.4	0.1082	0.1697	0.1683	0.08751	0.1926	0.0654	0.439	1.012	3.498	43.5	0.00523	0.03057	0.03576	0.01083	0.01768	0.00297	20.27	36.71	149.3	1269	0.1641	0.611	0.6335	0.2024	0.4027	0.09876
31	853201	M	17.57	15.05	115	955.1	0.09847	0.1157	0.09875	0.07953	0.1739	0.06149	0.6003	0.8225	4.655	61.1	0.00563	0.03033	0.03407	0.01354	0.01925	0.00374	20.01	19.52	134.9	1227	0.1255	0.2812	0.2489	0.1456	0.2756	0.07919
32	853401	M	18.63	25.11	124.8	1088	0.1064	0.1887	0.2319	0.1244	0.2183	0.06197	0.8307	1.466	5.574	105	0.00625	0.03374	0.05196	0.01158	0.02007	0.00456	23.15	34.01	160.5	1670	0.1491	0.4257	0.6133	0.1848	0.3444	0.09782
33	853612	M	11.84	18.7	77.93	440.6	0.1109	0.1516	0.1218	0.05182	0.2301	0.07799	0.4825	1.03	3.475	41	0.00555	0.03414	0.04205	0.01044	0.02273	0.00567	16.82	28.12	119.4	888.7	0.1637	0.5775	0.6956	0.1546	0.4761	0.1402
34	8.5E+07	M	17.02	23.98	112.8	899.3	0.1197	0.1496	0.2417	0.1203	0.2248	0.06382	0.6009	1.398	3.999	67.78	0.00827	0.03082	0.05042	0.01112	0.02102	0.00385	20.88	32.09	136.1	1344	0.1634	0.3559	0.5588	0.1847	0.353	0.08482
35	854002	M	19.27	26.47	127.9	1162	0.09401	0.1719	0.1657	0.07593	0.1853	0.06261	0.5558	0.6062	3.528	68.17	0.00502	0.03318	0.03497	0.00964	0.01543	0.0039	24.15	30.9	161.4	1813	0.1509	0.659	0.6091	0.1785	0.3672	0.1123
36	854039	M	16.13	17.88	107	807.2	0.104	0.1559	0.1354	0.07752	0.1998	0.06515	0.334	0.6857	2.183	35.03	0.00419	0.02868	0.02664	0.00907	0.01703	0.00382	20.21	27.26	132.7	1261	0.1446	0.5804	0.5274	0.1864	0.427	0.1233
37	854253	M	16.74	21.59	110.1	869.5	0.0961	0.1336	0.1348	0.06018	0.1896	0.05656	0.4615	0.9197	3.008	45.19	0.00578	0.02499	0.03695	0.01195	0.02789	0.00267	20.01	29.02	133.5	1229	0.1563	0.3835	0.5409	0.1813	0.4863	0.08633
38	854268	M	14.25	21.72	93.63	633	0.09823	0.1098	0.1319	0.05598	0.1885	0.06125	0.286	1.019	2.657	24.91	0.00588	0.02995	0.04815	0.01161	0.02028	0.00402	15.89	30.36	116.2	799.6	0.1446	0.4238	0.5186	0.1447	0.3591	0.1014
39	854941	B	13.03	18.42	82.61	523.8	0.08983	0.03766	0.02562	0.02923	0.1467	0.05863	0.1839	2.342	1.17	14.16	0.00435	0.0049	0.01343	0.01164	0.02671	0.00178	13.3	22.81	84.46	545.9	0.09701	0.04619	0.04833	0.05013	0.1987	0.06169
40	855133	M	14.99	25.2	95.54	698.8	0.09387	0.05131	0.02398	0.02899	0.1565	0.05504	1.214	2.188	8.077	106	0.00688	0.01094	0.01818	0.01917	0.00788	0.00175	14.99	25.2	95.54	698.8	0.09387	0.05131	0.02398	0.02899	0.1565	0.05504
41	855138	M	13.48	20.82	88.4	559.2	0.1016	0.1255	0.1063	0.05439	0.172	0.06419	0.213	0.5914	1.545	18.52	0.00537	0.02239	0.03049	0.01262	0.01377	0.00319	15.53	26.02	107.3	740.4	0.161	0.4225	0.503	0.2258	0.2807	0.1071
42	855167	M	13.44	21.58	86.18	563	0.08162	0.06031	0.0311	0.02031	0.1784	0.05587	0.2385	0.8265	1.572	20.53	0.00328	0.01102	0.0139	0.00688	0.0138	0.00129	15.93	30.25	102.5	787.9	0.1094	0.2043	0.2085	0.1112	0.2994	0.07146
43	855563	M	10.95	21.35	71.9	371.1	0.1227	0.1218	0.1044	0.05669	0.1895	0.0687	0.2366	1.428	1.822	16.97	0.00806	0.01764	0.02595	0.01037	0.01357	0.00304	12.84	35.34	87.22	514	0.1909	0.2698	0.4023	0.1424	0.2964	0.09606
44	855625	M	19.07	24.81	128.3	1104	0.09081	0.219	0.2107	0.09961	0.231	0.06343	0.9811	1.666	8.83	104.9	0.00655	0.1006	0.09723	0.02638	0.05333	0.00765	24.09	33.17	177.4	1651	0.1247	0.7444	0.7242	0.2493	0.467	0.1038
45	856106	M	13.28	20.28	87.32	545.2	0.1041	0.1436	0.09847	0.06158	0.1974	0.06782	0.3704	0.8249	2.427	31.33	0.00507	0.02147	0.02185	0.00956	0.01719	0.00332	17.38	28	113.1	907.2	0.153	0.3724	0.3664	0.1492	0.3739	0.1027
46	8.6E+07	M	13.17	21.81	85.42	531.5	0.09714	0.1047	0.08259	0.05252	0.1746	0.06177	0.1938	0.6123	1.334	14.49	0.00335	0.01384	0.01452	0.00685	0.01113	0.00172	16.23	29.89	105.5	740.7	0.1503	0.3904	0.3728	0.1607	0.3693	0.09618
47	857010	M	18.65	17.6	123.7	1076	0.1099	0.1686	0.1974	0.1009	0.1907	0.06049	0.6289	0.6633	4.293	71.56	0.00629	0.03994	0.05554	0.01695	0.02428	0.00354	22.82	21.32	150.6	1567	0.1679	0.509	0.7345	0.2378	0.3799	0.09185
48	8.6E+07	B	8.196	16.84	51.71	201.9	0.086	0.05943	0.01588	0.00592	0.1769	0.06503	0.1563	0.9567	1.094	8.205	0.00897	0.01646	0.01588	0.00592	0.02574	0.00258	8.964	21.96	57.26	242.2	0.1297	0.1357	0.0688	0.02564	0.3105	0.07409
49	85715	M	13.17	18.66	85.98	534.6	0.1158	0.1231	0.1226	0.0734	0.2128	0.06777	0.2871	0.8937	1.897	24.25	0.00653	0.02336	0.02905	0.01215	0.01743	0.00364	15.67	27.95	102.8	759.4	0.1786	0.4166	0.5006	0.2088	0.39	0.1179
50	857155	B	12.05	14.63	78.04	449.3	0.1031	0.09092	0.06592	0.02749	0.1675	0.06043	0.2636	0.7294	1.848	19.87	0.00549	0.01427	0.02322	0.00566	0.01428	0.00242	13.76	20.7	89.88	582.6	0.1494	0.2156	0.305	0.06548	0.2747	0.08301
51	857156	B	13.49	22.3	86.91	561	0.08752	0.07698	0.04751	0.03384	0.1809	0.05718	0.2338	1.353	1.735	20.2	0.00446	0.01382	0.02095	0.01184	0.01641	0.00196	15.15	31.82	99	698.8	0.1162	0.1711	0.2282	0.1282	0.2871	0.06917
52	857343	B	11.76	21.6	74.72	427.9	0.08637	0.04966	0.01657	0.01115	0.1495	0.05888	0.4062	1.21	2.635	28.47	0.00586	0.00976	0.01168	0.00745	0.02406	0.00177	12.98	25.72	82.98	516.5	0.1085	0.08615	0.05523	0.03715	0.2433	0.06563
53	857373	B	13.64	16.34	87.21	571.8	0.07685	0.06059	0.01857	0.01723	0.1353	0.05953	0.1872	0.9234	1.449	14.55	0.00448	0.01177	0.01079	0.00796	0.01325	0.00255	14.67	23.19	96.08	656.7	0.1089	0.1582	0.105	0.08586	0.2346	0.08025
54	857374	B	11.94	18.24	75.71	437.6	0.08261	0.04751	0.01972	0.01349	0.1868	0.0611	0.2273	0.6329	1.52	17.47	0.00721	0.00838	0.01311	0.008	0.01996	0.00264	13.1	21.33	83.67	527.2	0.1144	0.08906	0.09203	0.06296	0.2785	0.07408
55	857392	M	18.22	18.7	120.3	1033	0.1148	0.1485	0.1772	0.106	0.2092	0.0631	0.8337	1.593	4.877	98.81	0.0039	0.02961	0.02817	0.00922	0.02674	0.00513	20.6	24.13	135.1	1321	0.128	0.2297	0.2623	0.1325	0.3021	0.07987
56	857438	M	15.1	22.02	97.26	712.8	0.09056	0.07081	0.05253	0.03334	0.1616	0.05684	0.3105	0.8339	2.097	29.91	0.00468	0.0103	0.01603	0.00922	0.01095	0.00163	18.1	31.69	117.7	1030	0.1389	0.2057	0.2712	0.153	0.2675	0.07873
57	8.6E+07	B	11.52	18.75	73.34	409	0.09524	0.05473	0.03036	0.02278	0.192	0.05907	0.3249	0.9591	2.183	23.47	0.00833	0.00872	0.01349	0.00867	0.03218	0.00239	12.84	22.47	81.81	506.2	0.1249	0.0872	0.09076	0.06316	0.3306	0.07036
58	857637	M	19.21	18.57	125.5	1152	0.1053	0.1267	0.1323	0.08994	0.1917	0.05961	0.7275	1.193	4.837	102.5	0.00646	0.02306	0.02945	0.01538	0.01852	0.00261	26.14	28.14	170.1	2145	0.1624	0.3511	0.3879	0.2091	0.3537	0.08294
