VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

Predicting Breast Cancer

using machine learning

Machine Learning??

- **Machine learning** (**ML**) is the scientific
  study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.
- Machine Learning can be simply termed to be experiments that a computer does on data sets for further predictions.

Abstract

- In this project, a performance comparison between different machine learning algorithms such as, Support Vector Machine (SVM), Decision Tree, Naive Bayes (NB) and k Nearest Neighbours (k-NN) on the Breast_cancer dataset is conducted.

- The main objective is to assess the correctness in classifying data with respect to efficiency and effectiveness of each algorithm in terms of accuracy, sensitivity and specificity. Experimental results show that kNN gives the highest accuracy (97.13%) with lowest error rate. All experiments are executed using Jupyter Notebook with Anaconda 3(64-bit) as the Base.

Line Of Action:

- Importing Libraries
- Importing Datasets
- Split dataset into attributes using iloc
- Splitting dataset into train and test dataset
- Training and Prediction
- Evaluating the Algorithm

Importing Libraries and Dataset

```
import os
os.chdir('C:/Users/rajashekarreddy/Desktop/data sets')
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
data=pd.read_csv("Breast_cancer.csv")
data
```

[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	c point
0	842302	M	17.990	10.38	122.80	1001.0	0.11840	0.27760	0.300100	0
1	842517	M	20.570	17.77	132.90	1326.0	0.08474	0.07864	0.086900	0
2	84300903	M	19.690	21.25	130.00	1203.0	0.10960	0.15990	0.197400	0

ERROR 404:
NO NULL VALUES FOUND

Correlation

```
In [12]: data.corr()
```

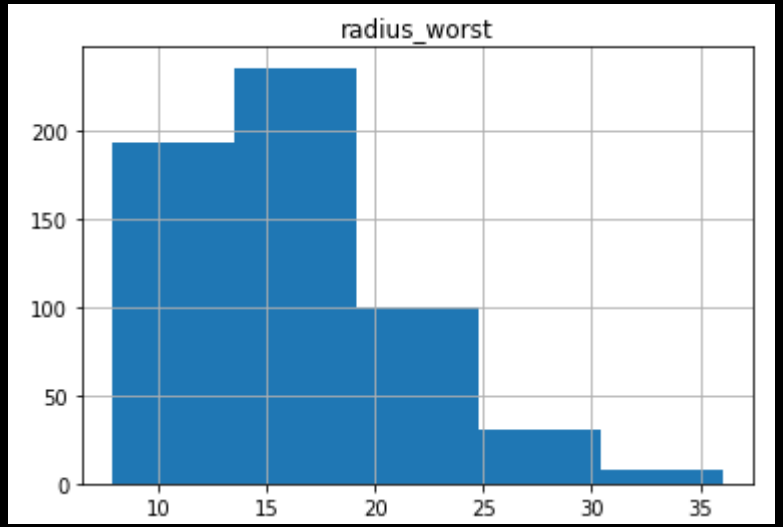
Out[12]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	poin
diagnosis	1.000000	0.730029	0.415185	0.742636	0.708984	0.358560	0.596534	0.696360	
radius_mean	0.730029	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	
texture_mean	0.415185	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	
perimeter_mean	0.742636	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	
area_mean	0.708984	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	
smoothness_mean	0.358560	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	
compactness_mean	0.596534	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	
concavity_mean	0.696360	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	
concave points_mean	0.776614	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	
symmetry_mean	0.330499	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	
fractal_dimension_mean	-0.012838	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	
radius_se	0.567134	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925	
texture_se	-0.008303	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218	
perimeter_se	0.556141	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391	
area_se	0.548236	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427	
smoothness_se	-0.067016	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564	

