A PROJECT REPORT ON

PREDICTION OF CANCEROUS CELL

BY USING

PROBABILISTIC MODEL OF MACHINE LEARNING



**SUBMITTED BY:**

SUBHOBRATA SINHA

SHASHWATA BASU

SUBHASISH DEB

RAHUL RANJAN RAJ

**PROJECT GUIDE:**

SOFIKUL MULLICK



**CERTIFICATE FROM SUPERVISOR:**

THIS IS TO CERTIFY THAT RAHUL RANJAN RAJ, SUBHOBRATA SINHA, SUBHASISH DEB, SHASHWATA BASU HAVE SUCCESSFULLY COMPLETED THE PROJECT TITLED “DETECTION OF CANCEROUS CELL USING MACHINE LEARNING” UNDER MY SUPERVISION DURING THE PERIOD of JULY TO AUGUST WHICH IS IN PARTIAL FULLFILLMENT OF REQUIRMENTS FOR THE AWARD OF BACHAELOR OF TECHNOLOGY AND SUBMITTED TO ME DEAPARTMENT OF TECHNO MAIN SALT LAKE, KOLKATA.

DATE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SIGNATURE OF SUPERVISOR

SOFKUL MULLICK (PROJECT ENGINEER, ARDENT COMPUTECH PVT. LTD.)

# Acknowledgment

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project MENTOR, SOFIKUL Mullick for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

## ABSTRACT

Every sixth death in the world is due to cancer, making it the second leading cause of death (second only to cardiovascular diseases). In 2017, 9.6 million people are estimated to have died from the various forms of cancer. The Institute for Health Metrics and Evaluation (IHME) put relatively small error margins around this global figure: the lower and upper estimates extend from 9.2 to 9.7 million.

Progress against many other causes of deaths and demographic drivers of increasing [population size](https://ourworldindata.org/world-population-growth#shares-by-world-regions), [life expectancy](https://ourworldindata.org/life-expectancy) and — particularly in higher-income countries — aging populations mean that the total number of cancer deaths continues to increase. This is a very personal topic to many: nearly everyone knows or has lost someone dear to them from this collection of diseases.

Using data mining methods to aid people to predict diabetes has gain major popularity. In this project, Logistic Regression is used to predict the persons whether diabetic or not. Logistic Regression are considered as helpful methods for the diagnosis of many diseases.

Cancers are defined by the [National Cancer Institute (NCI)](https://www.cancer.gov/about-cancer/understanding/what-is-cancer) as a collection of diseases in which abnormal cells can divide and spread to nearby tissue. As this definition suggests, cancers can arise in many parts of the body (leading to a range of cancer types, as shown below) and in some cases spread to other parts of the body through the blood and lymph systems.

They, in fact, are probable models which have been proved useful in displaying complex systems and showing the relationships between variables in a graphic way. The dataset used is from National Cancer Institute dataset, which collects the information of persons with and without cancerous cell.

**INTRODUCTION:**

Definition:-Cancers are a large family of diseases that involve abnormal [cell growth](https://en.wikipedia.org/wiki/Cell_growth) with the potential to invade or spread to other parts of the body. They form a subset of [neoplasms](https://en.wikipedia.org/wiki/Neoplasm). A neoplasm or tumor is a group of cells that have undergone unregulated growth and will often form a mass or lump, but may be distributed diffusely.

Classification: -

Cancers are classified by the [type of cell](https://en.wikipedia.org/wiki/List_of_distinct_cell_types_in_the_adult_human_body) that the tumor cells resemble and is therefore presumed to be the origin of the tumor. These types include:

* [Carcinoma](https://en.wikipedia.org/wiki/Carcinoma): Cancers derived from [epithelial](https://en.wikipedia.org/wiki/Epithelium) cells. This group includes many of the most common cancers and include nearly all those in the [breast](https://en.wikipedia.org/wiki/Breast_cancer), [prostate](https://en.wikipedia.org/wiki/Prostate_cancer), [lung](https://en.wikipedia.org/wiki/Lung_cancer), [pancreas](https://en.wikipedia.org/wiki/Pancreas) and [colon](https://en.wikipedia.org/wiki/Colorectal_cancer).
* [Sarcoma](https://en.wikipedia.org/wiki/Sarcoma): Cancers arising from [connective tissue](https://en.wikipedia.org/wiki/Connective_tissue) (i.e. [bone](https://en.wikipedia.org/wiki/Bone), [cartilage](https://en.wikipedia.org/wiki/Cartilage), fat, [nerve](https://en.wikipedia.org/wiki/Nerve)), each of which develops from cells originating in [mesenchymal](https://en.wikipedia.org/wiki/Mesenchyme) cells outside the bone marrow.
* [Lymphoma](https://en.wikipedia.org/wiki/Lymphoma) and [leukemia](https://en.wikipedia.org/wiki/Leukemia): These two classes arise from hematopoietic (blood-forming) cells that leave the marrow and tend to mature in the lymph nodes and blood, respectively.
* [Germ cell tumor](https://en.wikipedia.org/wiki/Germ_cell_tumor): Cancers derived from [pluripotent](https://en.wikipedia.org/wiki/Pluripotent) cells, most often presenting in the [testicle](https://en.wikipedia.org/wiki/Testicular_cancer) or the [ovary](https://en.wikipedia.org/wiki/Ovarian_cancer) ([seminoma](https://en.wikipedia.org/wiki/Seminoma) and [dysgerminoma](https://en.wikipedia.org/wiki/Dysgerminoma), respectively).
* [Blastoma](https://en.wikipedia.org/wiki/Blastoma): Cancers derived from immature "precursor" cells or embryonic tissue.

MACHINE LEARNING:

Machine learning is a branch of computer science that consists of algorithms that can learn from data, it provides set of methods that can detect patterns in the data and use the patterns to generate future predictions.