59	857793	M	14.71	21.59	95.55	656.9	0.1137	0.1365	0.1293	0.08123	0.2027	0.06758	0.4226	1.15	2.735	40.09	0.00366	0.02855	0.02572	0.01272	0.01817	0.00411	17.87	30.7	115.7	985.5	0.1368	0.429	0.3587	0.1834	0.3698	0.1094
60	857810	B	13.05	19.31	82.61	527.2	0.0806	0.03789	0.00069	0.00417	0.1819	0.05501	0.404	1.214	2.595	32.96	0.00749	0.00859	0.00069	0.00417	0.0219	0.00299	14.23	22.25	90.24	624.1	0.1021	0.06191	0.00185	0.01111	0.2439	0.06289
61	858477	B	8.618	11.79	54.34	224.5	0.09752	0.05272	0.02061	0.0078	0.1683	0.07187	0.1559	0.5796	1.046	8.322	0.01011	0.01055	0.01981	0.00574	0.0209	0.00279	9.507	15.4	59.9	274.9	0.1733	0.1239	0.1168	0.04419	0.322	0.09026
62	858970	B	10.17	14.88	64.55	311.9	0.1134	0.08061	0.01084	0.0129	0.2743	0.0696	0.5158	1.441	3.312	34.62	0.00751	0.01099	0.00767	0.00819	0.04183	0.00595	11.02	17.45	69.86	368.6	0.1275	0.09866	0.02168	0.02579	0.3557	0.0802
63	858981	B	8.598	20.98	54.66	221.8	0.1243	0.08963	0.03	0.00926	0.1828	0.06757	0.3582	2.067	2.493	18.39	0.01193	0.03162	0.03	0.00926	0.03357	0.00305	9.565	27.04	62.06	273.9	0.1639	0.1698	0.09001	0.02778	0.2972	0.07712
64	858986	M	14.25	22.15	96.42	645.7	0.1049	0.2008	0.2135	0.08653	0.1949	0.07292	0.7036	1.268	5.373	60.78	0.00941	0.07056	0.06899	0.01848	0.017	0.00611	17.67	29.51	119.1	959.5	0.164	0.6247	0.6922	0.1785	0.2844	0.1132
65	859196	B	9.173	13.86	59.2	260.9	0.07721	0.08751	0.05988	0.0218	0.2341	0.06963	0.4098	2.265	2.608	23.52	0.00874	0.03938	0.04312	0.0156	0.04192	0.00582	10.01	19.23	65.59	310.1	0.09836	0.1678	0.1397	0.05087	0.3282	0.0849
66	8.6E+07	M	12.68	23.84	82.69	499	0.1122	0.1262	0.1128	0.06873	0.1905	0.0659	0.4255	1.178	2.927	36.46	0.00778	0.02648	0.02973	0.0129	0.01635	0.0036	17.09	33.47	111.8	888.3	0.1851	0.4061	0.4024	0.1716	0.3383	0.1031
67	859283	M	14.78	23.94	97.4	668.3	0.1172	0.1479	0.1267	0.09029	0.1953	0.06654	0.3577	1.281	2.45	35.24	0.0067	0.0231	0.02315	0.01184	0.019	0.00322	17.31	33.39	114.6	925.1	0.1648	0.3416	0.3024	0.1614	0.3321	0.08911
68	859464	B	9.465	21.01	60.11	269.4	0.1044	0.07773	0.02172	0.01504	0.1717	0.06899	0.2351	2.011	1.66	14.2	0.01052	0.01755	0.01714	0.00933	0.02279	0.00424	10.41	31.56	67.03	330.7	0.1548	0.1664	0.09412	0.06517	0.2878	0.09211
69	859465	B	11.31	19.04	71.8	394.1	0.08139	0.04701	0.03709	0.0223	0.1516	0.05667	0.2727	0.9429	1.831	18.15	0.00928	0.00922	0.02063	0.00897	0.02183	0.00215	12.33	23.84	78	466.7	0.129	0.09148	0.1444	0.06961	0.24	0.06641
70	859471	B	9.029	17.33	58.79	250.5	0.1066	0.1413	0.313	0.04375	0.2111	0.08046	0.3274	1.194	1.885	17.67	0.00955	0.08606	0.3038	0.03322	0.04197	0.00956	10.31	22.65	65.5	324.7	0.1482	0.4365	1.252	0.175	0.4228	0.1175
71	859487	B	12.78	16.49	81.37	502.5	0.09831	0.05234	0.03653	0.02864	0.159	0.05653	0.2368	0.8732	1.471	18.33	0.00796	0.00561	0.01585	0.00866	0.02254	0.00191	13.46	19.76	85.67	554.9	0.1296	0.07061	0.1039	0.05882	0.2383	0.0641
72	859575	M	18.94	21.31	123.6	1130	0.09009	0.1029	0.108	0.07951	0.1582	0.05461	0.7888	0.7975	5.486	96.05	0.00444	0.01652	0.02269	0.0137	0.01386	0.0017	24.86	26.58	165.9	1866	0.1193	0.2336	0.2687	0.1789	0.2551	0.06589
73	859711	B	8.888	14.64	58.79	244	0.09783	0.1531	0.08606	0.02872	0.1902	0.0898	0.5262	0.8522	3.168	25.44	0.01721	0.09368	0.05671	0.01766	0.02541	0.02193	9.733	15.67	62.56	284.4	0.1207	0.2436	0.1434	0.04786	0.2254	0.1084
74	859717	M	17.2	24.52	114.2	929.4	0.1071	0.183	0.1692	0.07944	0.1927	0.06487	0.5907	1.041	3.705	69.47	0.00582	0.05616	0.04252	0.01127	0.01527	0.0063	23.32	33.82	151.6	1681	0.1585	0.7394	0.6566	0.1899	0.3313	0.1339
75	859983	M	13.8	15.79	90.43	584.1	0.1007	0.128	0.07789	0.05069	0.1662	0.06566	0.2787	0.6205	1.957	23.35	0.00472	0.02065	0.01759	0.00921	0.0122	0.00313	16.57	20.86	110.3	812.4	0.1411	0.3542	0.2779	0.1383	0.2589	0.103
76	8610175	B	12.31	16.52	79.19	470.9	0.09172	0.06829	0.03372	0.02272	0.172	0.05914	0.2505	1.025	1.74	19.68	0.00485	0.01819	0.01826	0.00797	0.01386	0.0023	14.11	23.21	89.71	611.1	0.1176	0.1843	0.1703	0.0866	0.2618	0.07609
77	8610404	M	16.07	19.65	104.1	817.7	0.09168	0.08424	0.09769	0.06638	0.1798	0.05391	0.7474	1.016	5.029	79.25	0.01082	0.02203	0.035	0.01809	0.0155	0.00195	19.77	24.56	128.8	1223	0.15	0.2045	0.2829	0.152	0.265	0.06387
78	8610629	B	13.53	10.94	87.91	559.2	0.1291	0.1047	0.06877	0.06556	0.2403	0.06641	0.4101	1.014	2.652	32.65	0.0134	0.02839	0.01162	0.00824	0.02572	0.00616	14.08	12.49	91.36	605.5	0.1451	0.1379	0.08539	0.07407	0.271	0.07191
79	8610637	M	18.05	16.15	120.2	1006	0.1065	0.2146	0.1684	0.108	0.2152	0.06673	0.9806	0.5505	6.311	134.8	0.00794	0.05839	0.04658	0.0207	0.02591	0.00705	22.39	18.91	150.1	1610	0.1478	0.5634	0.3786	0.2102	0.3751	0.1108
80	8610862	M	20.18	23.97	143.7	1245	0.1286	0.3454	0.3754	0.1604	0.2906	0.08142	0.9317	1.885	8.649	116.4	0.01038	0.06835	0.1091	0.02593	0.07895	0.00599	23.37	31.72	170.3	1623	0.1639	0.6164	0.7681	0.2508	0.544	0.09964
81	8610908	B	12.86	18	83.19	506.3	0.09934	0.09546	0.03889	0.02315	0.1718	0.05997	0.2655	1.095	1.778	20.35	0.00529	0.01661	0.02071	0.00818	0.01748	0.00285	14.24	24.82	91.88	622.1	0.1289	0.2141	0.1731	0.07926	0.2779	0.07918
82	861103	B	11.45	20.97	73.81	401.5	0.1102	0.09362	0.04591	0.02233	0.1842	0.07005	0.3251	2.174	2.077	24.62	0.01037	0.01706	0.02586	0.00751	0.01816	0.00398	13.11	32.16	84.53	525.1	0.1557	0.1676	0.1755	0.06127	0.2762	0.08851
83	8611161	B	13.34	15.86	86.49	520	0.1078	0.1535	0.1169	0.06987	0.1942	0.06902	0.286	1.016	1.535	12.96	0.00679	0.03575	0.0398	0.01383	0.02134	0.0046	15.53	23.19	96.66	614.9	0.1536	0.4791	0.4858	0.1708	0.3527	0.1016
84	8611555	M	25.22	24.91	171.5	1878	0.1063	0.2665	0.3339	0.1845	0.1829	0.06782	0.8973	1.474	7.382	120	0.00817	0.05693	0.0573	0.0203	0.01065	0.00589	30	33.62	211.7	2562	0.1573	0.6076	0.6476	0.2867	0.2355	0.1051
85	8611792	M	19.1	26.29	129.1	1132	0.1215	0.1791	0.1937	0.1469	0.1634	0.07224	0.519	2.91	5.801	67.1	0.00755	0.0605	0.02134	0.01843	0.03056	0.01039	20.33	32.72	141.3	1298	0.1392	0.2817	0.2432	0.1841	0.2311	0.09203
86	8612080	B	12	15.65	76.95	443.3	0.09723	0.07165	0.04151	0.01863	0.2079	0.05968	0.2271	1.255	1.441	16.16	0.00597	0.01812	0.02007	0.00703	0.01972	0.00261	13.67	24.9	87.78	567.9	0.1377	0.2003	0.2267	0.07632	0.3379	0.07924
87	8612399	M	18.46	18.52	121.1	1075	0.09874	0.1053	0.1335	0.08795	0.2132	0.06022	0.6997	1.475	4.782	80.6	0.00647	0.01649	0.02806	0.0142	0.0237	0.00376	22.93	27.68	152.2	1603	0.1398	0.2089	0.3157	0.1642	0.3695	0.08579