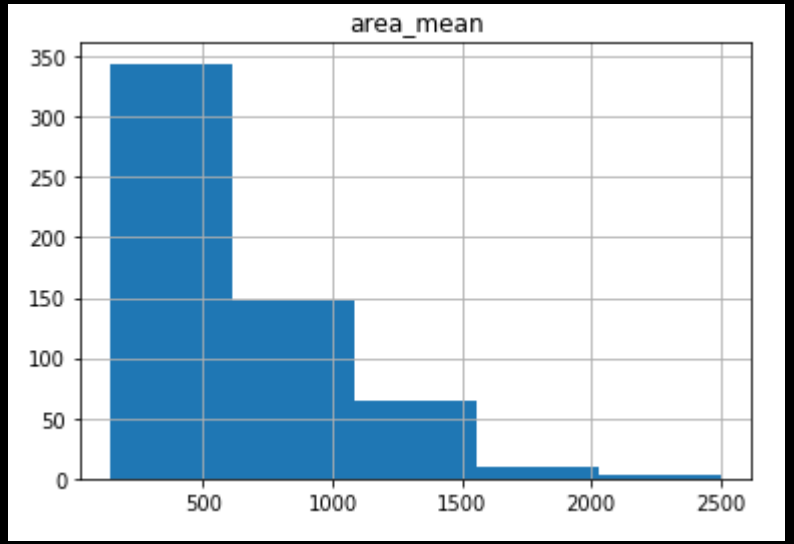
DATA VISUALISATION

Histograms:-

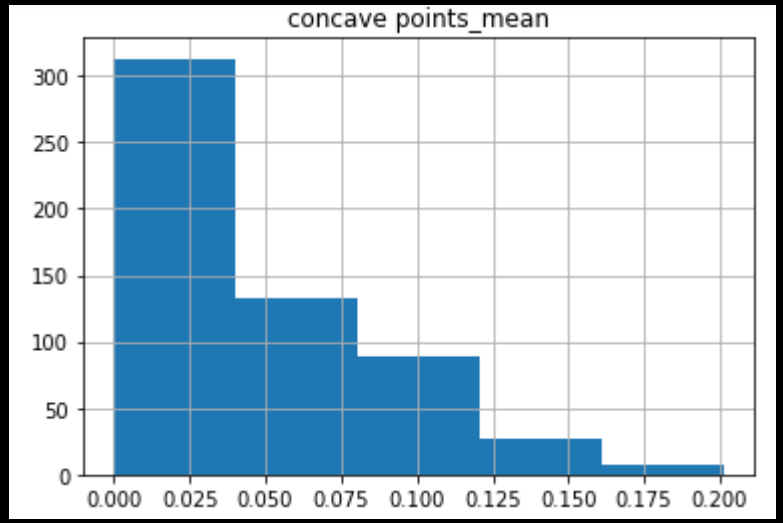
```
data.hist('radius_worst',bins=5)
```



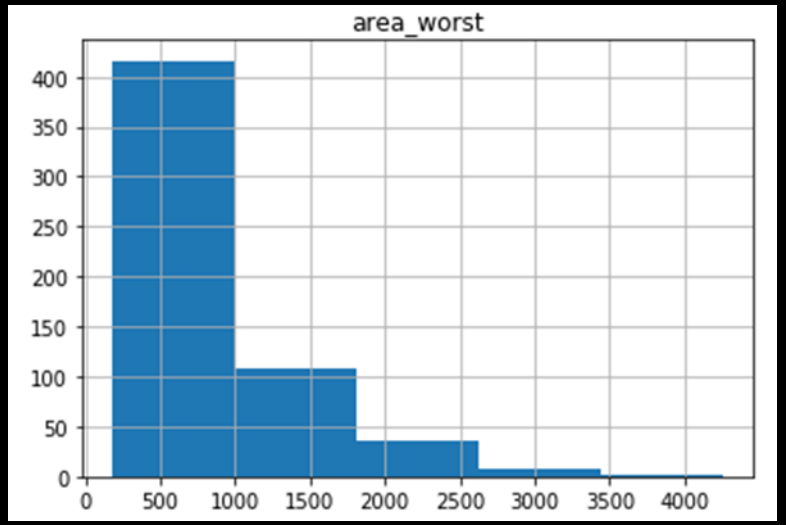

```
,data.hist('area_mean',bins=5)
```



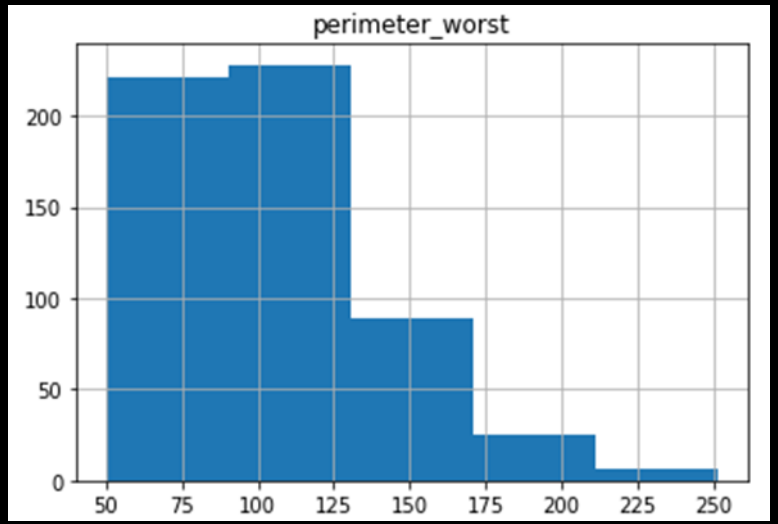
```
,data.hist('concave points_mean',bins=5)
```



