11 1

by 11 1 11 1

Submission date: 01-Jun-2022 08:09AM (UTC+0530)
Submission ID: 1848195711
File name: Breast_Cancer_Report_Final_CO19322.docx (3.25M)
Word count: 4340
Character count: 28192

Chandigarh College of Engineering and Technology
(Degree Wing)

Compiler Design (CS – 605C)
Final Project File

Submitted By
Dipesh Singla (CO19322)
Ashishraj Kalkhandey (CO19318)
C.S.E 6th Semester

Submitted To
Dr. Varun Gupta
Professor CSE, HOD (A.S)

TABLE OF CONTENTS

ABSTRACT

Women are genuinely sabotaged by chest illness with high frightfulness and mortality. The shortfall of enthusiastic perception models achieves inconvenience for experts to set up a treatment plan that could draw out calm perseverance time. Thusly, the essential of time is to cultivate the methodology which gives least bumble to increase accuracy. Four computation SVM, Strategic Relapse, Arbitrary Timberland and KNN which expect the chest threatening development result have been taken a gander at in the paper using different datasets. All preliminaries are executed inside a multiplication environment and coordinated in JUPYTER stage. Mark of assessment sorts in three spaces. First region is conjecture of illness before investigation, second space is assumption for assurance and treatment and third region revolves around result during treatment. The proposed work can be used to anticipate the consequence of different method and suitable system can be used depending on essential. This investigation is finished to expect the precision. The future investigation can be finished to anticipate the other different limits and chest harmful development assessment can be sorts on reason of various limits.

Keywords — Breast Cancer, machine learning, feature selection, classification, prediction, KNN , Random Forest.

INTRODUCTION

The ensuing huge justification for women's passing is chest threatening development (after cell breakdown in the lungs). 246,660 of women's new occasions of prominent chest sickness should be examined in the INDIA during 2016 and 40,450 of women's destruction is evaluated. Chest dangerous development is a kind of sickness that starting points in the chest. Sickness starts when cells begin to grow out of control. Chest infection cells regularly structure a development that can habitually be seen on a x-shaft or felt as a knock. Chest infection can spread when the dangerous development cells get into the blood or lymph system and are passed on to various bits of the body. The justification for Bosom Disease recalls changes and changes for DNA. There are different kinds of chest dangerous development and typical ones consolidate ductal carcinoma in situ (DCIS) and prominent carcinoma. Others, like phyllodes tumors and angiosarcoma are more surprising. There are various estimations for course of action of chest dangerous development results. The side effects of Bosom Malignant development are - Weakness, Migraines, Torment and deadness (periphery neuropathy), Bone mishap and osteoporosis. There are various estimations for portrayal and assumption for chest harmful development results. The ongoing paper gives an assessment between the show of four classifiers: SVM , Strategic Relapse , Irregular Backwoods and kNN which are among the most convincing data mining computations. It might be restoratively recognized exactly on schedule during a screening evaluation through mammography or by helpful harmful development expressive gadget. Damaging chest tissues change with the development of the infection, which can be directly associated with threatening development arranging. The period of chest dangerous development (I-IV) depicts how far

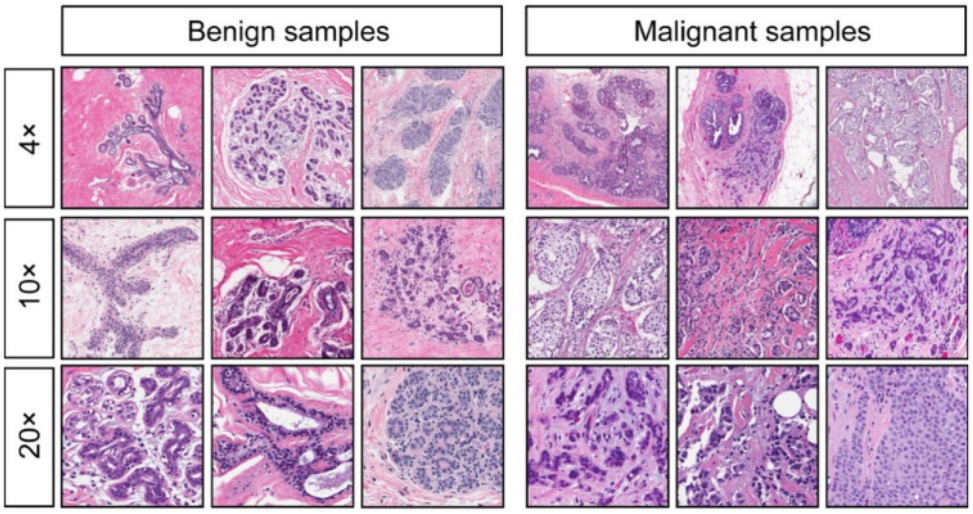


Figure 1: Breast Cancer basic classification based on severity

a patient's sickness has duplicated. Verifiable pointers, for instance, disease size, lymph center metastasis, and far away metastasis, and so on are used to choose stages. To hold sickness back from spreading, patients need to go through chest threatening development operation, chemotherapy, radiotherapy and endocrine. The goal of the investigation is to recognize and bunch Harmful and Harmless patients and anticipating that how might parametrize our course of action techniques subsequently to achieve high accuracy. We are exploring various datasets and how further AI computations can be used to depict Bosom Malignant development. We really want to diminish the screw up rates with most prominent precision. 10-overlay cross endorsement test which is an AI Strategy is used in JUPYTER to evaluate the data and take apart data concerning suitability and efficiency.

MOTIVATION FOR DOING THIS PROJECT

Bosom Malignant development is the most affected ailment present in women all over the planet. 246,660 of women's new occurrences of prominent chest illness should be dissected in the INDIA during 2016 and 40,450 of women's destruction is evaluated. The improvement in Bosom Malignant development and its conjecture spellbound. The UCI Wisconsin AI Vault Bosom Malignant development Dataset pulled in as gigantic patients with multivariate characteristics were taken as test set.

RELATED WORK

The justification behind Bosom Disease recollects changes and changes for DNA. Infection starts when cells begin to grow out of control. Chest illness cells normally structure a development that can regularly be seen on a x-pillar or felt as a bulge. There are different kinds of chest sickness and typical ones consolidate ductal carcinoma in situ (DCIS) and prominent carcinoma. Others, like phyllodes developments and angiosarcoma are more surprising. Wang, D.; Zhang and Y.- H Huang (2018) et al. [1] utilized Logistic