88	8.6E+07	M	14.48	21.46	94.25	648.2	0.09444	0.09947	0.1204	0.04938	0.2075	0.05636	0.4204	2.22	3.301	38.87	0.00937	0.02983	0.05371	0.01761	0.02418	0.00325	16.21	29.25	108.4	808.9	0.1306	0.1976	0.3349	0.1225	0.302	0.06846
89	8.6E+07	M	19.02	24.59	122	1076	0.09029	0.1206	0.1468	0.08271	0.1953	0.05629	0.5495	0.6636	3.055	57.65	0.00387	0.01842	0.0371	0.012	0.01964	0.00334	24.56	30.41	152.9	1623	0.1249	0.3206	0.5755	0.1956	0.3956	0.09288
90	861597	B	12.36	21.8	79.78	466.1	0.08772	0.09445	0.06015	0.03745	0.193	0.06404	0.2978	1.502	2.203	20.95	0.00711	0.02493	0.02703	0.01293	0.01958	0.00446	13.83	30.5	91.46	574.7	0.1304	0.2463	0.2434	0.1205	0.2972	0.09261
91	861598	B	14.64	15.24	95.77	651.9	0.1132	0.1339	0.09966	0.07064	0.2116	0.06346	0.5115	0.7372	3.814	42.76	0.00551	0.04412	0.04436	0.01623	0.02427	0.00484	16.34	18.24	109.4	803.6	0.1277	0.3089	0.2604	0.1397	0.3151	0.08473
92	861648	B	14.62	24.02	94.57	662.7	0.08974	0.08606	0.03102	0.02957	0.1685	0.05866	0.3721	1.111	2.279	33.76	0.00487	0.01818	0.01121	0.00861	0.02085	0.00289	16.11	29.11	102.9	803.7	0.1115	0.1766	0.09189	0.06946	0.2522	0.07246
93	861799	M	15.37	22.76	100.2	728.2	0.092	0.1036	0.1122	0.07483	0.1717	0.06097	0.3129	0.8413	2.075	29.44	0.00988	0.02444	0.04531	0.01763	0.02471	0.00214	16.43	25.84	107.5	830.9	0.1257	0.1997	0.2846	0.1476	0.2556	0.06828
94	861853	B	13.27	14.76	84.74	551.7	0.07355	0.05055	0.03261	0.02648	0.1386	0.05318	0.4057	1.153	2.701	36.35	0.00448	0.01038	0.01358	0.01082	0.01069	0.00144	16.36	22.35	104.5	830.6	0.1006	0.1238	0.135	0.1001	0.2027	0.06206
95	862009	B	13.45	18.3	86.6	555.1	0.1022	0.08165	0.03974	0.0278	0.1638	0.0571	0.295	1.373	2.099	25.22	0.00588	0.01491	0.01872	0.00937	0.01884	0.00182	15.1	25.94	97.59	699.4	0.1339	0.1751	0.1381	0.07911	0.2678	0.06603
96	862028	M	15.06	19.83	100.3	705.6	0.1039	0.1553	0.17	0.08815	0.1855	0.06284	0.4768	0.9644	3.706	47.14	0.00925	0.03715	0.04867	0.01851	0.01498	0.00352	18.23	24.23	123.5	1025	0.1551	0.4203	0.5203	0.2115	0.2834	0.08234
97	86208	M	20.26	23.03	132.4	1264	0.09078	0.1313	0.1465	0.08683	0.2095	0.05649	0.7576	1.509	4.554	87.87	0.00602	0.03482	0.04232	0.01269	0.02657	0.00441	24.22	31.59	156.1	1750	0.119	0.3539	0.4098	0.1573	0.3689	0.08368
98	86211	B	12.18	17.84	77.79	451.1	0.1045	0.07057	0.0249	0.02941	0.19	0.06635	0.3661	1.511	2.41	24.44	0.00543	0.01179	0.01131	0.01519	0.0222	0.00341	12.83	20.92	82.14	495.2	0.114	0.09358	0.0498	0.05882	0.2227	0.07376
99	862261	B	9.787	19.94	62.11	294.5	0.1024	0.05301	0.00683	0.00794	0.135	0.0689	0.335	2.043	2.132	20.05	0.01113	0.01463	0.00531	0.00525	0.01801	0.00567	10.92	26.29	68.81	366.1	0.1316	0.09473	0.02049	0.02381	0.1934	0.08988
100	862485	B	11.6	12.84	74.34	412.6	0.08983	0.07525	0.04196	0.0335	0.162	0.06582	0.2315	0.5391	1.475	15.75	0.00615	0.0133	0.01693	0.00688	0.01651	0.00255	13.06	17.16	82.96	512.5	0.1431	0.1851	0.1922	0.08449	0.2772	0.08756
101	862548	M	14.42	19.77	94.48	642.5	0.09752	0.1141	0.09388	0.05839	0.1879	0.0639	0.2895	1.851	2.376	26.85	0.00801	0.02895	0.03321	0.01424	0.01462	0.00445	16.33	30.86	109.5	826.4	0.1431	0.3026	0.3194	0.1565	0.2718	0.09353
102	862717	M	13.61	24.98	88.05	582.7	0.09488	0.08511	0.08625	0.04489	0.1609	0.05871	0.4565	1.29	2.861	43.14	0.00587	0.01488	0.02647	0.00992	0.01465	0.00236	16.99	35.27	108.6	906.5	0.1265	0.1943	0.3169	0.1184	0.2651	0.07397
103	862722	B	6.981	13.43	43.79	143.5	0.117	0.07568	0	0	0.193	0.07818	0.2241	1.508	1.553	9.833	0.01019	0.01084	0	0	0.02659	0.0041	7.93	19.54	50.41	185.2	0.1584	0.1202	0	0	0.2932	0.09382
104	862965	B	12.18	20.52	77.22	458.7	0.08013	0.04038	0.02383	0.0177	0.1739	0.05677	0.1924	1.571	1.183	14.68	0.00508	0.0061	0.01069	0.0068	0.01447	0.00153	13.34	32.84	84.58	547.8	0.1123	0.08862	0.1145	0.07431	0.2694	0.06878
105	862980	B	9.876	19.4	63.95	298.3	0.1005	0.09697	0.06154	0.03029	0.1945	0.06322	0.1803	1.222	1.528	11.77	0.00906	0.02196	0.03029	0.01112	0.01609	0.00357	10.76	26.83	72.22	361.2	0.1559	0.2302	0.2644	0.09749	0.2622	0.0849
106	862989	B	10.49	19.29	67.41	336.1	0.09989	0.08578	0.02995	0.01201	0.2217	0.06481	0.355	1.534	2.302	23.13	0.0076	0.02219	0.0288	0.00861	0.0271	0.00345	11.54	23.31	74.22	402.8	0.1219	0.1486	0.07987	0.03203	0.2826	0.07552
107	863030	M	13.11	15.56	87.21	530.2	0.1398	0.1765	0.2071	0.09601	0.1925	0.07692	0.3908	0.9238	2.41	34.66	0.00716	0.02912	0.05473	0.01388	0.01547	0.0071	16.31	22.4	106.4	827.2	0.1862	0.4099	0.6376	0.1986	0.3147	0.1405
108	863031	B	11.64	18.33	75.17	412.5	0.1142	0.1017	0.0707	0.03485	0.1801	0.0652	0.306	1.657	2.155	20.62	0.00854	0.0231	0.02945	0.01398	0.01565	0.00384	13.14	29.26	85.51	521.7	0.1688	0.266	0.2873	0.1218	0.2806	0.09097
109	863270	B	12.36	18.54	79.01	466.7	0.08477	0.06815	0.02643	0.01921	0.1602	0.06066	0.1199	0.8944	0.8484	9.227	0.00346	0.01047	0.01167	0.00556	0.01251	0.00136	13.29	27.49	85.56	544.1	0.1184	0.1963	0.1937	0.08442	0.2983	0.07185
110	86355	M	22.27	19.67	152.8	1509	0.1326	0.2768	0.4264	0.1823	0.2556	0.07039	1.215	1.545	10.05	170	0.00652	0.08668	0.104	0.0248	0.03112	0.00504	28.4	28.01	206.8	2360	0.1701	0.6997	0.9608	0.291	0.4055	0.09789
111	864018	B	11.34	21.26	72.48	396.5	0.08759	0.06575	0.05133	0.01899	0.1487	0.06529	0.2344	0.9861	1.597	16.41	0.00911	0.01557	0.02443	0.00644	0.01568	0.00248	13.01	29.15	83.99	518.1	0.1699	0.2196	0.312	0.08278	0.2829	0.08832
112	864033	B	9.777	16.99	62.5	290.2	0.1037	0.08404	0.04334	0.01778	0.1584	0.07065	0.403	1.424	2.747	22.87	0.01385	0.02932	0.02722	0.01023	0.03281	0.00464	11.05	21.47	71.68	367	0.1467	0.1765	0.13	0.05334	0.2533	0.08468
113	86408	B	12.63	20.76	82.15	480.4	0.09933	0.1209	0.1065	0.06021	0.1735	0.0707	0.3424	1.803	2.711	20.48	0.01291	0.04042	0.05101	0.02295	0.02144	0.00589	13.33	25.47	89	527.4	0.1287	0.225	0.2216	0.1105	0.2226	0.08486
114	86409	B	14.26	19.65	97.83	629.9	0.07837	0.2233	0.3003	0.07798	0.1704	0.07769	0.3628	1.49	3.399	29.25	0.0053	0.07446	0.1435	0.02292	0.02566	0.01298	15.3	23.73	107	709	0.08949	0.4193	0.6783	0.1505	0.2398	0.1082
115	864292	B	10.51	20.19	68.64	334.2	0.1122	0.1303	0.06476	0.03068	0.1922	0.07782	0.3336	1.86	2.041	19.91	0.01188	0.03747	0.04591	0.01544	0.02287	0.00679	11.16	22.75	72.62	374.4	0.13	0.2049	0.1295	0.06136	0.2383	0.09026
116	864496	B	8.726	15.83	55.84	230.9	0.115	0.08201	0.04132	0.01924	0.1649	0.07633	0.1665	0.5864	1.354	8.966	0.00826	0.02213	0.03259	0.0104	0.01708	0.00381	9.628	19.62	64.48	284.4	0.1724	0.2364	0.2456	0.105	0.2926	0.1017