Because of new computing technologies, machine learning today is not like machine learning of the past. With the rise of Machine Learning approaches, we have the ability to find a solution to this issue, we have developed a system using logistic regression which has the ability to predict whether the patient has diabetes or not.

Furthermore, predicting the disease early leads to treating the patients before it becomes critical. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results.

There are three core types of machine learning- supervised learning, unsupervised learning, and reinforcement learning.

1. Supervised Learning: The main goal in supervised learning is to learn a model from labeled training data that allows us to make predictions about unseen or future data. Here, the term supervised refers to a set of samples where the desired output signals (labels) are already known.
2. Semi-supervised Learning: It uses both labeled and unlabeled data for training-typically a small amount of labeled data and large amount of unlabeled data (because unlabeled data is less expensive and take less effort to acquire).
3. Reinforcement Learning: This learning used for robotics, gaming and navigation. With reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest rewards

PROJECT OBJECTIVE:

* The main objective of this project is to predict the diabetes.
* The purposed work focuses on to predict diabetes using probabilistic model of Regression.
* This integrated technique of classification gives a promising classification results with utmost accuracy rate.
* For detecting a disease, a number of tests should be required from the patient.
* But by using logistic regression method the number of tests should be reduced. This reduced test plays an important role in time and performance.
* This disease is referred to cancer. The cause of cancer is mainly attributed to carcinogens.
* In early the ability to diagnose cancer plays an important role for the patient’s treatment process via chemotherapy.

Problem Statement and DESCRIPTION:

Prediction of cancerous cell using Logistic Regression: To identify whether a given person in dataset will be having cancerous cell or no cancerous cell based on the basis of the given attributed values.

Dataset contains all the details of person like radius mean, texture mean, perimeter mean, smoothness means, concave position mean, perimeter worst, area worst, symmetry worst, fractal dimension worst texture worst.

Attributes like smoothness mean, radius mean, texture mean values exceeding a specific value may contribute to identify whether a person is having breast cancer or not.

The aim of prediction of cancer is to make people aware and what it takes to treat it and gives the power to control. It makes necessary chances to improve lifestyle, classification and evaluation for the prediction of Unsupervised learning to predict whether the person has cancer cell. The proposed Logistic Regression will predict the persons having cancer or not.

HARDWARE & SOFTWARE REQUIRMENTS:

HARDWARE USED:

* Intel Core i5(7th gen, 2.2Ghz, Cache 3M)
* 4gb DDR3 Ram
* Hard Disk
* Intel HD Graphics

SOFTWARE USED:

* Jupyter Notebook
* Python 3.7
* .CSV file format

MINIMUM SOFTWARE REQUIRMENTS:

* 2 gb of Ram
* 20 gb of HDD
* Any Operating system

## The Logistic Regression Algorithm

## Logistic Regression is one of the most used Machine Learning algorithms for binary classification. It is a simple Algorithm that you can use as a performance baseline, it is easy to implement and it will do well enough in many tasks. Therefore, every Machine Learning engineer should be familiar with its concepts. The building block concepts of Logistic Regression can also be helpful in deep learning while building neural networks.

Like many other machine learning techniques, it is borrowed from the field of statistics and despite its name, it is not an algorithm for regression problems, where you want to predict a continuous outcome. Instead, Logistic Regression is the go-to method for binary classification. It gives you a discrete binary outcome between 0 and 1. To say it in simpler words, its outcome is either one thing or another.

A simple example of a Logistic Regression problem would be an algorithm used for cancerous cell detection that takes an input and should tell if a patient has diabetes (1) or not (0).

## How it works

Logistic Regression measures the relationship between the dependent variable (our label, what we want to predict) and the one or more independent variables (our features), by estimating probabilities using its underlying logistic function.

These probabilities must then be transformed into binary values in order to actually make a prediction.

## Advantages of logistic regression:

* It is more robust: the independent variables don’t have to be normally distributed, or have equal variance in each group.
* It does not assume a linear relationship between the IV and DV.
* It may handle nonlinear effects.
* You can add explicit interaction and power terms.
* The DV need not be normally distributed.
* There is no homogeneity of variance assumption.
* Normally distributed error terms are not assumed.
* It does not require that the independents be interval.
* It does not require that the independents be unbounded.

## Disadvantages of logistic regression:

* Identifying Independent Variables
* Limited Outcome Variables
* Independent Observations Required
* Overfitting the Model

PACKAGES USED:

Python Tools:

Contains several basic utility functions including: shape() for counting the number of columns and rows in the dataset, isna().sum() to count the empty values (NaN) in each column, dropna(axis=1) to drop all the missing values (NaN) also having libraries such as numpy, pandas, matplotlib.pyplot, %matplotlib inline, seaborn

CLASSIFICATION & REGRESSION TRAINING:

Msc functions for training and plotting classification and regression models.

confusionMatrix Create a confusion matrix

confusionMatrix.default Create a confusion matrix

confusionMatrix.table Create a confusion matrix

COUNTPLOT:

A visualization of a correlation matrix.

A graphical display of correlation matrix or general matrix.

It also contains some algorithms to do matrix reordering. In addition, corrplot is good at details including choosing colour, text labels, colour labels, layout, etc.

## ROCR:

Visualizing the Performance of Scoring Classifiers.

ROC graphs, sensitivity/specificity curves, lift charts, and precision/recall plots are popular examples of trade-off visualizations for specific pairs of performance measures. ROCR is a flexible tool for creating cutoff- parameterized 2D performance curves by freely combining two from over 25 performance measures (new performance measures can be added using a standard interface).