Regression and accomplished an Accuracy of 96.4 %. Akbugday etal., [2] performed arrangement on Breast Cancer Dataset by utilizing KNN, SVM and accomplished precision of 96.85%. KAYA KELES et al., [3] in the paper named "Bosom Cancer Prediction and Detection Using Data Mining" utilized Random Forest and accomplished precision of 92.2%.Vikas Chaurasia and Saurabh Pal et al., [4] think about the presentation standard of regulated learning classifiers, for example, Naïve Bayes, SVM-RBF piece, RBF brain organizations, Decision trees (J48) and straightforward CART; to find the best classifier in bosom disease datasets. Dalen, D.Walker and G. Kadam et al. [5] utilized ADABOOST and accomplished precision of 97.5% better than Random Forest. Kavitha et al., [6] utilized group techniques with Neural Networks and accomplished precision of 96.3% lesser than past examinations. As indicated by Sinthia et al., [7]. used backpropagation system with 94.2 % accuracy. The exploratory result shows that SVM-RBF piece is more exact than various classifiers; it scores precision of 96.84% in Wisconsin Bosom Disease (extraordinary) datasets . We have used request systems like SVM, KNN, Arbitrary Woods, Credulous Bayes, ANN. Assumption and gauge of illness progression are based on three huge spaces: risk assessment or assumption for harmful development lack of protection, conjecture of infection break faith, and estimate of sickness perseverance rate. The essential region incorporates assumption for the probability of cultivating explicit harmful development before the patient diagnostics. The ensuing issue is associated with assumption for threatening development rehash concerning diagnostics and treatment, and the third case is centered around conjecture of a couple of potential limits depicting harmful development improvement and treatment after the finding of the sickness: perseverance time, future, development, drug responsiveness, etc. The survivability rate and the dangerous development fall away from the faith are dependent particularly on the clinical treatment and the idea of the investigation. As we likely know that data pre-taking care of is a data digging strategy that used for divert data in a usable design. Since this current reality dataset essentially open in different association. It isn't open as per our essential so it ought to be filtered in sensible association. Data pre-dealing with is a shown method for settling such issues. Data pre-taking care of convert the dataset into usable association for pre-dealing with we have used standardization method.

DATA SET

Highlights are processed from a digitized picture of a fine needle suction (FNA) of a bosom mass. They portray attributes of the cell cores present in the picture. A couple of the pictures can be found at [Web Link]

Isolating plane portrayed above was gotten utilizing Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Choice Tree Construction Via Linear Programming." Proceedings of the fourth Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a characterization strategy which utilizes direct programming to build a choice tree. Pertinent elements were chosen involving a thorough pursuit in about 1-4 highlights and 1-3 isolating planes.

The real straight program used to get the isolating plane in the 3-layered space is that portrayed in: [K. P. Bennett and O. L. Mangasarian: "Powerful Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This data set is additionally accessible through the UW CS ftp server: ftp ftp.cs.wisc.edu disc math-prog/cpo-dataset/machine-learn/WDBC/

ATTRIBUTE INFORMATION:

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

    - radius (mean of distances from center to points on the perimeter)
    - texture (standard deviation of gray-scale values)
    - perimeter
    - area
    - smoothness (local variation in radius lengths)
    - compactness (perimeter^2 / area - 1.0)
    - concavity (severity of concave portions of the contour)
    - concave points (number of concave portions of the contour)
    - symmetry
    - fractal dimension ("coastline approximation" - 1)

TABLE OF COMPARISSON

TECHNIQUE USED	TOOL USED	ACCURACY	ERROR RATE	AUTHOR
Logistic Regression	WEKA	96.4 %	0.33%	Wang et al.
Statistical Feature Selection	WEKA	92.3 %	0.3%	V Chaurisya & S Paul
KNN and SVM	WEKA	KNN - 96.85% NAÏVE BAYES - 95.99% SVM – 96.85%	0.66%	Akbugday
SVM vs KNN, decision trees and Naives- bayes	PYTHON	up to 96.91%	0.33%	Keles, M. Kaya
RANDOM FOREST	WEKA	92.2 %		KELES et al.
ADABOOST	WEKA	97.5 %		Delen at al.
FastCorrelation-	WEKA	96.1%	0.0404	Khourdifi et al.

Based Filter with SVM, Random Forest, Naive Bayes, K-NN and MLP				
DECISION TREE	WEKA TOOL	96.3 %		Mohana,et. Al
SVM	Spyder	92.7%		Shravya at al.
Naïve Bayes	WEKA	96.3 %		Bellaachia et. al

Table 1: Table of comparison of top different models on Breast Cancer Detection

PROPOSED METHODOLOGY

Proposed Methodology is as shown in Figure 2.

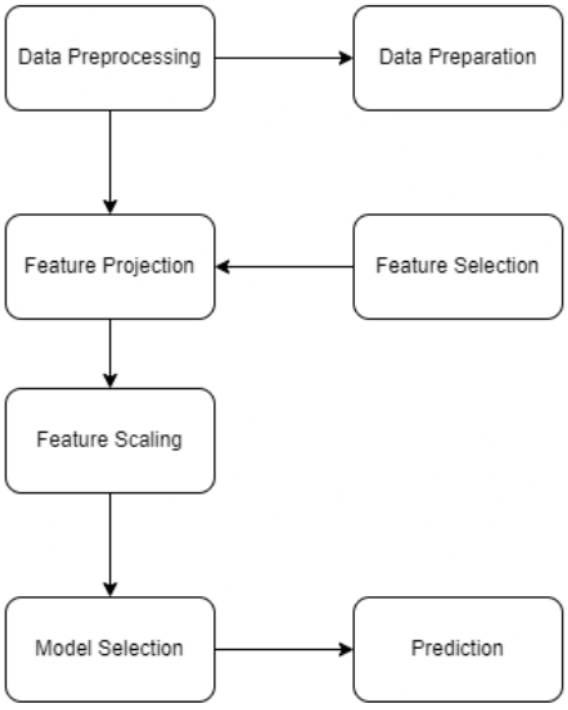


FIGURE 2- Phases of Machine Learning consists of seven phases, the phases are elaborated as given below:

1. **Phase 1 - Pre-Processing Data**

The fundamental stage we do is to accumulate the data that we are enthusiastic about social occasion for pre-dealing with and to apply request and Relapse procedures. Data pre-taking care of is a data mining technique that incorporates changing rough data into a legitimate design. Genuine data is often deficient, clashing, and without sure to contain various bungles. Data pre-taking care of is an exhibited procedure for settling such issues. Data pre-dealing with prepares unrefined data for extra taking care of. For pre-taking care of we have used standardization procedure to pre-process the UCI dataset. This movement is indispensable in light of the fact that the quality and measure of data that you aggregate will clearly conclude how extraordinary your farsighted model can be. For this present circumstance we assemble the Bosom Disease tests which are Harmless and Threatening. This will be our planning data.

2. **Phase 2 - Data Preparation**

Data Planning, where we load our data into a sensible spot and set it up for use in our AI getting ready. We'll at first collect all of our data, and a while later randomize the mentioning.

3. **Phase 3 - Features Selection**

In AI and estimations, feature decision, generally called variable assurance, quality assurance, is the course of assurance a subset of relevant components for use in model turn of events. Data Record and Component Choice Bosom Disease Wisconsin (Symptomatic):- Informational file from Kaggle vault and out of 31 limits we have picked around 8-9 limits. Our objective limit is chest illness finding - hazardous or innocuous. We have involved Covering Strategy for Component Choice. The huge components found by the audit are: curved concentrates generally clearly horrendous, Region generally horrendous, Region se, Surface generally extremely awful, Surface mean, Perfection generally dreadful, Perfection mean, Span mean, Evenness mean.

We have involved Covering Technique for Component Determination. The significant highlights found by the review are:

1. Inward focuses most terrible
2. Region most horrendously awful
3. Region se
4. Surface most horrendously awful
5. Surface mean
6. Perfection most obviously terrible
7. Perfection mean
8. Span mean

9. Evenness implies.

Attribute Information:
ID number 2) Diagnosis (M = malignant, B = benign) 3–32)

4. Phase 4 - Feature Projection

Incorporate projection is change of high-layered space data to a lower layered space (with few credits). Both immediate and nonlinear reduction methodology can be used according to the sort of associations among the features in the dataset.