Chapter 7: Machine Learning Algorithms

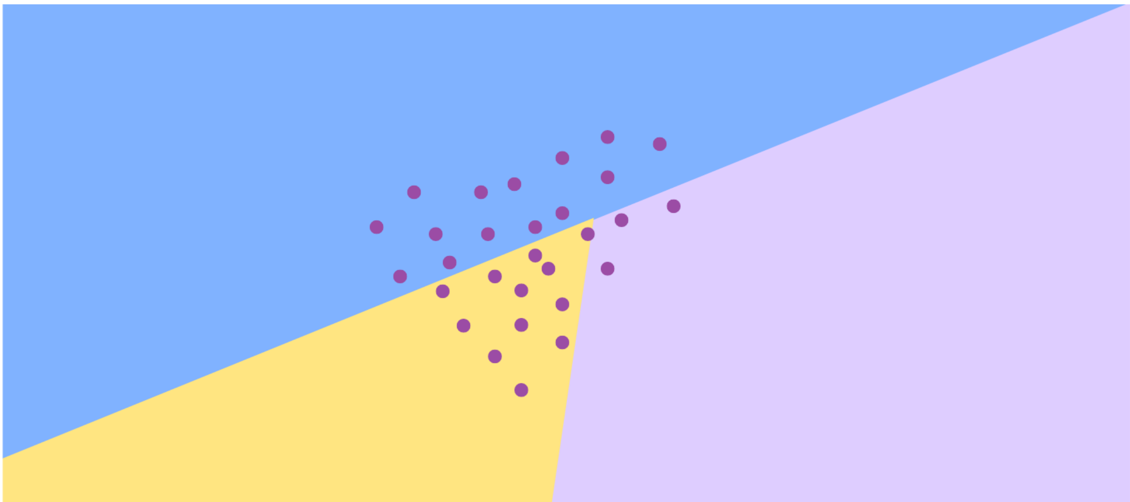
7.1 Random Forest Classifier



Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

7.2 SVC



The most applicable machine learning algorithm for our problem is Linear SVC. Before hopping into Linear SVC with our data, we're going to show a very simple example that should help solidify your understanding of working with Linear SVC.

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. This makes this specific algorithm rather suitable for our uses, though you can use this for many situations. Let's get started.

7.3 Selection of model

For random forest, we obtained an accuracy of 98.24%, while for SVC, we obtained an accuracy of 97.36%. As a result, we decided to use random forest as our primary model because it provided us with better accuracy.

Chapter 8: Source Code

8.1 Importing Libraries:

In [1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import seaborn as s
from sklearn import model_selection
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn import svm
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder
#optimum parameter choosing
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
from xgboost import XGBClassifier
import pickle
import os
import warnings
from pandas import MultiIndex, Int64Index
warnings.filterwarnings('ignore')
```

8.2 Importing Dataset:

In [2]:
```
os.chdir ('C:\\Users\\Nilesh\Documents\\Cancer Detection Project')
data = pd.read_csv('data.csv')
data
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...
...	...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...

569 rows × 32 columns

8.3 Creating x variable:

In [9]:
```python
x= df.drop (labels='diagnosis' ,axis =1 )
x
```

Out[9]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mear
0	842302	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419
1	842517	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812
2	84300903	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069
3	84348301	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597
4	84358402	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809
...	...	...	...	...	...	...	...	...	...	..
564	926424	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726
565	926682	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752
566	926954	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590
567	927241	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397
568	92751	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587

569 rows × 31 columns

8.4 Normalizing x variable:

In [13]:
```python
df_norm = (x- x.mean()) / (x.max()- x.min())
df_norm= pd.concat ([df_norm,y], axis =1 )
df_norm
```

Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mear
0	-0.032403	0.182815	-0.301307	0.213053	0.146813	0.198968	0.531437	0.495081	0.487976	0.30675
1	-0.032403	0.304923	-0.051392	0.282848	0.284671	-0.104905	-0.078833	-0.004450	0.105621	0.00019
2	0.059177	0.263274	0.066295	0.262808	0.232497	0.119524	0.170416	0.254453	0.392549	0.12999
3	0.059229	-0.128132	0.036874	-0.099434	-0.114014	0.416536	0.550761	0.357546	0.279726	0.39665
4	0.059241	0.291671	-0.167388	0.298051	0.272369	0.035567	0.087292	0.255859	0.275253	-0.00132
...	...	...	...	...	...	...	...	...	...	..
564	-0.032311	0.351778	0.104848	0.345733	0.349570	0.132163	0.035455	0.363404	0.447221	-0.04324
565	-0.032311	0.284098	0.303022	0.271101	0.257099	0.012997	-0.002886	0.129336	0.243493	-0.03011
566	-0.032310	0.117029	0.297273	0.112853	0.086198	-0.106620	-0.006260	0.008694	0.020382	-0.11192
567	-0.032310	0.306342	0.339545	0.332603	0.258796	0.193552	0.529596	0.615278	0.512330	0.29564
568	-0.033226	-0.301353	0.177557	-0.304395	-0.201013	-0.394785	-0.186249	-0.208058	-0.243137	-0.11344

569 rows × 32 columns

8.5 Creating y variable:

In [10]:
```python
y = df['diagnosis']
y
```

Out[10]:
```
0      M
1      M
2      M
3      M
4      M
      ..
564    M
565    M
566    M
567    M
568    B
Name: diagnosis, Length: 569, dtype: category
Categories (2, object): ['B', 'M']
```

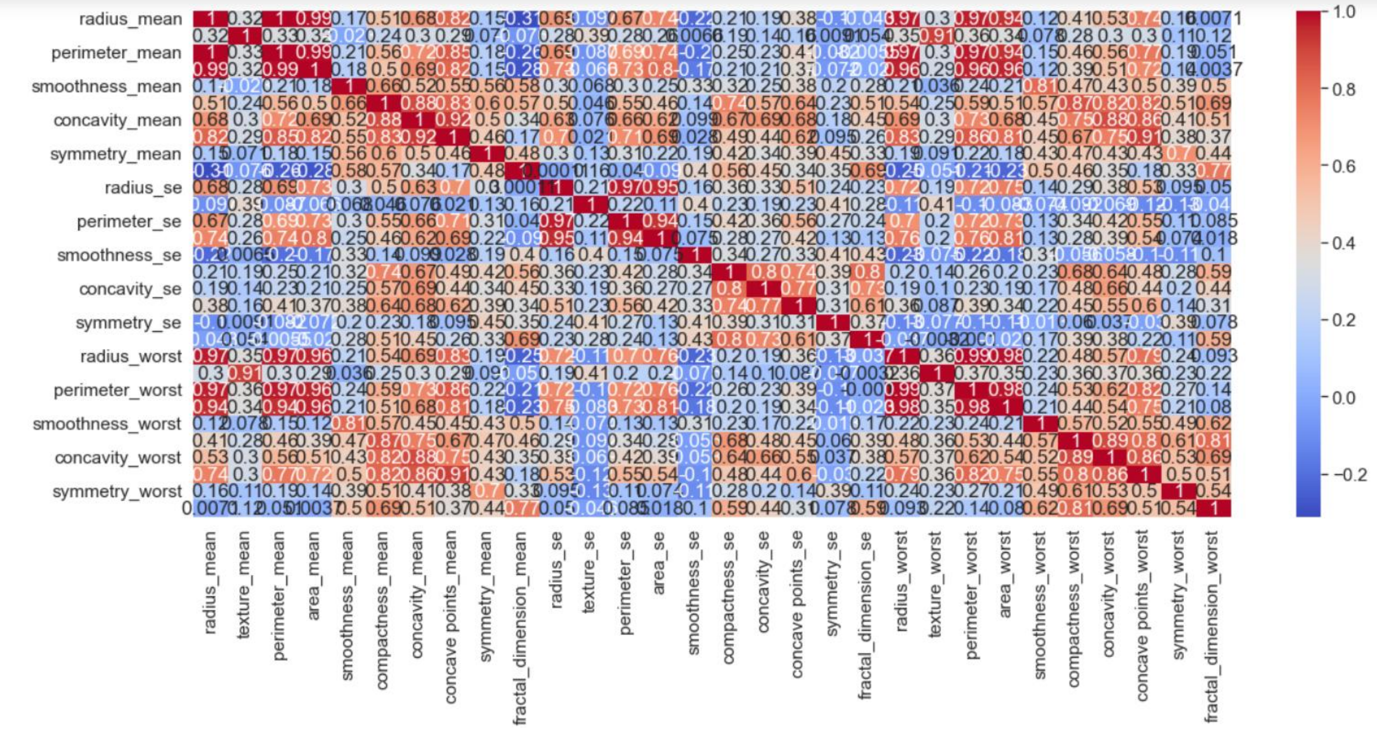
8.6 Normalizing y variable:

```
In [21]: le = LabelEncoder()
         le.fit (y_norm)
         y_norm = le.transform(y_norm)
         y_norm = pd.DataFrame(y_norm)
         print (y_norm)
```

```
     0
0    1
1    1
2    1
3    1
4    1
..  ..
564  1
565  1
566  1
567  1
568  0

[569 rows x 1 columns]
```