Curves from different cross-validation or bootstrapping runs can be averaged by different methods, and standard deviations, standard

errors or box plots can be used to visualize the variability across the runs. The parameterization can be visualized by printing cutoff values at the corresponding curve positions, or by coloring the curve according to cutoff. All components of a performance plot can be quickly adjusted using a flexible parameter dispatching mechanism. Despite its flexibility, ROCR is easy to use, with only three commands and reasonable default values for all optional parameters.

|  |  |
| --- | --- |
| performance | Function to create performance objects |
| performance class | Class “performance” |
| plot-method | Plot method for performance objects |
| plot-methods | Plot method for performance objects |
| plot.performance | Plot method for performance objects |
| prediction | Function to create prediction objects |
| prediction.class | Class “prediction” |

PROJECT CODE:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

sns.countplot(df['diagnosis'], label='count')

df.hist(bins=10,figsize=(20,20),grid=False)

sns.pairplot(df,hue='diagnosis')

l=len(df.columns)

plt.rcParams['figure.figsize']=(10,4)

columns=list(df.columns)

columns.remove('diagnosis')

i=0

for col in columns:

if i%5==0:

i=0

axes=[i for i in range(5)]

f,axes=plt.subplots(1,5)

f.tight\_layout()

sns.boxplot('diagnosis',y=col,data=df,ax=axes[i-1])

i+=1

l=len(df.columns)

plt.rcParams['figure.figsize']=(10,4)

columns=list(df.columns)

columns.remove('diagnosis')

i=0

for col in columns:

if i%5==0:

i=0

axes=[i for i in range(5)]

f,axes=plt.subplots(1,5)

f.tight\_layout()

sns.swarmplot('diagnosis',y=col,data=df,ax=axes[i-1])

i+=1

l=len(df.columns)

plt.rcParams['figure.figsize']=(10,4)

columns=list(df.columns)

columns.remove('diagnosis')

i=0

for col in columns:

if i%5==0:

i=0

axes=[i for i in range(5)]

f,axes=plt.subplots(1,5)

f.tight\_layout()

sns.violinplot('diagnosis',y=col,data=df,ax=axes[i-1])

i+=1

pd.get\_dummies(df['diagnosis'])

diagnosis=pd.get\_dummies(df['diagnosis'],drop\_first=True)

diagnosis.head(5)

df1=pd.concat([df,diagnosis],axis=1)

df1.head(5)

df1.drop(['diagnosis'],axis=1,inplace=True)

df1.head(5)

df1.corr()

plt.figure(figsize=(20,20)) #This is used to change the size of the figure/ heatmap

sns.heatmap(df1.corr(), annot=True, fmt='.0%')

x=df1.drop('M',axis=1)

y=df1['M']

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test= train\_test\_split(x,y,test\_size=0.3, random\_state=0)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

from sklearn.linear\_model import LogisticRegression

logmodel=LogisticRegression()

logmodel.fit(x\_train,y\_train)

predictions=logmodel.predict(x\_test)

print(predictions)

from sklearn.metrics import classification\_report

confusion\_matrix(y\_test,predictions)

accuracy\_score(y\_test,predictions)

count the empty(NaN) values in each column

df.isna().sum()

id 0

diagnosis 0

radius\_mean 0

texture\_mean 0

perimeter\_mean 0

area\_mean 0

smoothness\_mean 0

compactness\_mean 0

concavity\_mean 0

concave points\_mean 0

symmetry\_mean 0

fractal\_dimension\_mean 0

radius\_se 0

texture\_se 0

perimeter\_se 0

area\_se 0

smoothness\_se 0

compactness\_se 0

concavity\_se 0

concave points\_se 0

symmetry\_se 0

fractal\_dimension\_se 0

radius\_worst 0

texture\_worst 0

perimeter\_worst 0

area\_worst 0

smoothness\_worst 0

compactness\_worst 0

concavity\_worst 0

concave points\_worst 0

symmetry\_worst 0

fractal\_dimension\_worst 0

Unnamed: 32 569

dtype: int64

get a count of the no of Malignant(M) (harmful) or Benign((B) cells (not harmful)

df['diagnosis'].value\_counts()

B 357

M 212

Name: diagnosis, dtype: int64

visualize this count

sns.countplot(df['diagnosis'], label='count')



Look at the data types to see which columns need to be transformed / encoded to a number