5. Phase 5 - Feature Scaling

Most of the times, your dataset will contain incorporates outstandingly contrasting in sizes, units and reach. However, since, by far most of the AI estimations use Euclidian distance between two information of interest in their computations. We truly need to convey all components to comparable level of sizes. This can achieved by scale.

6. Phase 6 - Model Selection

Overseen learning is the method wherein the machine is ready on the data which the data and result are generally around named. The model can learn on the arrangement data and can deal with the future data to anticipate result. They are accumulated to Relapse and Arrangement techniques. A backslide issue is the place where the result is a certified or persevering worth, for instance, "pay" or "weight". A gathering issue is the place where the result is a characterization like filtering messages spam" or "not spam". Independent Learning: Solo learning is offering information to the machine that is neither arranged nor named and allowing the computation to inspect the given information without giving any headings. In performance acquiring computation the machine is ready from the data which isn't named or gathered making the estimation to work without fitting headings. In our dataset we have the outcome variable or Ward variable for instance Y having only two plans of values, either M (Defame) or B (Harmless). So Order computation of overseen learning is applied on it. We have picked three novel kinds of portrayal computations in AI. We can use somewhat straight model, which is a fundamental.

7. Phase 7 - Prediction

Computer based intelligence is using data to resolve questions. So Forecast, or derivation, is the movement where we get to answer specific requests. This is the sign of this work, where the value of AI is real.

Effectiveness

In this part, we survey the sufficiency of all classifiers to the extent that opportunity to develop the model, precisely described cases, wrongly organized models and precision.

The ROC space is portrayed with real up-sides and counterfeit up-sides as the x and y organizes, independently. ROC twist summarizes the display across each possible edge. The corner to corner of the ROC graph can be translated as inconsistent hypothesizing, and portrayal models that fall under the

inclining are seen as more deplorable than sporadic estimating.

- You have not implemented Naïve however below figure is Naïve Bayes.

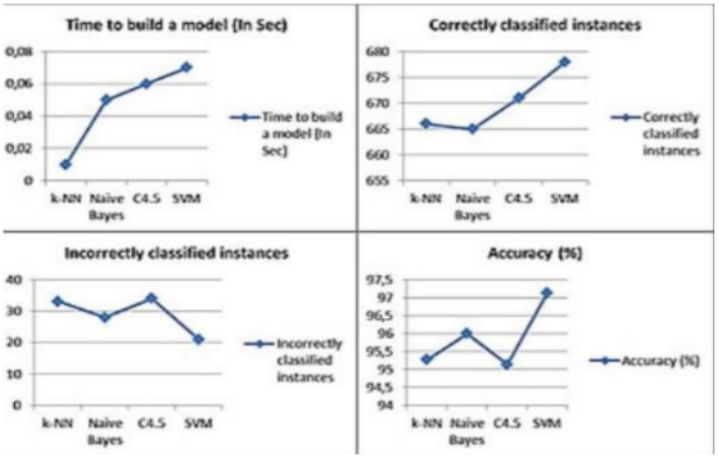


Figure 2: Naïve Bayes

Following making the expected model, we can now inspect results got. SVM and C4.5 got the most important worth (97%) of TP for innocuous class yet KNN precisely predicts 97 % of model that has a spot with undermining class.

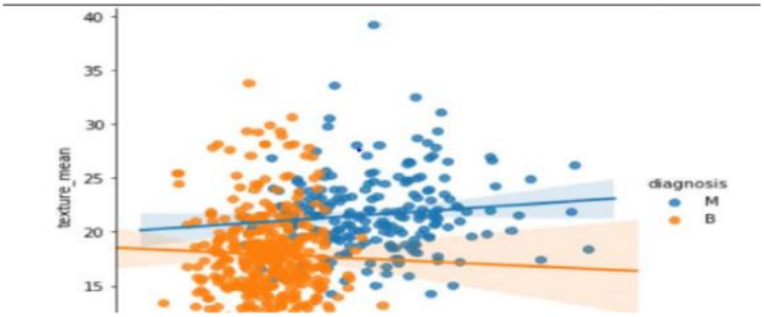


FIGURE 3- Comparison of compactness mean vs smoothness mean where orange represents Benign and Blue represents Malignant

FUTURE WORK

The assessment of the results implies that the blend of mind-boggling data close by different gathering, feature assurance and dimensionality decline techniques can give great gadgets to allowance around here. Further assessment in this field should be finished for the better presentation of the gathering techniques with the objective that it can expect on extra factors. We are proposing how to parametrize our request techniques subsequently to achieve high accuracy. We are exploring various datasets and how further AI computations can be used to portray Bosom Disease. We really want to decrease the misstep rates with most outrageous accuracy.

CONCLUSION

We can see that SVM takes around 0.07 s to create model not the least bit like k-NN takes basically 0.01 s. It might be gotten a handle on by the way that k-NN is a lazy understudy and doesn't do a ton during planning process not by any stretch like others classifiers that structure the models. In other hand, the precision got by SVM (97.13%) is better than the accuracy got by C4.5, Gullible Bayes and k-NN that have an accuracy that contrasts between 95.12 % and 95.28 %. It can moreover be really seen that SVM has the most significant worth of precisely requested models and the lower worth of incorrectly organized events than various classifiers.

Following making the expected model, we can now look at results got in evaluating efficiency of our estimations. SVM and C4.5 got the most raised regard (97 %) of TP for innocuous class yet k-NN precisely predicts 97% of model that have a spot with compromising class. The FP rate is lower while using SVM classifiers (0.03 for innocuous class and 0.02 for compromising class), and subsequently various estimations follow: k-NN, C4.5 and NB. From these results, we can understand the motivation behind why SVM has beated various classifiers.

In synopsis, SVM had the option to show its power as far as adequacy and proficiency in view of precision and review.

APPENDIX/SOURCE CODE

LIBRARY IMPORTS

MANIPULATION DATA

```
import pandas as pd
import numpy as np
```

#VISUALIATION DATA

```
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib
```

```
import plotly.graph_objects as go
import plotly.express as px

#DEFAULT THEME
sns.set(context='notebook', style='darkgrid', palette='colorblind', font='sans-serif', font_scale=1, rc=None)
matplotlib.rcParams['figure.figsize'] =[8,8]
matplotlib.rcParams.update({'font.size': 15})
matplotlib.rcParams['font.family'] = 'sans-serif'
```

SOURCE CODE

1. LOAD AND ANALYS DATA

```
train = pd.read_csv('../input/breastcancernewdatasettds/data.csv')
train.head()
train.info()
```

like we see all our feautres are numirical values exept the target value *diagnosis* (M = malignant, B = benign)

```
train.shape
```

we had 569 Rows and 33 columns (small data)

```
train.describe(include='all')
```

	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	0.3001	0.14710
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430

5 rows × 33 columns

A) FINDING MISSING VALUES

```
missing_values=train.isnull().sum()
percent_missing = train.isnull().sum()/train.shape[0]*100
```

```
  value = {
     'missing_values ':missing_values,
     'percent_missing %':percent_missing
  }
  frame=pd.DataFrame(value)
frame
```

like we c our data is clean exept the last columns that is empty so we gonna drop it

```
  train=train.drop('Unnamed: 32',axis=1)
train.head()
```