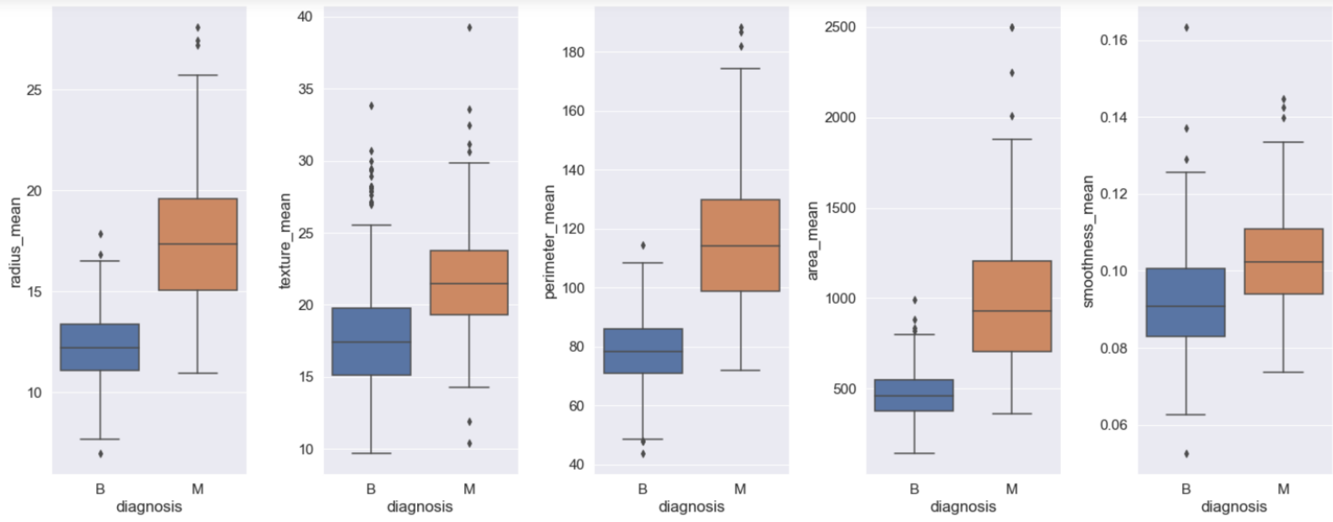
8.7 Heatmap:

```
In [15]: plt.rcParams['figure.figsize']=(20,8)
         s.set(font_scale=1.4)
         # In co relation 1 is the highest and -1 is lowest
         s.heatmap (df.drop('diagnosis',axis =1).drop('id',axis =1).corr(),cmap = 'coolwarm',annot = True)
```



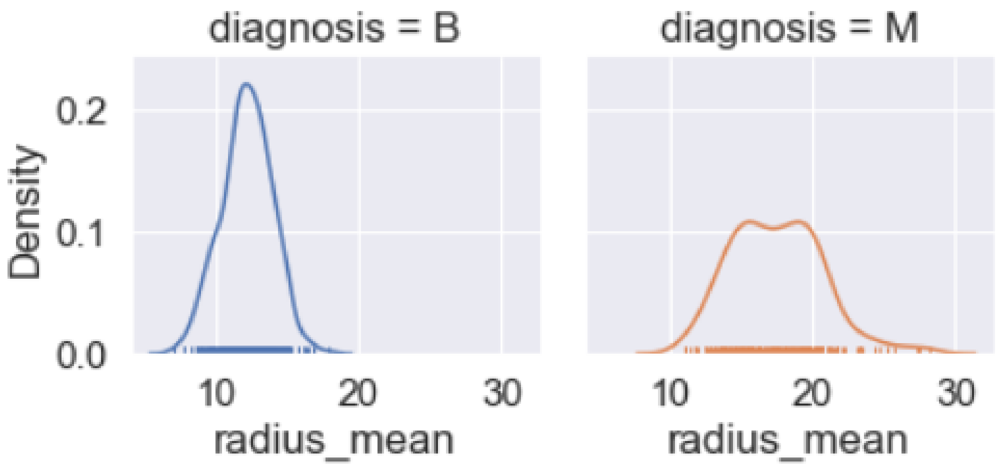
8.8 Boxplot:

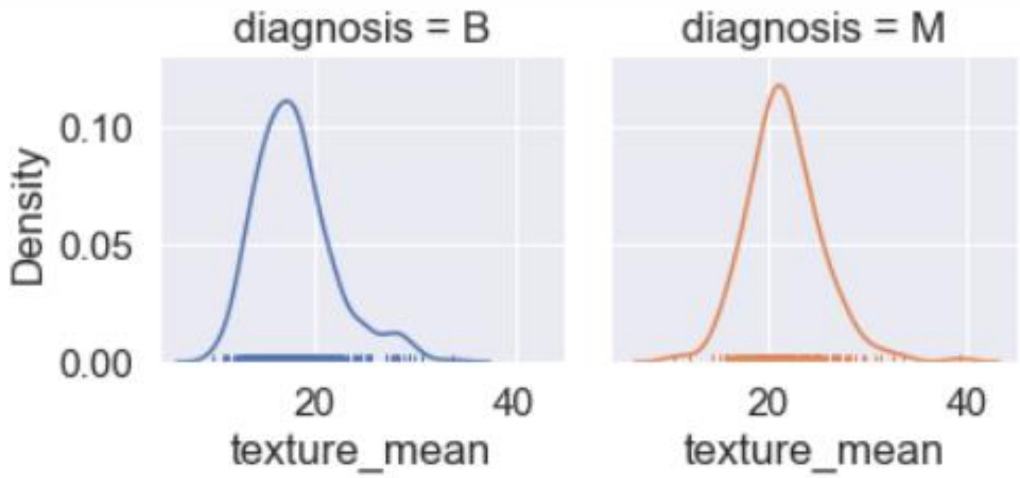
In [16]:
```
plt.rcParams['figure.figsize']=(20,8)
f, (ax1,ax2,ax3,ax4,ax5) = plt.subplots (1,5)
s.boxplot ('diagnosis', y = 'radius_mean',data = df , ax = ax1)
s.boxplot ('diagnosis', y = 'texture_mean',data = df , ax = ax2)
s.boxplot ('diagnosis', y = 'perimeter_mean',data = df , ax = ax3)
s.boxplot ('diagnosis', y = 'area_mean',data = df , ax = ax4)
s.boxplot ('diagnosis', y = 'smoothness_mean',data = df , ax = ax5)
f .tight_layout()

f, (ax1,ax2,ax3,ax4,ax5) = plt.subplots (1,5)
s.boxplot ('diagnosis', y = 'compactness_mean',data = df , ax = ax1)
s.boxplot ('diagnosis', y = 'concavity_mean',data = df , ax = ax2)
s.boxplot ('diagnosis', y = 'concave points_mean',data = df , ax = ax3)
s.boxplot ('diagnosis', y = 'symmetry_mean',data = df , ax = ax4)
s.boxplot ('diagnosis', y = 'fractal_dimension_mean',data = df , ax = ax5)
f .tight_layout()
```

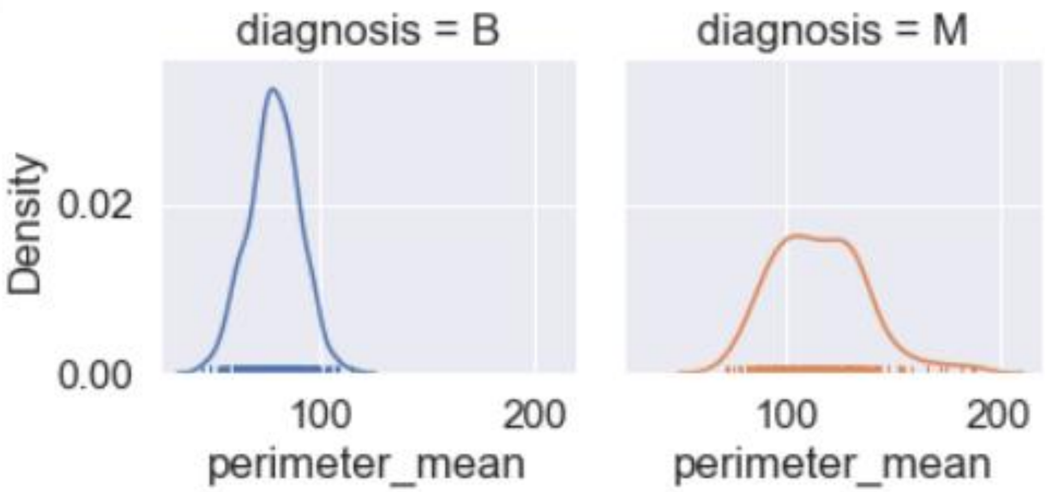


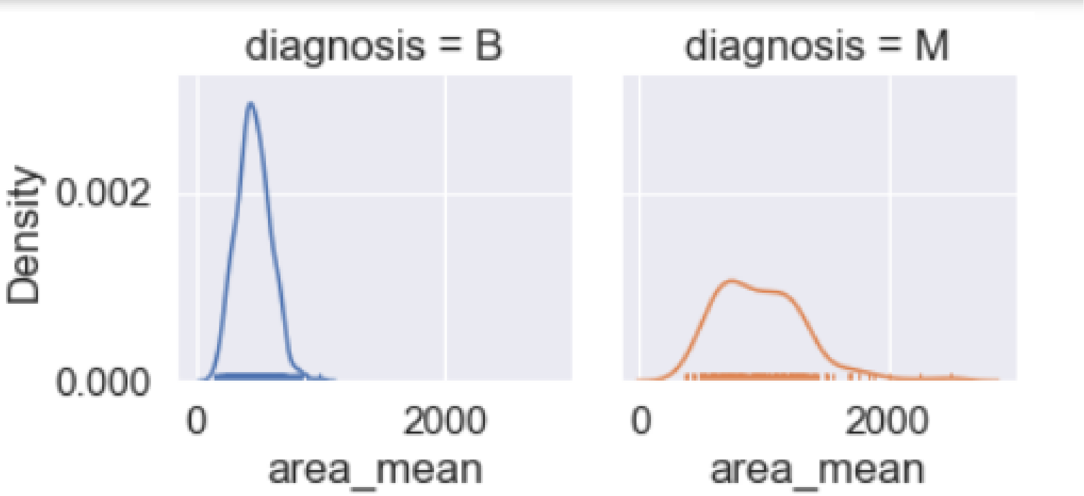
8.9 Distplot:

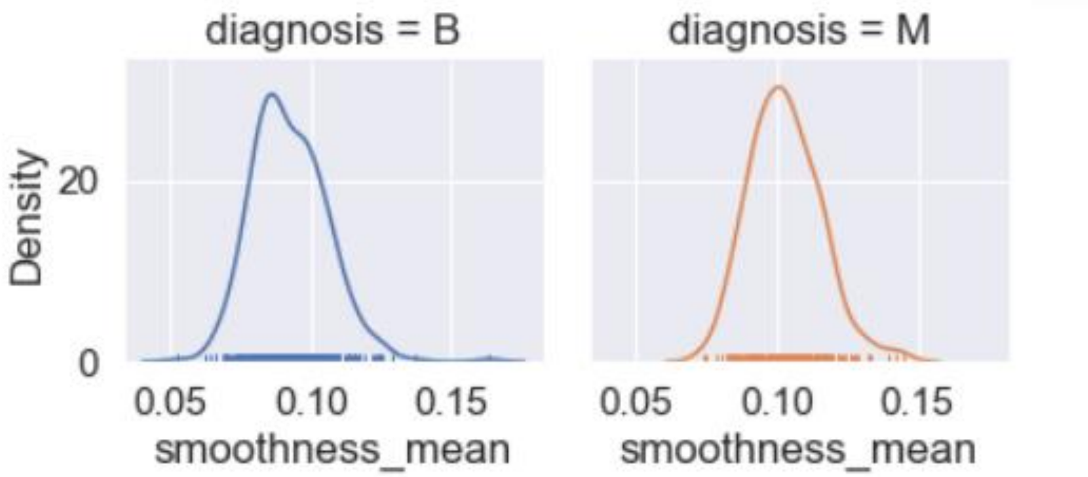
In [17]:
```
g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "radius_mean", hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, 'texture_mean', hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, 'perimeter_mean', hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "area_mean", hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "smoothness_mean", hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "compactness_mean", hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "concavity_mean", hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "concave points_mean", hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "symmetry_mean", hist = False, rug = True)

g = s.FacetGrid (df,col = 'diagnosis', hue = 'diagnosis')
g.map (s.distplot, "fractal_dimension_mean", hist = False, rug = True)
```

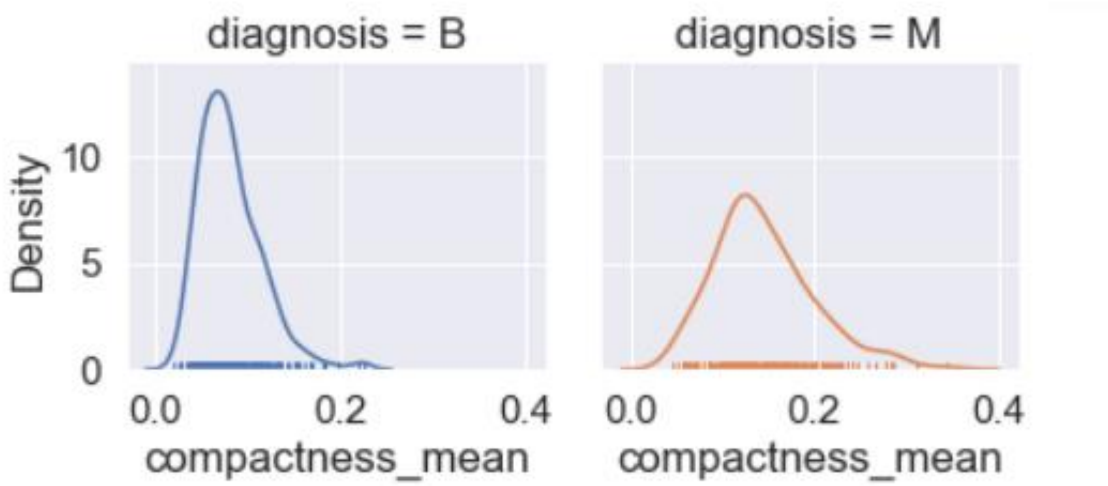

diagnosis = B
diagnosis = M
Density
0.2
0.1
0.0
10
20
30
radius_mean
radius_mean

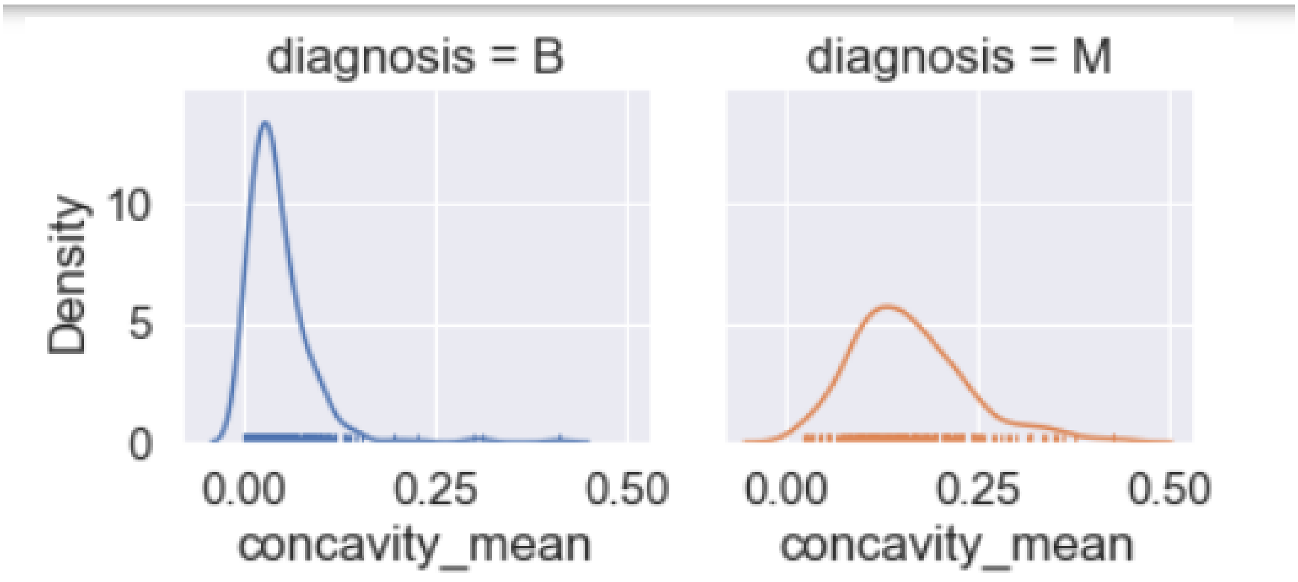

diagnosis = B
diagnosis = M
Density
0.10
0.05
0.00
20
40
texture_mean
texture_mean

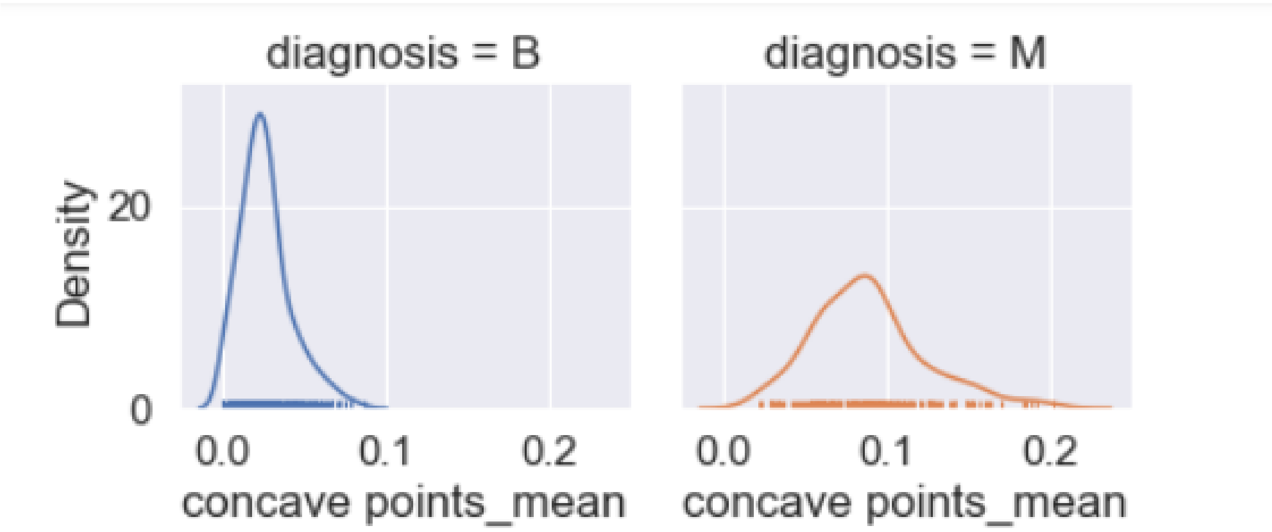

diagnosis = B
diagnosis = M
Density
0.02
0.00
100
200
perimeter_mean
perimeter_mean