df.dtypes

id int64

diagnosis object

radius\_mean float64

texture\_mean float64

perimeter\_mean float64

area\_mean float64

smoothness\_mean float64

compactness\_mean float64

concavity\_mean float64

concave points\_mean float64

symmetry\_mean float64

fractal\_dimension\_mean float64

radius\_se float64

texture\_se float64

perimeter\_se float64

area\_se float64

smoothness\_se float64

compactness\_se float64

concavity\_se float64

concave points\_se float64

symmetry\_se float64

fractal\_dimension\_se float64

radius\_worst float64

texture\_worst float64

perimeter\_worst float64

area\_worst float64

smoothness\_worst float64

compactness\_worst float64

concavity\_worst float64

concave points\_worst float64

symmetry\_worst float64

fractal\_dimension\_worst float64

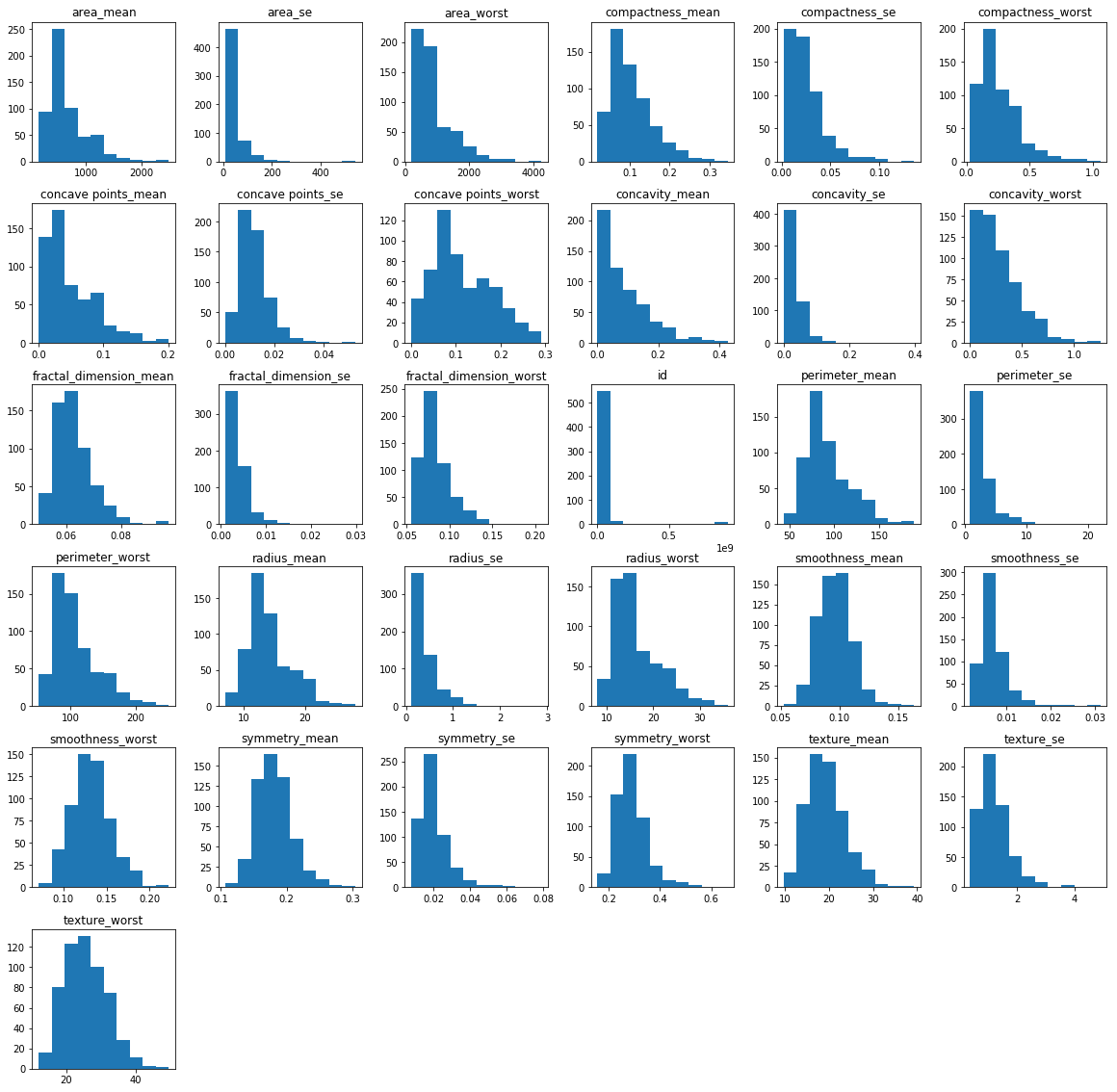
dtype: object

# Data Preprocessing

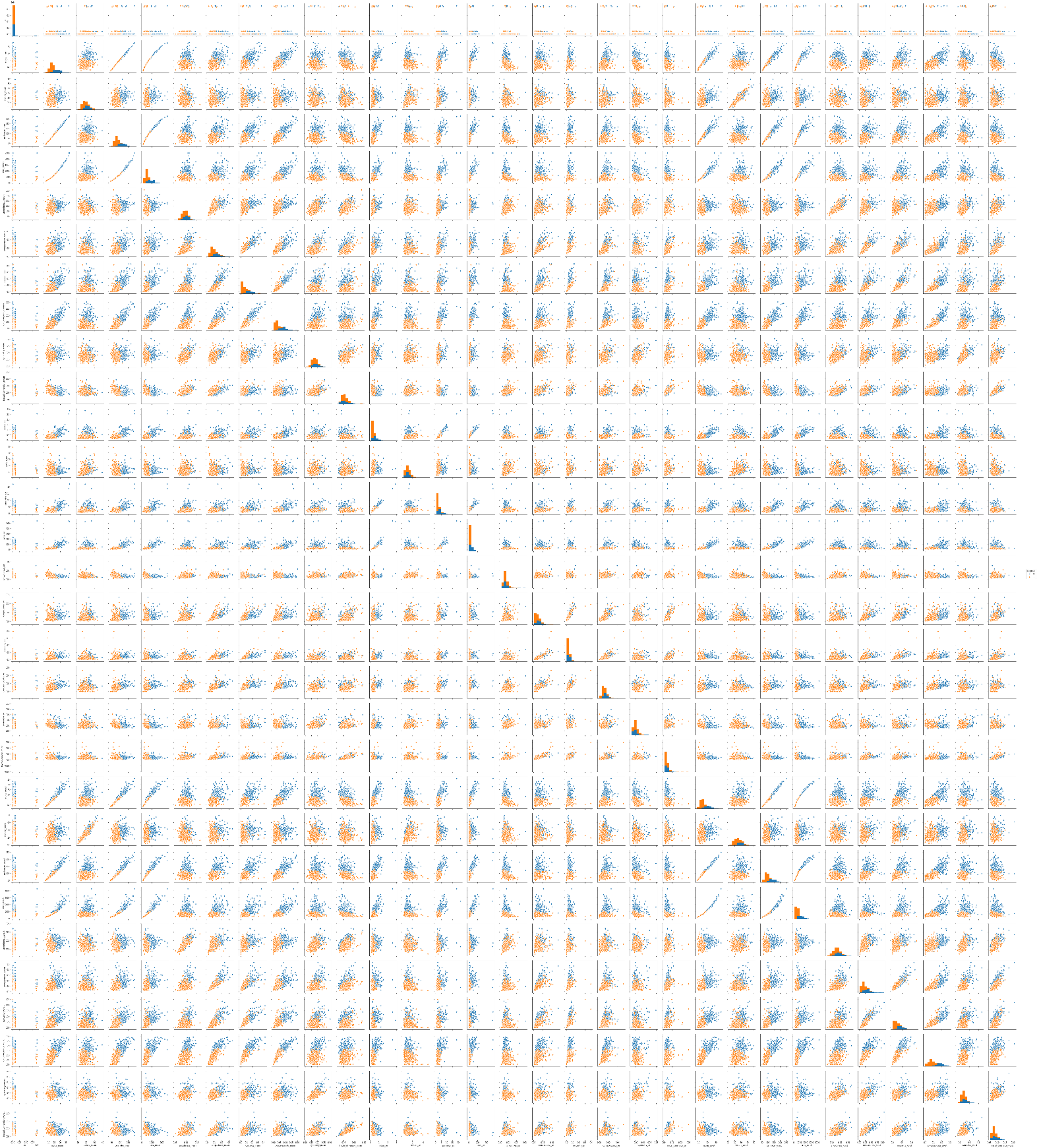
#Visualization

#getting insight of data distribution based on frequency of unique values in the features