2. FEAUTRES SELECTION

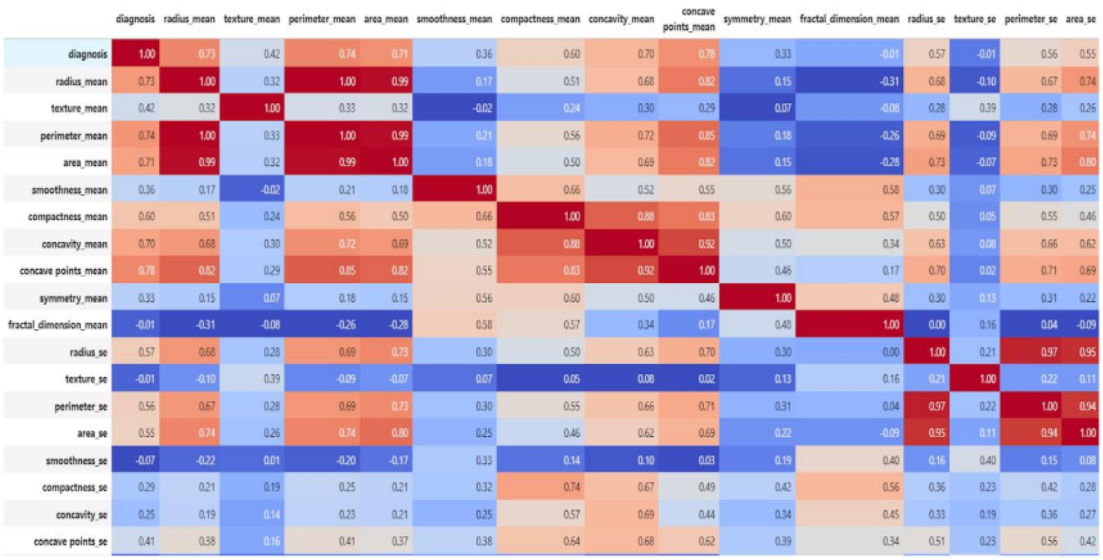
```
  # drop the id columns
  train=train.drop('id',axis=1)

  # transformation of type of the target value to numerical
  from sklearn import preprocessing
  le = preprocessing.LabelEncoder()
  train.diagnosis = le.fit_transform(train.diagnosis)
  train.diagnosis
```

Diagnosis

1. M = malignant ==> 1
2. B = benign ==> 0

A) CORRELATION MAP

```
  train.corr().style.background_gradient(cmap='coolwarm').set_precision(2)

  # drop this columns
  train=train.drop(['fractal_dimension_mean','texture_se','smoothness_se','symmetry_se','fractal_dimension
     _se'],axis=1)
```

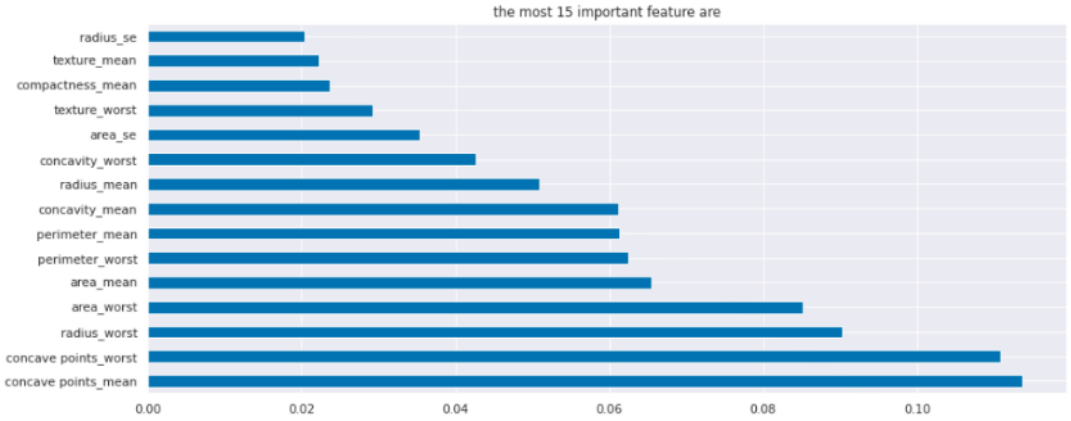


	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean	radius_se	texture_se	perimeter_se	area_se
diagnosis	1.00	0.73	0.42	0.74	0.71	0.36	0.60	0.70	0.78	0.33	-0.01	0.57	-0.01	0.56	0.55
radius_mean	0.73	1.00	0.32	1.00	0.99	0.17	0.51	0.68	0.82	0.15	-0.31	0.68	-0.10	0.67	0.74
texture_mean	0.42	0.32	1.00	0.33	0.32	-0.02	0.24	0.30	0.29	0.07	-0.08	0.28	0.39	0.28	0.26
perimeter_mean	0.74	1.00	0.33	1.00	0.99	0.21	0.56	0.72	0.85	0.18	-0.26	0.69	-0.09	0.69	0.74
area_mean	0.71	0.99	0.32	0.99	1.00	0.18	0.50	0.69	0.82	0.15	-0.28	0.73	-0.07	0.73	0.80
smoothness_mean	0.36	0.17	-0.02	0.21	0.18	1.00	0.66	0.52	0.55	0.56	0.58	0.30	0.07	0.30	0.25
compactness_mean	0.60	0.51	0.24	0.56	0.50	0.66	1.00	0.88	0.83	0.60	0.57	0.50	0.05	0.55	0.46
concavity_mean	0.70	0.68	0.30	0.72	0.69	0.52	0.88	1.00	0.92	0.50	0.34	0.63	0.08	0.66	0.62
concave points_mean	0.78	0.82	0.29	0.85	0.82	0.55	0.83	0.92	1.00	0.46	0.17	0.70	0.02	0.71	0.69
symmetry_mean	0.33	0.15	0.07	0.18	0.15	0.56	0.60	0.50	0.46	1.00	0.48	0.30	0.13	0.31	0.22
fractal_dimension_mean	-0.01	-0.31	-0.08	-0.26	-0.28	0.58	0.57	0.34	0.17	0.48	1.00	0.00	0.16	0.04	-0.09
radius_se	0.57	0.68	0.28	0.69	0.73	0.30	0.50	0.63	0.70	0.30	0.00	1.00	0.21	0.97	0.95
texture_se	-0.01	-0.10	0.39	-0.09	-0.07	0.07	0.05	0.08	0.02	0.13	0.16	0.21	1.00	0.22	0.11
perimeter_se	0.56	0.67	0.28	0.69	0.73	0.30	0.55	0.66	0.71	0.31	0.04	0.97	0.22	1.00	0.94
area_se	0.55	0.74	0.26	0.74	0.80	0.25	0.46	0.62	0.69	0.22	-0.09	0.95	0.11	0.94	1.00
smoothness_se	-0.07	-0.22	0.01	-0.20	-0.17	0.33	0.14	0.10	0.03	0.19	0.40	0.16	0.40	0.15	0.08
compactness_se	0.29	0.21	0.19	0.25	0.21	0.32	0.74	0.67	0.49	0.42	0.56	0.36	0.23	0.42	0.28
concavity_se	0.25	0.19	0.14	0.23	0.21	0.25	0.57	0.69	0.44	0.34	0.45	0.33	0.19	0.36	0.27
concave points_se	0.41	0.38	0.16	0.41	0.37	0.38	0.64	0.68	0.62	0.39	0.34	0.51	0.23	0.56	0.42