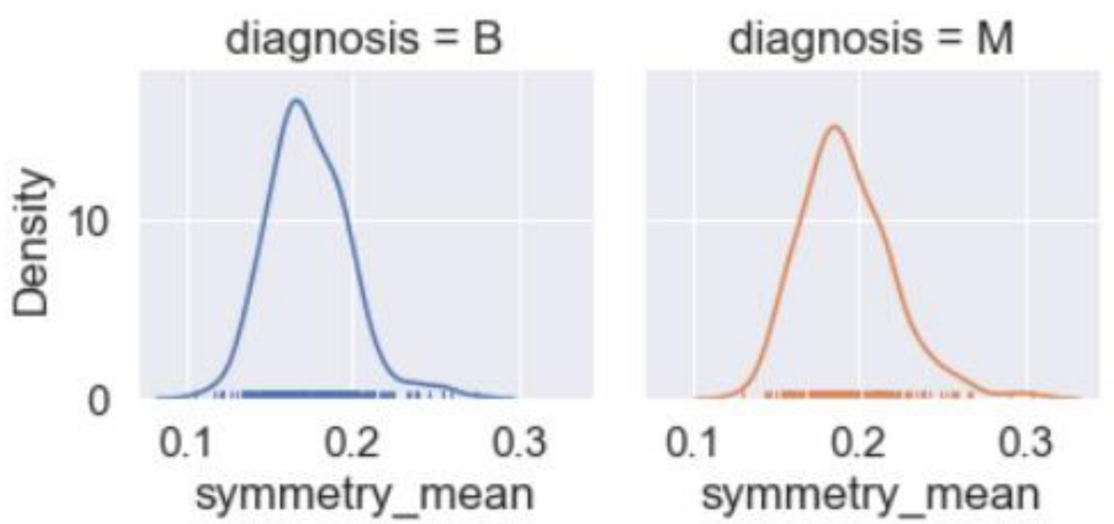

diagnosis = B
diagnosis = M
Density
0.002
0.000
0
2000
area_mean

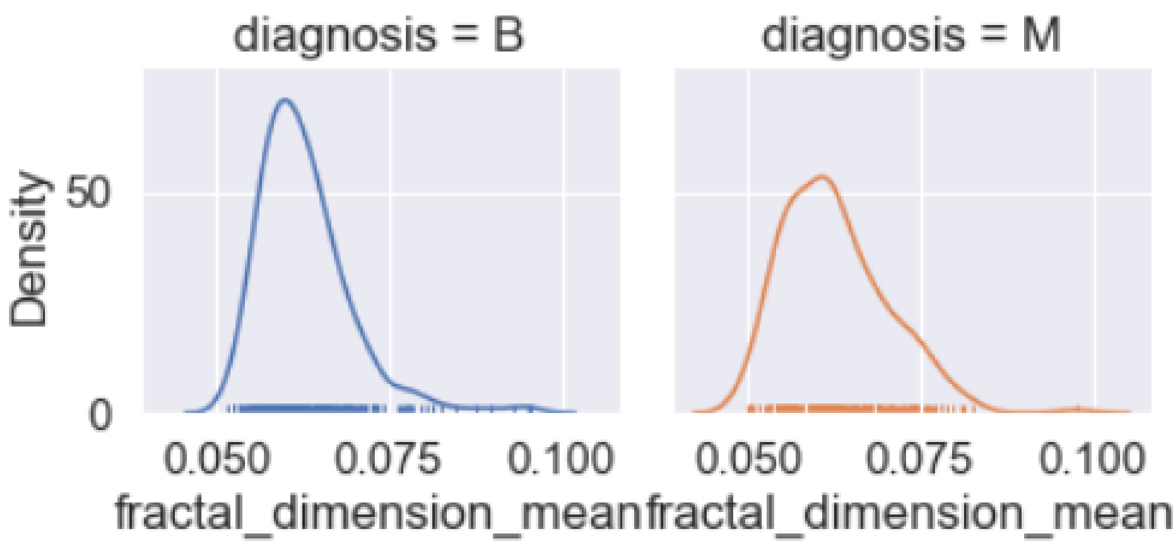

diagnosis = B
diagnosis = M
Density
20
0
0.05
0.10
0.15
smoothness_mean


diagnosis = B
diagnosis = M
Density
10
5
0
0.0
0.2
0.4
compactness_mean


diagnosis = B
diagnosis = M
Density
10
5
0
0.00
0.25
0.50
concavity_mean


diagnosis = B
diagnosis = M
Density
20
0
0.0
0.1
0.2
concave points_mean


diagnosis = B
diagnosis = M
Density
10
0
0.1
0.2
0.3
symmetry_mean



8.10 Creating function for fitting model:

```
In [22]: def FitModel (X,Y, algo_name , algorithm, gridSearchParams, cv):
             np.random.seed(10)
             x_train, x_test, y_train, y_test = train_test_split (X,Y,test_size = 0.2)

             # Find the Parameters , then choose best parameters

             grid = GridSearchCV(estimator = algorithm, param_grid = gridSearchParams,
                                 cv = cv, scoring = 'accuracy', verbose = 1 , n_jobs = -1 )

             grid_result = grid.fit(x_train, y_train)
             best_params = grid_result.best_params_
             pred = grid_result.predict (x_test)
             cm = confusion_matrix (y_test,pred)

             print (pred)
             pickle.dump(grid_result,open(algo_name,'wb'))

             print ('Best Params :', best_params)
             print ('Classification Report:',classification_report(y_test,pred))
             print ('Accuracy Score', (accuracy_score(y_test,pred)))
             print ('Confusion Matrix :\n',cm)
```

8.11 SVC Model:

```
In [30]: param = {
                     'C': [0.1,1,100,1000],
                     'gamma':[0.0001,0.001, 0.005, 0.1,1, 3,5,10, 100]
                 }

         FitModel (x_norm,y_norm,'SVC',SVC(), param, cv =5)
```

```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
[1 0 0 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 1 1 0 0 0
 0 1 1]
Best Params : {'C': 1, 'gamma': 1}
Classification Report:               precision    recall  f1-score   support

           0       1.00      0.96      0.98        75
           1       0.93      1.00      0.96        39

    accuracy                           0.97       114
   macro avg       0.96      0.98      0.97       114
weighted avg       0.98      0.97      0.97       114

Accuracy Score 0.9736842105263158
Confusion Matrix :
 [[72  3]
 [ 0 39]]
```

8.12 Random Forest Model:

```
In [31]: param = { 'n_estimators': [100,500,1000,2000]  }
         FitModel (x_norm,y_norm,'Random Forest',RandomForestClassifier(), param, cv =10)
```

```
Fitting 10 folds for each of 4 candidates, totalling 40 fits
[1 0 0 1 0 0 0 0 1 1 0 0 1 0 1 1 0 0 0 1 1 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 1 1 0 0 0
 0 1 1]
Best Params : {'n_estimators': 100}
Classification Report:               precision    recall  f1-score   support

           0       1.00      0.97      0.99        75
           1       0.95      1.00      0.97        39

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Accuracy Score 0.9824561403508771
Confusion Matrix :
 [[73  2]
 [ 0 39]]
```

Chapter 9: Testing of Model

9.1 Loading pickle file for execution

```
In [2]: import pickle
        import os
        import pandas as pd

        load_model =pickle.load(open("SVC","rb"))

        os.chdir ('C:\\Users\\Nilesh\Documents\\Cancer Detection Project')
        data = pd.read_csv('Reports.csv')

        x= data.drop (labels='Actual diagnosis' ,axis =1 )
        x_norm = (x- x.mean()) / (x.max()- x.min())

        load_model =pickle.load(open("SVC","rb"))

        pred = load_model.predict(x_norm)
        pred=pred.astype(str)

        pred[pred=="1"]="Malignant"
        pred[pred=="0"]="Benign"

        data['Predictions'] = pred
        data.to_csv('Result.csv')
```

9.2 Input

1	id	radius_m	texture_m	perimeter	area_mea	smoothne	compactn	concavity	concave p	symmetry	fractal_di	radius_se	texture_se	perimeter	area_se	smoothne	compactn	concavity	concave p	symmetry	fractal_di	radius_w	texture_w	perimeter	area_wor	smoothne	compactn	concavity	concave p	symmetry	fractal_di	Actual diagnosis
2	859575	18.94	21.31	123.6	1130	0.09009	0.1029	0.108	0.07951	0.1582	0.05461	0.7888	0.7975	5.486	96.05	0.00444	0.01652	0.02269	0.0137	0.01386	0.0017	24.86	26.58	165.9	1866	0.1193	0.2336	0.2687	0.1789	0.2551	0.06589	M
3	8611161	13.34	15.86	86.49	520	0.1078	0.1535	0.1169	0.06987	0.1942	0.06902	0.286	1.016	1.535	12.96	0.00679	0.03575	0.0398	0.01383	0.02134	0.0046	15.53	23.19	96.66	614.9	0.1536	0.4791	0.4858	0.1708	0.3527	0.1016	B
4	859717	17.2	24.52	114.2	929.4	0.1071	0.183	0.1692	0.07944	0.1927	0.06487	0.5907	1.041	3.705	69.47	0.00582	0.05616	0.04252	0.01127	0.01527	0.0063	23.32	33.82	151.6	1681	0.1585	0.7394	0.6566	0.1899	0.3313	0.1339	M
5	8911163	17.93	24.48	115.2	998.9	0.08855	0.07027	0.05699	0.04744	0.1538	0.0551	0.4212	1.433	2.765	45.81	0.00544	0.01169	0.01622	0.00852	0.01419	0.00275	20.92	34.69	135.1	1320	0.1315	0.1806	0.208	0.1136	0.2504	0.07948	M
6	913102	14.64	16.85	94.21	666	0.08641	0.06698	0.05192	0.02791	0.1409	0.05355	0.2204	1.006	1.471	19.98	0.00354	0.01393	0.018	0.00614	0.01254	0.00122	16.46	25.44	106	831	0.1142	0.207	0.2437	0.07828	0.2455	0.06596	B
7	8610908	12.86	18	83.19	506.3	0.09934	0.09546	0.03889	0.02315	0.1718	0.05997	0.2655	1.095	1.778	20.35	0.00529	0.01661	0.02071	0.00818	0.01748	0.00285	14.24	24.82	91.88	622.1	0.1289	0.2141	0.1731	0.07926	0.2779	0.07918	B
8	912193	12.16	18.03	78.29	455.3	0.09087	0.07838	0.02916	0.01527	0.1464	0.06284	0.2194	1.19	1.678	16.26	0.00491	0.01666	0.01397	0.00516	0.01454	0.00186	13.34	27.87	88.83	547.4	0.1208	0.2279	0.162	0.0569	0.2406	0.07729	B
9	8610637	18.05	16.15	120.2	1006	0.1065	0.2146	0.1684	0.108	0.2152	0.06673	0.9806	0.5505	6.311	134.8	0.00794	0.05839	0.04658	0.0207	0.02591	0.00705	22.39	18.91	150.1	1610	0.1478	0.5634	0.3786	0.2102	0.3751	0.1108	M
10	8610862	20.18	23.97	143.7	1245	0.1286	0.3454	0.3754	0.1604	0.2906	0.08142	0.9317	1.885	8.649	116.4	0.01038	0.06835	0.1091	0.02593	0.07895	0.00599	23.37	31.72	170.3	1623	0.1639	0.6164	0.7681	0.2508	0.544	0.09964	M
11	91227	13.9	19.24	88.73	602.9	0.07991	0.05326	0.02995	0.0207	0.1579	0.05594	0.3316	0.9264	2.056	28.41	0.0037	0.01082	0.0153	0.00628	0.01062	0.00222	16.41	26.42	104.4	830.5	0.1064	0.1415	0.1673	0.0815	0.2356	0.07603	B
12	912519	13.47	14.06	87.32	546.3	0.1071	0.1155	0.05786	0.05266	0.1779	0.06639	0.1588	0.5733	1.102	12.84	0.00445	0.01452	0.01334	0.00879	0.01698	0.00279	14.83	18.32	94.94	660.2	0.1393	0.2499	0.1848	0.1335	0.3227	0.09326	B
13	927241	20.6	29.33	140.1	1265	0.1178	0.277	0.3514	0.152	0.2397	0.07016	0.726	1.595	5.772	86.22	0.00652	0.06158	0.07117	0.01664	0.02324	0.00619	25.74	39.42	184.6	1821	0.165	0.8681	0.9387	0.265	0.4087	0.124	M
14	912558	13.7	17.64	87.76	571.1	0.0995	0.07957	0.04548	0.0316	0.1732	0.06088	0.2431	0.9462	1.564	20.64	0.00325	0.00819	0.01698	0.00923	0.01285	0.00152	14.96	23.53	95.78	686.5	0.1199	0.1346	0.1742	0.09077	0.2518	0.0696	B
15	8610404	16.07	19.65	104.1	817.7	0.09168	0.08424	0.09769	0.06638	0.1798	0.05391	0.7474	1.016	5.029	79.25	0.01082	0.02203	0.035	0.01809	0.0155	0.00195	19.77	24.56	128.8	1223	0.15	0.2045	0.2829	0.152	0.265	0.06387	M
16	884180	19.4	23.5	129.1	1155	0.1027	0.1558	0.2049	0.08886	0.1978	0.06	0.5243	1.802	4.037	60.41	0.01061	0.03252	0.03915	0.01559	0.02186	0.00395	21.65	30.53	144.9	1417	0.1463	0.2968	0.3458	0.1564	0.292	0.07614	M
17																																