df.hist(bins=10,figsize=(20,20),grid=False)



sns.pairplot(df,hue='diagnosis')



## Plotting of boxplots of different attributes against diagnosis(malignant,benign)

l=len(df.columns)

plt.rcParams['figure.figsize']=(10,4)

columns=list(df.columns)

columns.remove('diagnosis')

i=0

for col in columns:

if i%5==0:

i=0

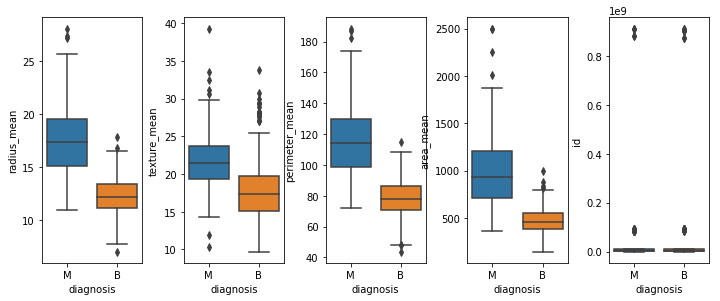
axes=[i for i in range(5)]

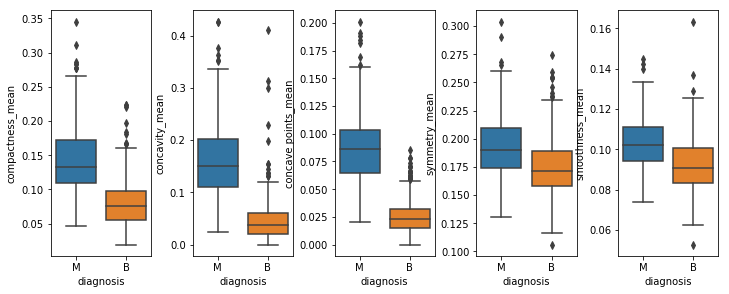
f,axes=plt.subplots(1,5)

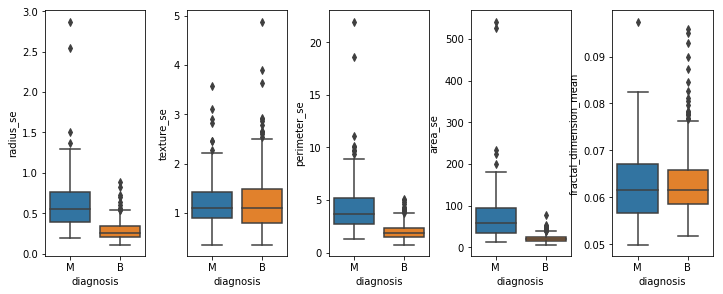
f.tight\_layout()

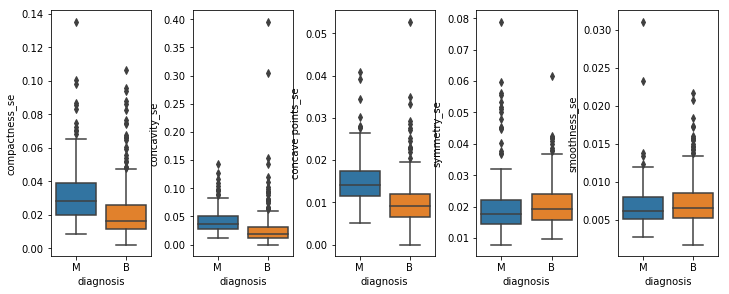
sns.boxplot('diagnosis',y=col,data=df,ax=axes[i-1])

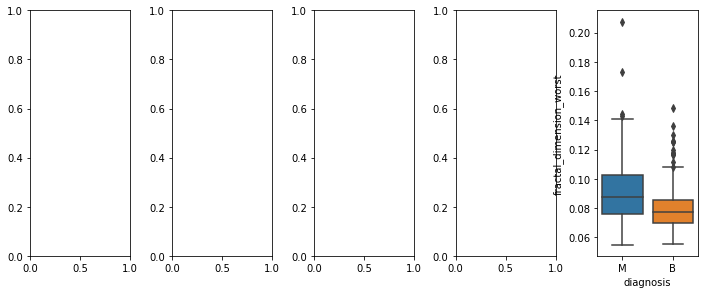
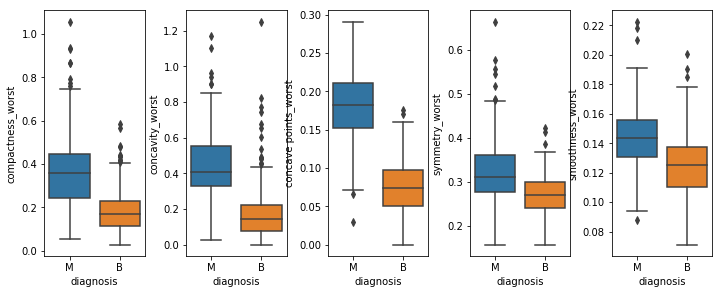
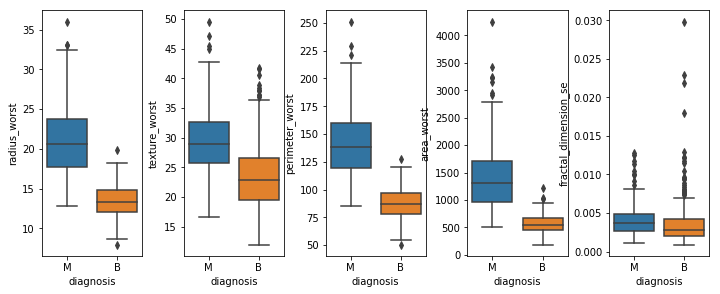
i+=1