Feature Selection

```
plt.rcParams['figure.figsize']=15,6
sns.set_style("darkgrid")

x = train.drop('diagnosis',axis=1)
y = train.diagnosis

from sklearn.ensemble import ExtraTreesClassifier

model = ExtraTreesClassifier()
model.fit(x,y)
print(model.feature_importances_)
feat_importances = pd.Series(model.feature_importances_, index=x.columns)
feat_importances.nlargest(15).plot(kind='barh')
plt.title('the most 15 important feature are')
plt.show()

train.columns

train=train.drop(['texture_mean','smoothness_mean','compactness_mean','symmetry_mean','perimeter_se',
```

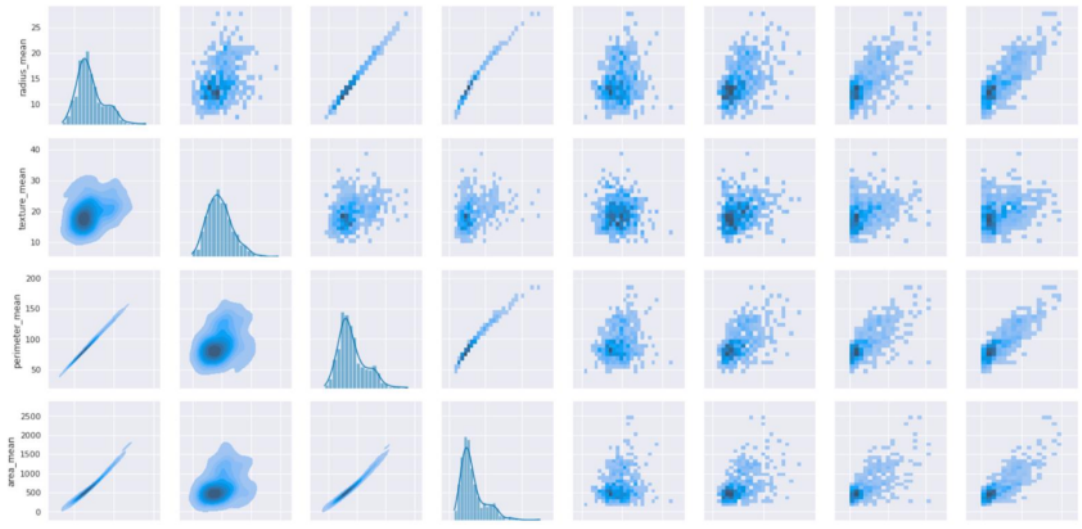
'compactness_se','concavity_se','concave
points_se','smoothness_worst','symmetry_worst','fractal_dimension_worst'],axis=1)



3. DATA VIZUALISATION

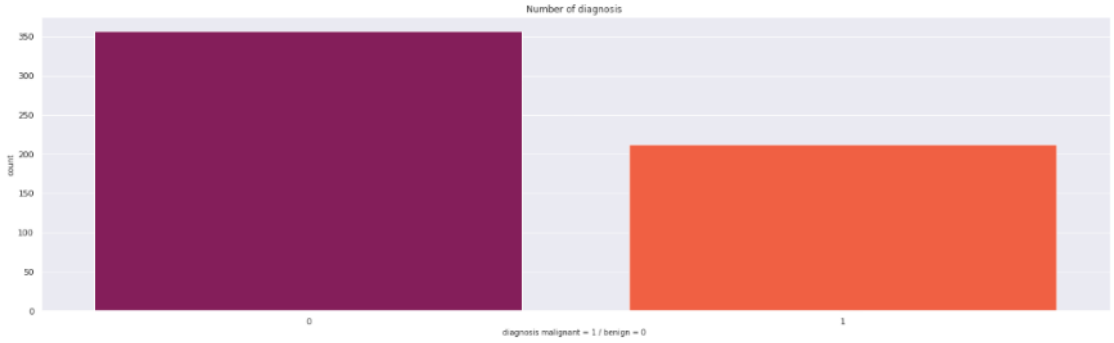
Plotting many distributions

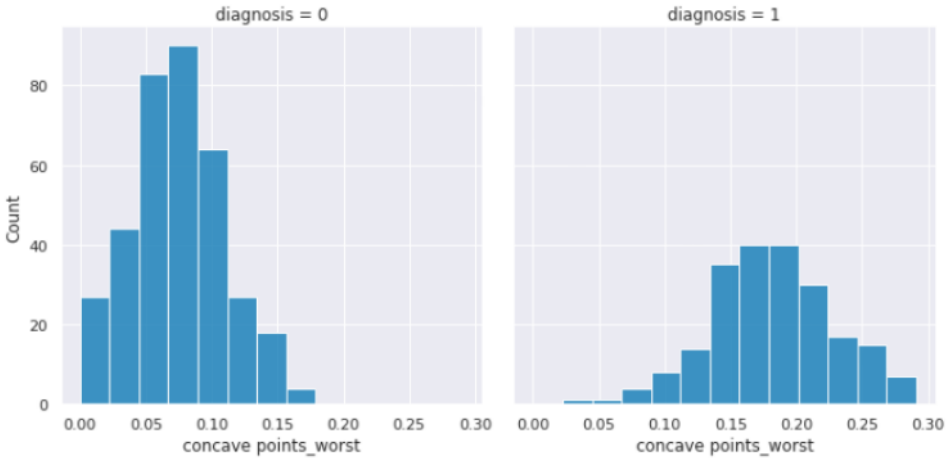
```
g = sns.PairGrid(x)
g.map_upper(sns.histplot)
g.map_lower(sns.kdeplot, fill=True)
g.map_diag(sns.histplot, kde=True)
```



A) DIAGNOSIS

```
plt.rcParams['figure.figsize']=25,7
sns.set_style("darkgrid")
ax = sns.countplot(x=train.diagnosis , palette = "rocket", saturation =1.5)
plt.xlabel("diagnosis malignant = 1 / benign = 0 ", fontsize = 10 )
plt.ylabel("count", fontsize = 10)
plt.title('Number of diagnosis ')
```





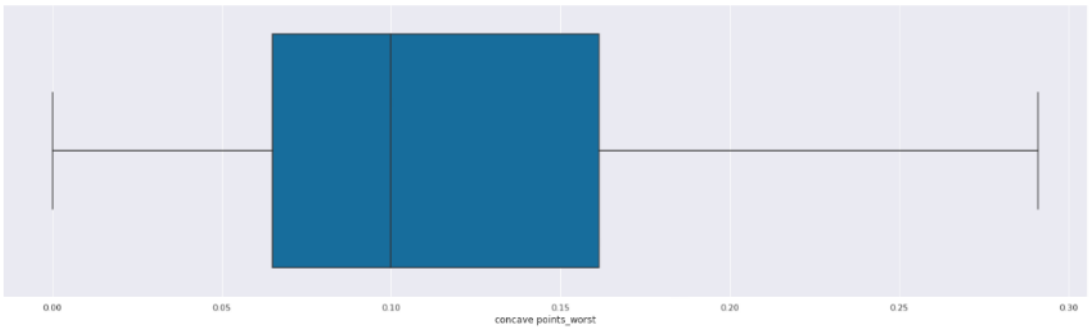
B) CONCAVE POINTS_WORST

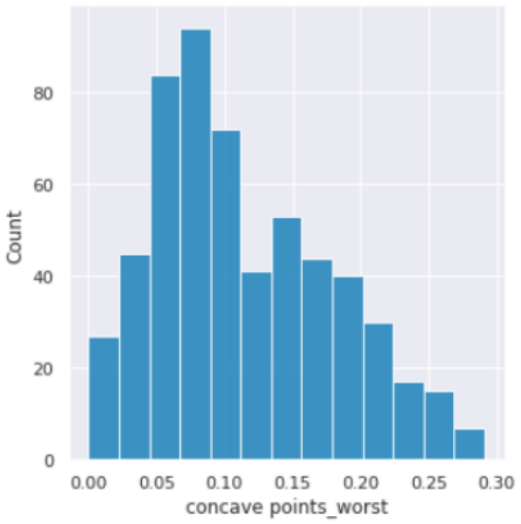
sns.boxplot(x=train['concave points_worst'])