9.3 Output

1	id	radius_m	texture_n	perimete	area_me	smoothn	compactr	concavity	concave p	symmetry	fractal_di	radius_se	texture_s	perimete	area_se	smoothn	compactr	concavity	concave p	symmetry	fractal_di	radius_w	texture_w	perimete	area_wor	smoothn	compactr	concavity	concave p	symmetry	fractal_di	Actual diagnosis	Predictions
2	859575	18.94	21.31	123.6	1130	0.09009	0.1029	0.108	0.07951	0.1582	0.05461	0.7888	0.7975	5.486	96.05	0.004444	0.01652	0.02269	0.0137	0.01386	0.001698	24.86	26.58	165.9	1866	0.1193	0.2336	0.2687	0.1789	0.2551	0.06589	M	Malignant
3	8611161	13.34	15.86	86.49	520	0.1078	0.1535	0.1169	0.06987	0.1942	0.06902	0.286	1.016	1.535	12.96	0.006794	0.03575	0.0398	0.01383	0.02134	0.004603	15.53	23.19	96.66	614.9	0.1536	0.4791	0.4858	0.1708	0.3527	0.1016	B	Benign
4	859717	17.2	24.52	114.2	929.4	0.1071	0.183	0.1692	0.07944	0.1927	0.06487	0.5907	1.041	3.705	69.47	0.00582	0.05616	0.04252	0.01127	0.01527	0.006299	23.32	33.82	151.6	1681	0.1585	0.7394	0.6566	0.1899	0.3313	0.1339	M	Malignant
5	8911163	17.93	24.48	115.2	998.9	0.08855	0.07027	0.05699	0.04744	0.1538	0.0551	0.4212	1.433	2.765	45.81	0.005444	0.01169	0.01622	0.008522	0.01419	0.002751	20.92	34.69	135.1	1320	0.1315	0.1806	0.208	0.1136	0.2504	0.07948	M	Malignant
6	913102	14.64	16.85	94.21	666	0.08641	0.06698	0.05192	0.02791	0.1409	0.05355	0.2204	1.006	1.471	19.98	0.003535	0.01393	0.018	0.006144	0.01254	0.001219	16.46	25.44	106	831	0.1142	0.207	0.2437	0.07828	0.2455	0.06596	B	Benign
7	8610908	12.86	18	83.19	506.3	0.09934	0.09546	0.03889	0.02315	0.1718	0.05997	0.2655	1.095	1.778	20.35	0.005293	0.01661	0.02071	0.008179	0.01748	0.002848	14.24	24.82	91.88	622.1	0.1289	0.2141	0.1731	0.07926	0.2779	0.07918	B	Benign
8	912193	12.16	18.03	78.29	455.3	0.09087	0.07838	0.02916	0.01527	0.1464	0.06284	0.2194	1.19	1.678	16.26	0.004911	0.01666	0.01397	0.005161	0.01454	0.001858	13.34	27.87	88.83	547.4	0.1208	0.2279	0.162	0.0569	0.2406	0.07729	B	Benign
9	8610637	18.05	16.15	120.2	1006	0.1065	0.2146	0.1684	0.108	0.2152	0.06673	0.9806	0.5505	6.311	134.8	0.00794	0.05839	0.04658	0.0207	0.02591	0.007054	22.39	18.91	150.1	1610	0.1478	0.5634	0.3786	0.2102	0.3751	0.1108	M	Malignant
10	8610862	20.18	23.97	143.7	1245	0.1286	0.3454	0.3754	0.1604	0.2906	0.08142	0.9317	1.885	8.649	116.4	0.01038	0.06835	0.1091	0.02593	0.07895	0.005987	23.37	31.72	170.3	1623	0.1639	0.6164	0.7681	0.2508	0.544	0.09964	M	Malignant
11	91227	13.9	19.24	88.73	602.9	0.07991	0.05326	0.02995	0.0207	0.1579	0.05594	0.3316	0.9264	2.056	28.41	0.003704	0.01082	0.0153	0.006275	0.01062	0.002217	16.41	26.42	104.4	830.5	0.1064	0.1415	0.1673	0.0815	0.2356	0.07603	B	Benign
12	912519	13.47	14.06	87.32	546.3	0.1071	0.1155	0.05786	0.05266	0.1779	0.06639	0.1588	0.5733	1.102	12.84	0.00445	0.01452	0.01334	0.008791	0.01698	0.002787	14.83	18.32	94.94	660.2	0.1393	0.2499	0.1848	0.1335	0.3227	0.09326	B	Benign
13	927241	20.6	29.33	140.1	1265	0.1178	0.277	0.3514	0.152	0.2397	0.07016	0.726	1.595	5.772	86.22	0.006522	0.06158	0.07117	0.01664	0.02324	0.006185	25.74	39.42	184.6	1821	0.165	0.8681	0.9387	0.265	0.4087	0.124	M	Malignant
14	912558	13.7	17.64	87.76	571.1	0.0995	0.07957	0.04548	0.0316	0.1732	0.06088	0.2431	0.9462	1.564	20.64	0.003245	0.008186	0.01698	0.009233	0.01285	0.001524	14.96	23.53	95.78	686.5	0.1199	0.1346	0.1742	0.09077	0.2518	0.0696	B	Benign
15	8610404	16.07	19.65	104.1	817.7	0.09168	0.08424	0.09769	0.06638	0.1798	0.05391	0.7474	1.016	5.029	79.25	0.01082	0.02203	0.035	0.01809	0.0155	0.001948	19.77	24.56	128.8	1223	0.15	0.2045	0.2829	0.152	0.265	0.06387	M	Malignant
16	884180	19.4	23.5	129.1	1155	0.1027	0.1558	0.2049	0.08886	0.1978	0.06	0.5243	1.802	4.037	60.41	0.01061	0.03252	0.03915	0.01559	0.02186	0.003949	21.65	30.53	144.9	1417	0.1463	0.2968	0.3458	0.1564	0.292	0.07614	M	Malignant

Chapter 10: Conclusion

In this project, we talked about machine learning (ML) ideas and described how they may be used to predict cancer. It is clear from a study of their findings that combining multidimensional heterogeneous data with the use of various feature selection and classification algorithms might result in useful inference tools for the cancer domain. The proposed machine-learning approaches could predict breast cancer as the early detection of this disease could help slow down the progress of the disease and reduce the mortality rate through appropriate therapeutic interventions at the right time.

Acknowledgement

We have great pleasure in presenting the mini project report on “**Cancer Detection”.** We take this opportunity to express our sincere thanks towards our guide **Prof**.**Uma Ade** Department of Computer Engineering, Watumull thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Dhanjay Raut** Head of Department, Computer Engineering, Watumull for his encouragement during progress meeting and providing guidelines to write this report.

We also thank the entire staff of Watumull for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of Watumull for their encouragement.

Nilesh Parab (D221452)
Ananya Yadav (D221427)
Riya Singh (D221421)
Vivek Tiwari (D221422)

Bibliography

- https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)
- https://www.cancer.net/navigating-cancer-care/diagnosing-cancer/stages-cancer
- https://www.javatpoint.com/machine-learning-random-forest-algorithm
- https://www.cancer.org/treatment/understanding-your-diagnosis/tests/ct-scan-for-cancer.html