l=len(df.columns)

plt.rcParams['figure.figsize']=(10,4)

columns=list(df.columns)

columns.remove('diagnosis')

i=0

for col in columns:

if i%5==0:

i=0

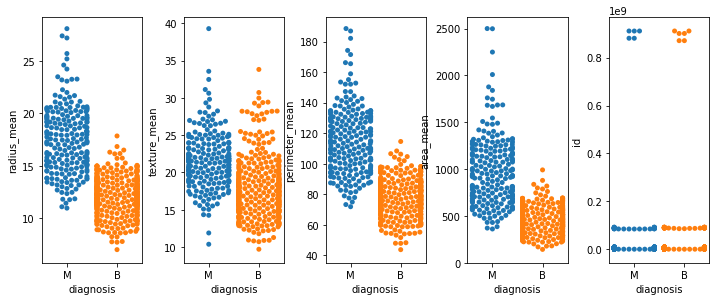
axes=[i for i in range(5)]

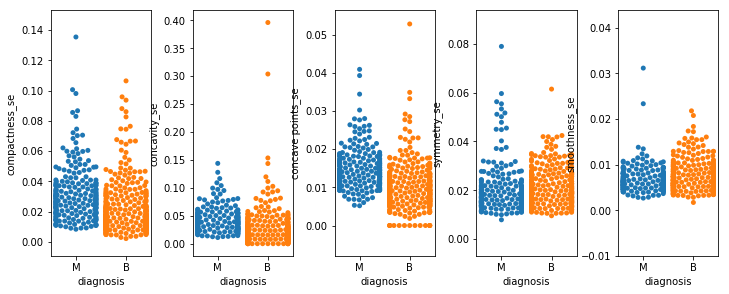
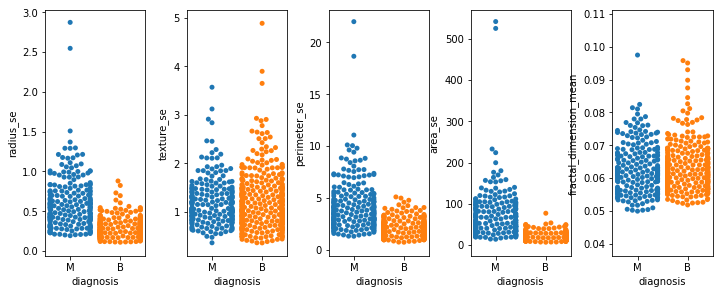
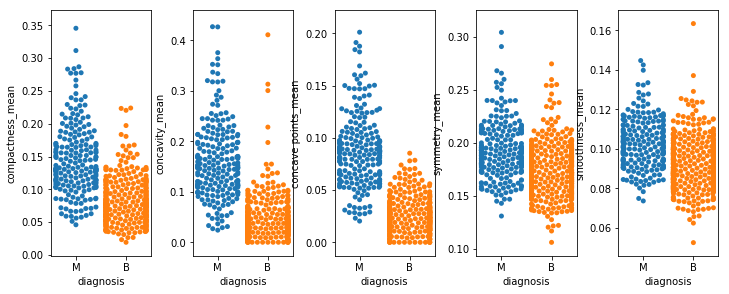
f,axes=plt.subplots(1,5)

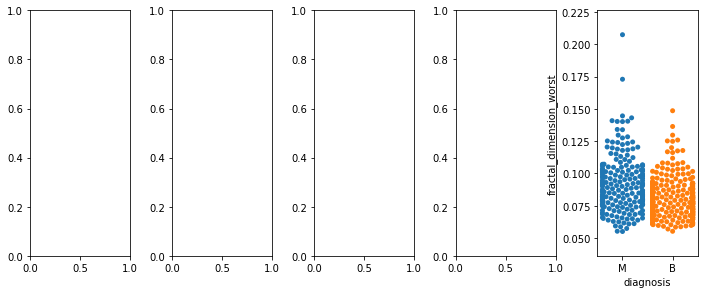
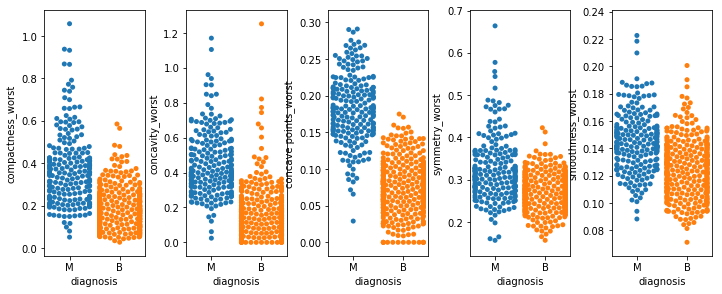
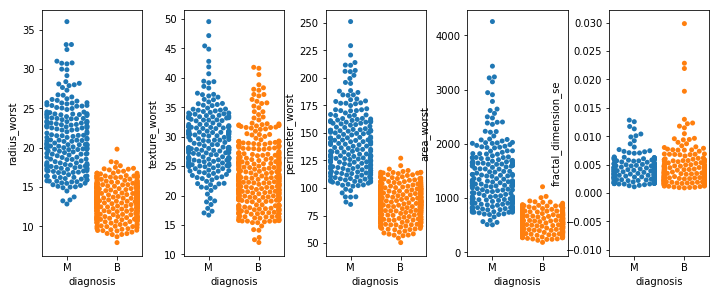
f.tight\_layout()