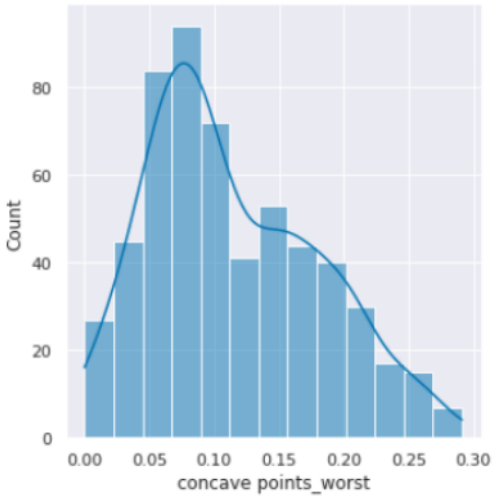
sns.displot(train, x='concave points_worst')

sns.displot(train, x="concave points_worst", kde=True)

sns.displot(train, x="concave points_worst", col="diagnosis", multiple="dodge")\





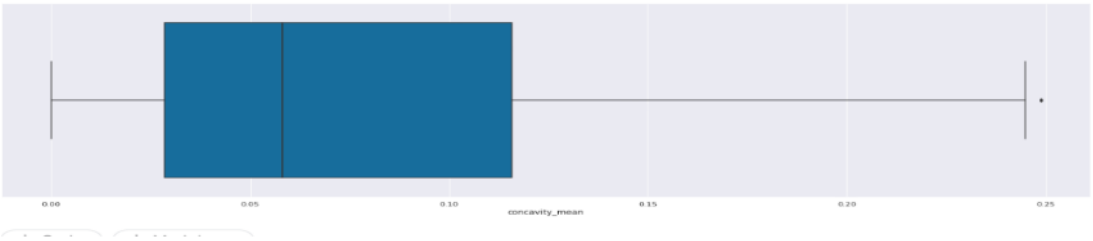


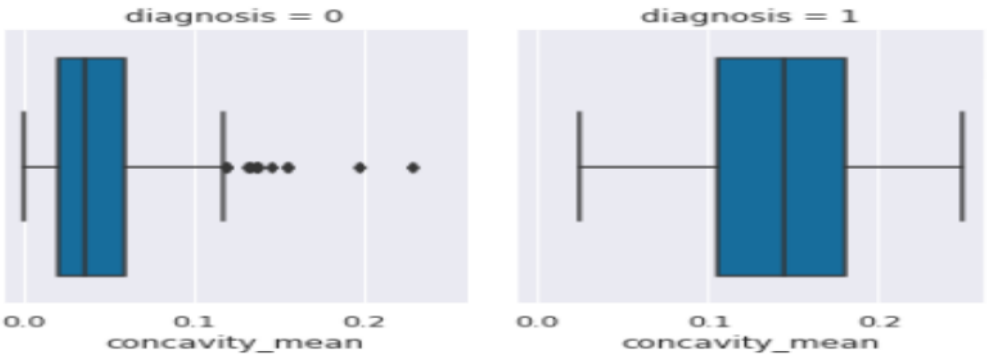
C) CONCAVITY_MEAN

sns.boxplot(x=train['concavity_mean'])

We can see there are some outliers. Let's remove them (0.3- 0.5)

```
outlier=train[train['concavity_mean']>=0.25]
outlier

train = train[train['concavity_mean']<0.25]
train

sns.boxplot(x=train['concavity_mean'])

sns.catplot(x="concavity_mean",
            col="diagnosis",
            data=train, kind="box",
            height=4, aspect=.7);

sns.boxplot(x=train['perimeter_worst'])

outlier=train[train['perimeter_worst']>=165]
outlier

sns.boxplot(x=train['perimeter_worst'])
```





4. MACHINE LEARNING APPLICATION

A) SPLIT DATA

```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score

x=train.drop('diagnosis',axis=1)
y=train.diagnosis

print(x.shape)
print(y.shape)

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

B) FEATURE SCALING

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

A) LOGISTIC REGRESSION

```
# Making Confusion Matrix and calculating accuracy score
```

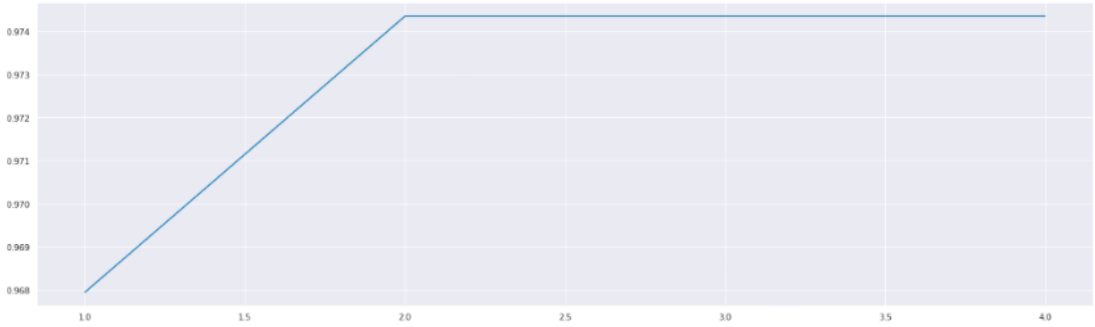
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score

model = LogisticRegression()

#Fit the model
model.fit(x_train, y_train)
y_pred = model.predict(x_test)

mylist = []
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
# accuracy score
acc_logreg = accuracy_score(y_test, y_pred)

mylist.append(acc_logreg)
print(cm)
print(acc_logreg,'%')
```

```
[[109   0]
[  3  44]]
0.9807692307692307 %
```

B) KNN

```
# Finding the optimum number of neighbors

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score

list1 = []
for neighbors in range(1,5):
    classifier = KNeighborsClassifier(n_neighbors=neighbors, metric='minkowski')
    classifier.fit(x_train, y_train)
    y_pred = classifier.predict(x_test)
    list1.append(accuracy_score(y_test,y_pred))
plt.plot(list(range(1,5)), list1)
plt.show()

# Training the K Nearest Neighbor Classifier on the Training set

classifier = KNeighborsClassifier(n_neighbors=3)
```