sns.swarmplot('diagnosis',y=col,data=df,ax=axes[i-1])

i+=1







l=len(df.columns)

plt.rcParams['figure.figsize']=(10,4)

columns=list(df.columns)

columns.remove('diagnosis')

i=0

for col in columns:

if i%5==0:

i=0

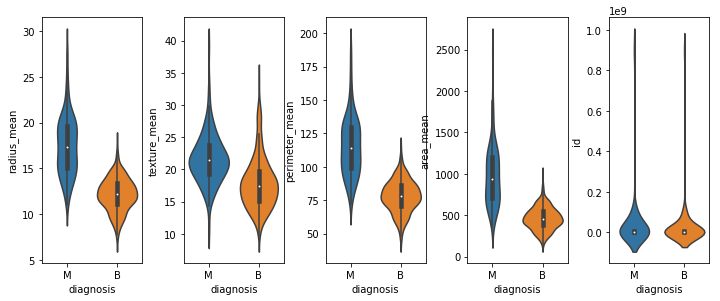
axes=[i for i in range(5)]

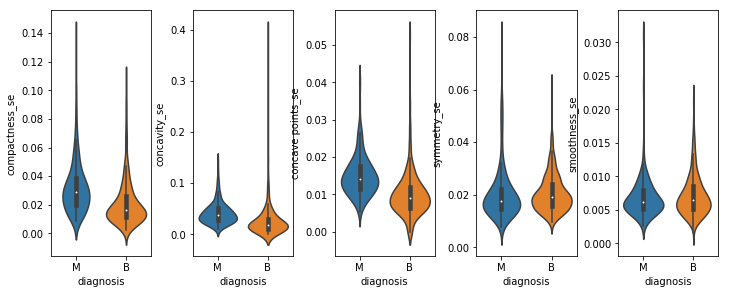
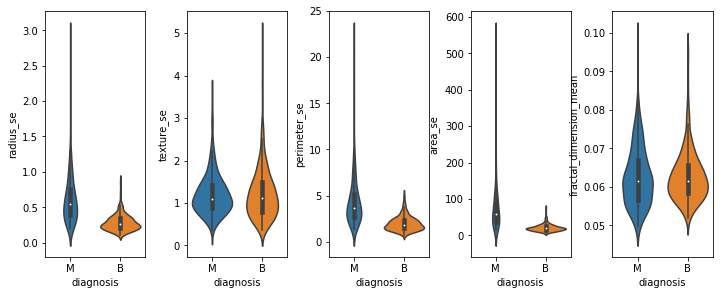
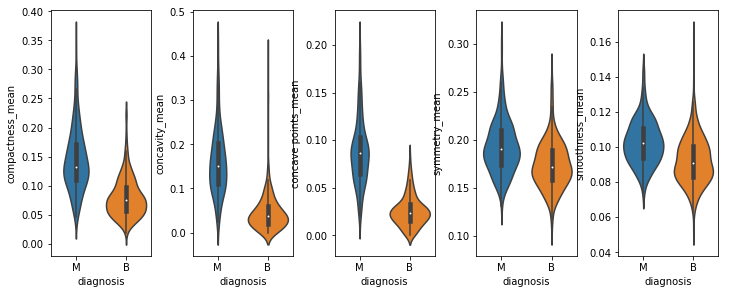
f,axes=plt.subplots(1,5)

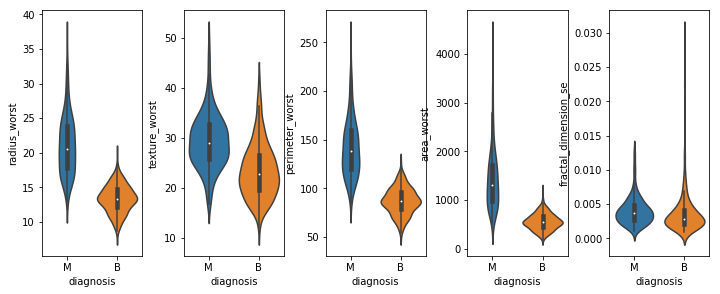
f.tight\_layout()

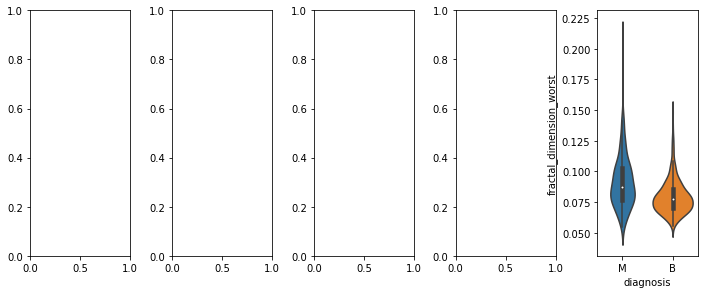
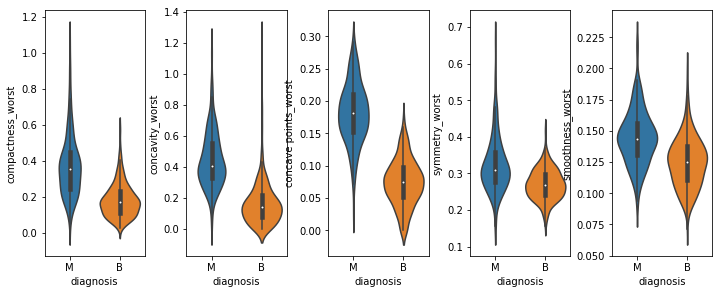
sns.violinplot('diagnosis',y=col,data=df,ax=axes[i-1])