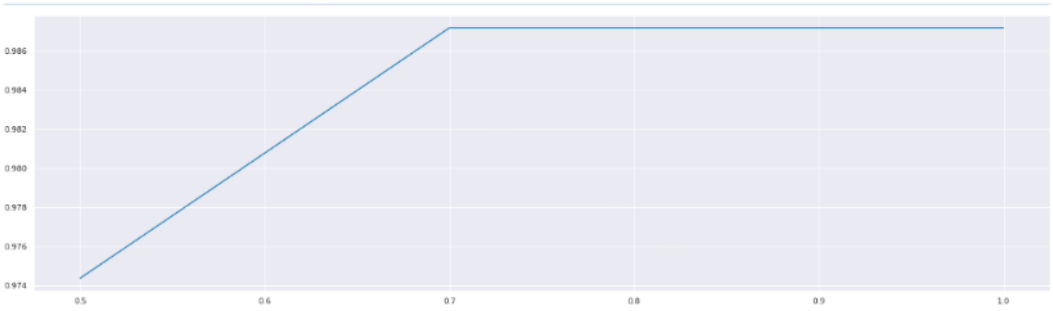
```
classifier.fit(x_train, y_train)

# Predicting the Test set results

y_pred = classifier.predict(x_test)
print(y_pred)

# Making the confusion matrix and calculating accuracy score

from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
acc_knn = accuracy_score(y_test, y_pred)
mylist.append(acc_knn)
print(cm)
print(acc_knn)
```



C) SUPPORT VECTOR MACHINES

```
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, accuracy_score
list1 = []
for c in [0.5,0.6,0.7,0.8,0.9,1.0]:
    classifier = SVC(C = c, random_state=0, kernel = 'rbf')
    classifier.fit(x_train, y_train)
    y_pred = classifier.predict(x_test)
    list1.append(accuracy_score(y_test,y_pred))
plt.plot([0.5,0.6,0.7,0.8,0.9,1.0], list1)
plt.show()

# Training the Support Vector Classifier on the Training set

from sklearn.svm import SVC
classifier = SVC(C = 0.9, random_state=0, kernel = 'rbf')
classifier.fit(x_train, y_train)
```

```
# Predicting the test set results

y_pred = classifier.predict(x_test)
print(y_pred)

# Making the confusion matrix and calculating accuracy score

from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
acc_svc = accuracy_score(y_test, y_pred)
print(cm)
print(acc_svc,'%')
mylist.append(acc_svc)
```



Calculating accuracy

[[108 1]

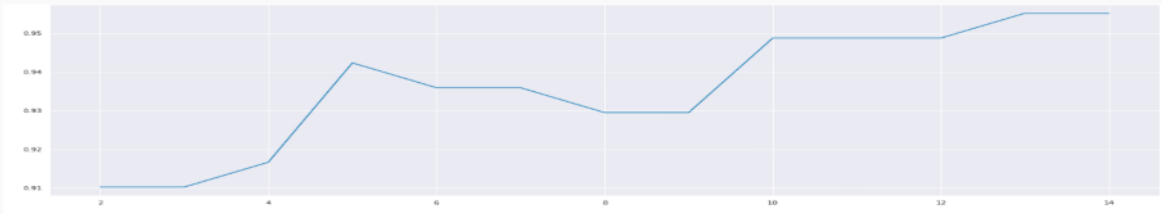
[1 46]]

0.9871794871794872 %

D) DECISIONTREECLASSIFIER

```
# Finding the optimum number of max_leaf_nodes

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
list1 = []
for leaves in range(2,15):
    classifier = DecisionTreeClassifier(max_leaf_nodes = leaves, random_state=0, criterion='entropy')
    classifier.fit(x_train, y_train)
    y_pred = classifier.predict(x_test)
```

```
    list1.append(accuracy_score(y_test,y_pred))
#print(mylist)
plt.plot(list(range(2,15)), list1)
plt.show()

# Training the Decision Tree Classifier on the Training set

classifier = DecisionTreeClassifier(max_leaf_nodes = 5, random_state=0, criterion='entropy')
classifier.fit(x_train, y_train)

#Predicting the test set results

y_pred = classifier.predict(x_test)
print(y_pred)

# Making the confusion matrix and calculating accuracy score

from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
acc_decisiontree = accuracy_score(y_test, y_pred)
print(cm)
print(acc_decisiontree)
mylist.append(acc_decisiontree)
```

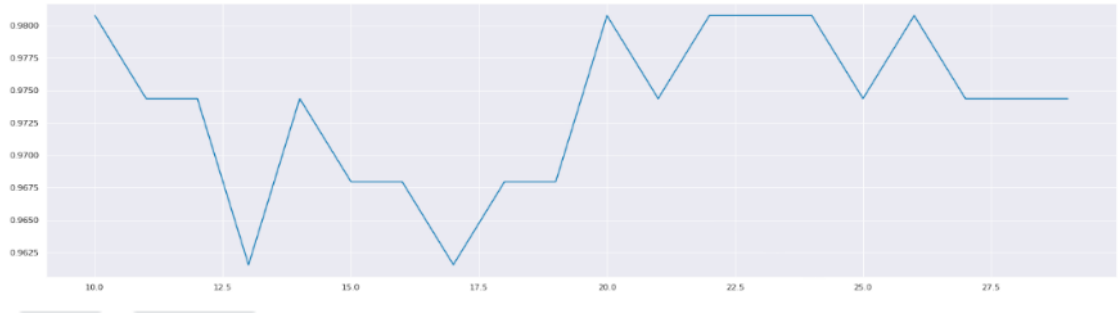


```
[[102   7]
 [  2  45]]
0.9423076923076923
```

E) RANDOM FOREST CLASSIFCATION

```
#Finding the optimum number of n_estimators


from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
list1 = []
for estimators in range(10,30):
    classifier = RandomForestClassifier(n_estimators = estimators, random_state=0,
      criterion='entropy')
    classifier.fit(x_train, y_train)
    y_pred = classifier.predict(x_test)
    list1.append(accuracy_score(y_test,y_pred))
#print(mylist)
plt.plot(list(range(10,30)), list1)
plt.show()

# Training the RandomForest Classifier on the Training set

from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 15, criterion='entropy', random_state=0)
classifier.fit(x_train,y_train)

# Predicting the test set results

y_pred = classifier.predict(x_test)
print(y_pred)
```




```
[[106   3]
 [  2  45]]
```

0.967948717948718

F) ANN (NEURAL NETWORK)

```
np.random.seed(0)
import tensorflow as tf

# Initialising the ANN

ann = tf.keras.models.Sequential()

# Adding the input layer and the first hidden layer

ann.add(tf.keras.layers.Dense(units = 7, activation = 'relu'))

# Adding the second hidden layer

ann.add(tf.keras.layers.Dense(units = 7, activation = 'relu'))

# Adding the third hidden layer

ann.add(tf.keras.layers.Dense(units = 7, activation = 'relu'))

# Adding the fourth hidden layer

ann.add(tf.keras.layers.Dense(units = 7, activation = 'relu'))

# Adding the output layer

ann.add(tf.keras.layers.Dense(units = 1, activation = 'sigmoid'))
```

```
        # Compiling the ANN

        ann.compile(optimizer = 'adam', loss = 'binary_crossentropy' , metrics = ['accuracy'] )

        # Training the ANN on the training set

ann.fit(x_train, y_train, batch_size = 16, epochs = 100)

        # Predicting the test set results

        y_pred = ann.predict(x_test)
        y_pred = (y_pred > 0.9)
np.set_printoptions()
```

```
# Making the confusion matrix, calculating accuracy_score

from sklearn.metrics import confusion_matrix, accuracy_score

# confusion matrix
cm = confusion_matrix(y_test,y_pred)
print("Confusion Matrix")
print(cm)
print()

# accuracy
ac_ann = accuracy_score(y_test,y_pred)
print("Accuracy")
print(ac_ann)
mylist.append(ac_ann)
```

accuracy: 0.9890
Confusion Matrix

[[109 0]