i+=1

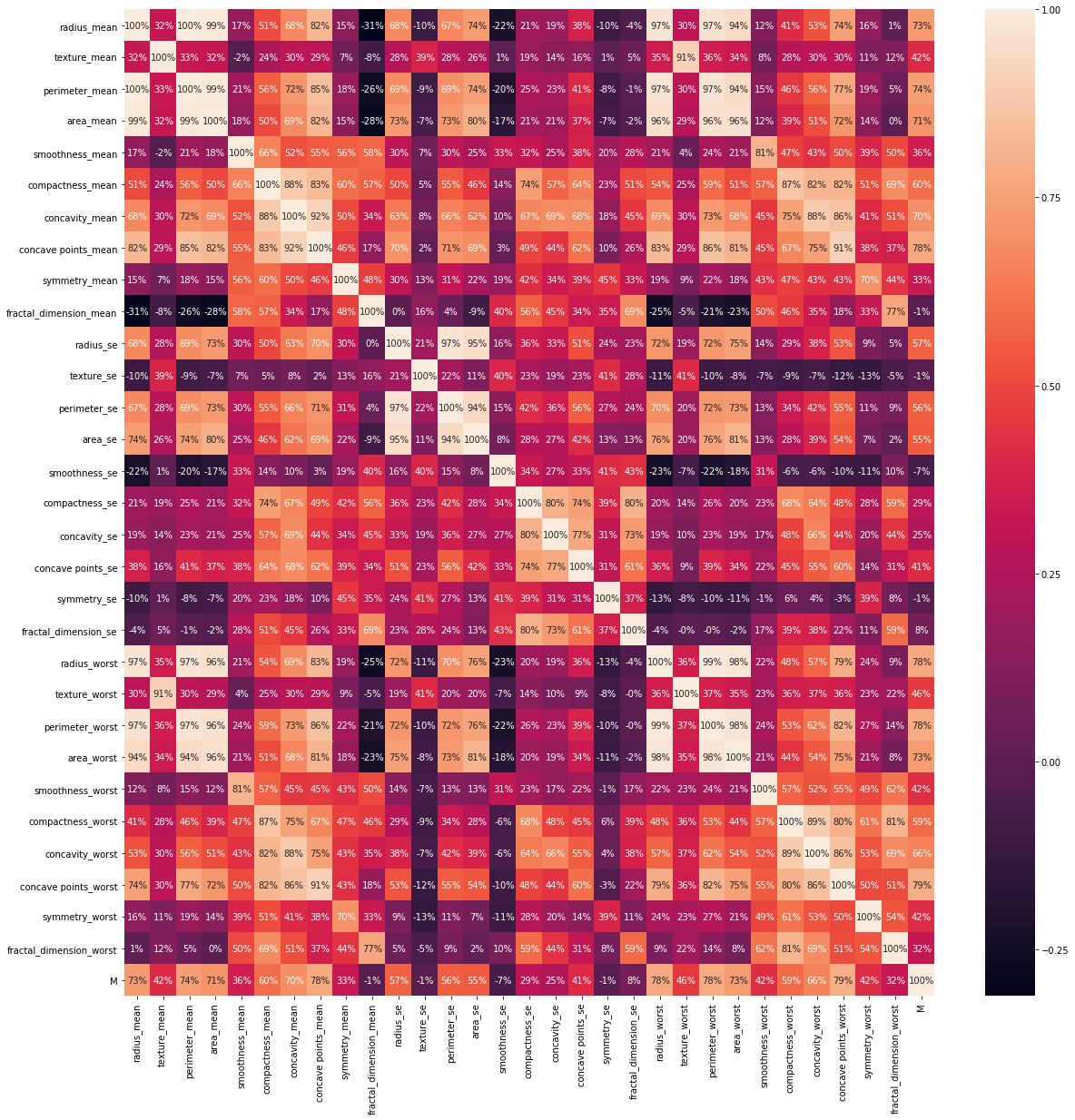








Visualize the correlation



PROCEDURE

# import libraries

# load data

# count columns with missing value

# processing the data

# get dummies

# drop columns with missing values

# train the model

# test the model

# get prediction and prediction accuracy

## Results:

The problem of work is about predicting whether a person has cancerous cell or not in a dataset by applying Logistic Regression. This problem is solved using the primary attribute.

So, for predicting the accuracy of the program we give

accuracy\_score(y\_test,predictions) which gives the accuracy rate at 0.976608.

# CONCLUSION

Lately, medical machine learning has gained in interest by the scientific and research communities. Cancer is considered as the world's fastest-growing chronic disease. It needs continuous self-management. The vast majority of cancer cases are due to environmental risk factors. Many of these environmental factors are controllable lifestyle choices. Thus, cancer is generally preventable. Between 70% and 90% of common cancers are due to environmental factors and therefore potentially preventable.

Cancer is a group of diseases involving abnormal [cell growth](https://en.wikipedia.org/wiki/Cell_growth) with the potential to invade or spread to other parts of the body. These contrast with [benign tumors](https://en.wikipedia.org/wiki/Benign_tumor), which do not spread. Possible [signs and symptoms](https://en.wikipedia.org/wiki/Cancer_signs_and_symptoms) include a lump, abnormal bleeding, prolonged cough, unexplained [weight loss](https://en.wikipedia.org/wiki/Weight_loss), and a change in [bowel movements](https://en.wikipedia.org/wiki/Defecation). While these symptoms may indicate cancer, they can also have other causes. Over 100 types of cancers affect humans. We proposed a model in predicting diabetes by applying data mining technique.

Cancer is a chronic disease and a major public health challenge worldwide. Using machine learning to aid people to predict cancer has gain major popularity. In this Logistic Regression is proposed to predict whether the person has cancer or not.

Results have been obtained. Moreover, we recommend the proposed models to be tested on a larger dataset.

# LIMITATIONS

The logistic regression model has been used for prediction of cancerous cell. In this model, we have achieved accuracy level of approximate 97%, using different model could help in increasing the accuracy level.

FUTURE SCOPE

This project has accuracy of 97.6608% and it can be increased by using different approximations in future.