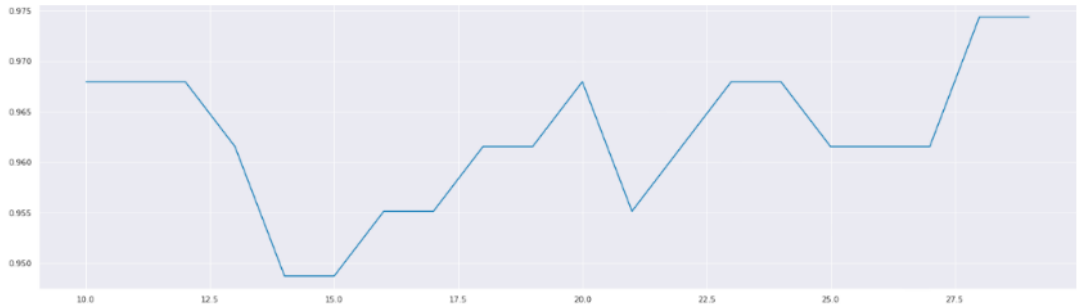
[2 45]]

Accuracy

0.9871794871794872

G) XGBOOST

```
from xgboost import XGBClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
list1 = []
for estimators in range(10,30,1):
    classifier = XGBClassifier(n_estimators = estimators, max_depth=12, subsample=0.7)
    classifier.fit(x_train, y_train)
    y_pred = classifier.predict(x_test)
    list1.append(accuracy_score(y_test,y_pred))
#print(mylist)
plt.plot(list(range(10,30,1)), list1)
plt.show()

from xgboost import XGBClassifier
classifier = XGBClassifier(n_estimators = 15, max_depth=12, subsample=0.7)
classifier.fit(x_train,y_train)

y_pred = classifier.predict(x_test)
print(y_pred)

# Making the confusion matrix and calculating the accuracy score

from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
ac_xgboost = accuracy_score(y_test, y_pred)
mylist.append(ac_xgboost)
print(cm)
print(ac_xgboost)
```




```
[[106   3]
 [  5  42]]
0.9487179487179487
```

H) CATBOOST

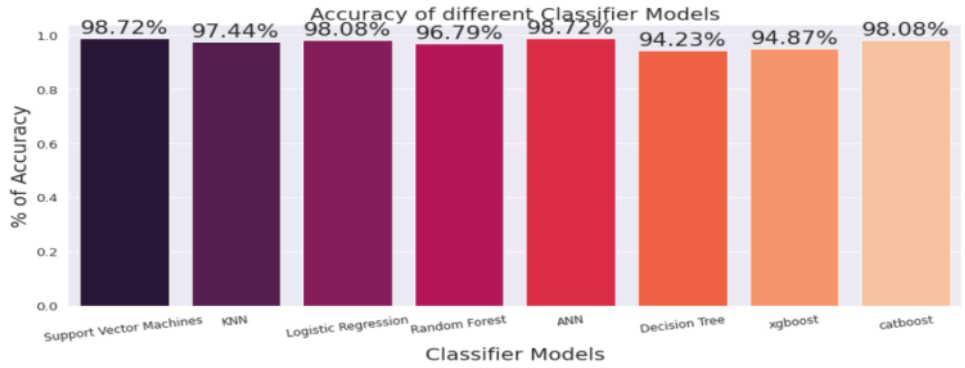
```
  from catboost import CatBoostClassifier
  classifier = CatBoostClassifier()
  classifier.fit(x_train, y_train)

  y_pred = classifier.predict(x_test)
  print(y_pred)

# Making the confusion matrix and calculating the accuracy score

from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
ac_catboost = accuracy_score(y_test, y_pred)
mylist.append(ac_catboost)
print(cm)
print(ac_catboost)

models = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression',
              'Random Forest', 'ANN',
              'Decision Tree','xgboost','catboost'],
    'Score': [acc_svc, acc_knn, acc_logreg,
              acc_randomforest, ac_ann, acc_decisiontree,ac_xgboost,ac_catboost
              ]})
models.sort_values(by='Score', ascending=False)
```

```
 plt.rcParams['figure.figsize']=15,6
sns.set_style("darkgrid")
ax = sns.barplot(x=models.Model, y=models.Score, palette = "rocket", saturation =1.5)
plt.xlabel("Classifier Models", fontsize = 20 )
plt.ylabel("% of Accuracy", fontsize = 20)
plt.title("Accuracy of different Classifier Models", fontsize = 20)
plt.xticks(fontsize = 12, horizontalalignment = 'center', rotation = 8)
plt.yticks(fontsize = 13)
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate(f'{height:.2%}', (x + width/2, y + height*1.02), ha='center', fontsize = 'x-large')
plt.show()
```

```
[[107   2]
 [  1  46]]
0.9807692307692307
```

COMPARISSION

	Model	Score
0	Support Vector Machines	0.987179
4	ANN	0.987179
2	Logistic Regression	0.980769
7	catboost	0.980769
1	KNN	0.974359
3	Random Forest	0.967949
6	xgboost	0.948718
5	Decision Tree	0.942308



REFRENCES

[1] Wang, D. Zhang and Y. H. Huang "Breast Cancer Prediction Using Machine Learning" (2018), Vol. 66, NO. 7.

[2] B. Akbugday, "Classification of Breast Cancer Data Using Machine Learning Algorithms," 2019 Medical Technologies Congress (TIPTEKNO), Izmir, Turkey, 2019, pp. 1-4.

[3] Keles, M. Kaya, "Breast Cancer Prediction and Detection Using Data Mining Classification Algorithms: A Comparative Study." Tehnicki Vjesnik - Technical Gazette, vol. 26, no. 1, 2019, p. 149+.
[4] V. Chaurasia and S. Pal, "Data Mining Techniques: To Predict and Resolve Breast Cancer Survivability", IJCSMC, Vol. 3, Issue. 1, January 2014, pg.10 – 22.
[5]] Delen, D.; Walker, G.; Kadam, A. Predicting breast cancer survivability: A comparison of three data mining methods. Artif. Intell. Med. 2005, 34, 113–127.
[6] R. K. Kavitha1, D. D. Rangasamy, "Breast Cancer Survivability Using Adaptive Voting Ensemble Machine Learning Algorithm Adaboost and CART Algorithm" Volume 3, Special Issue 1, February 2014
[7] P. Sinthia, R. Devi, S. Gayathri and R. Sivasankari, "Breast Cancer detection using PCPCET and ADEWNN", CIEEE' 17, p.63-65
[8] Vikas Chaurasia and S.Pal, "Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis" (FAMS 2016) 83 (2016) 1064 – 1069
[9] N. Khuriwal, N. Mishra. "A Review on Breast Cancer Diagnosis in Mammography Images Using Deep Learning Techniques", (2018), Vol. 1, No. 1.
[10] Y. Khourdifi and M. Bahaj, "Feature Selection with Fast Correlation-Based Filter for Breast Cancer Prediction and Classification Using Machine Learning Algorithms," 2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), Rabat, Morocco, 2018, pp. 1-6.
[11] R. M. Mohana, R. Delshi Howsalya Devi, Anita Bai, "Lung Cancer Detection using Nearest Neighbour Classifier", International Journal of Recent Technology and Engineering (IJRTE), Volume-8, Issue-2S11, September 2019.

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30