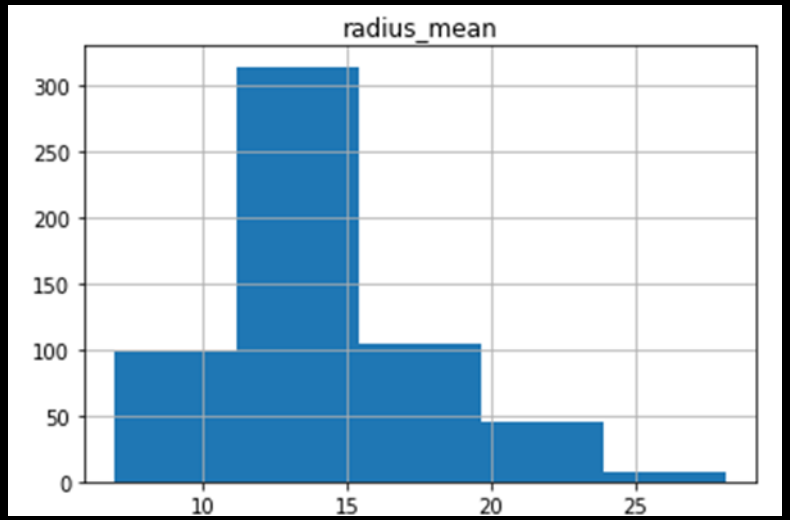

```
data.hist('area_worst',bins=5)
```



```
data.hist('perimeter_worst',bins=5)
```

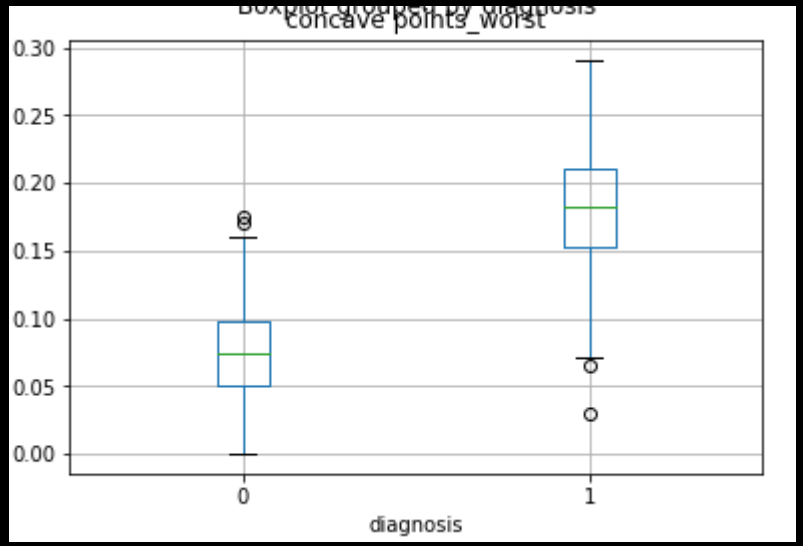


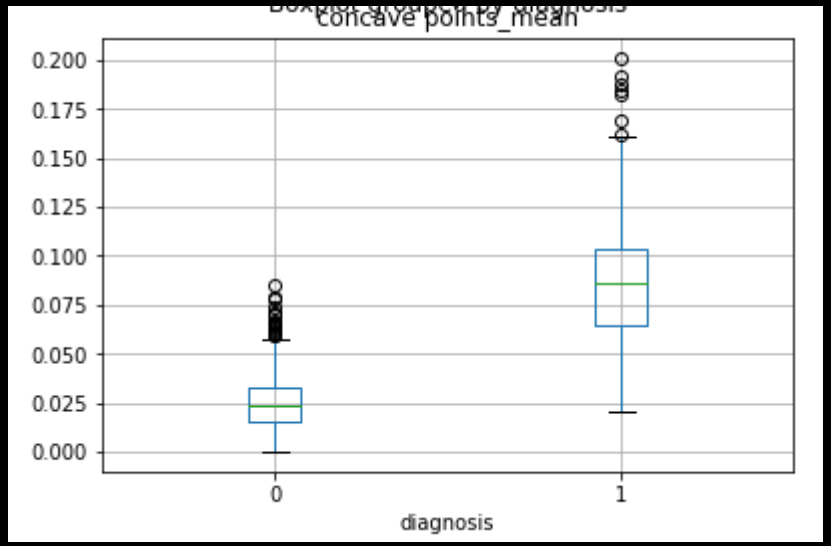

```
data.hist('radius_mean',bins=5)
```



BoxPlots:-

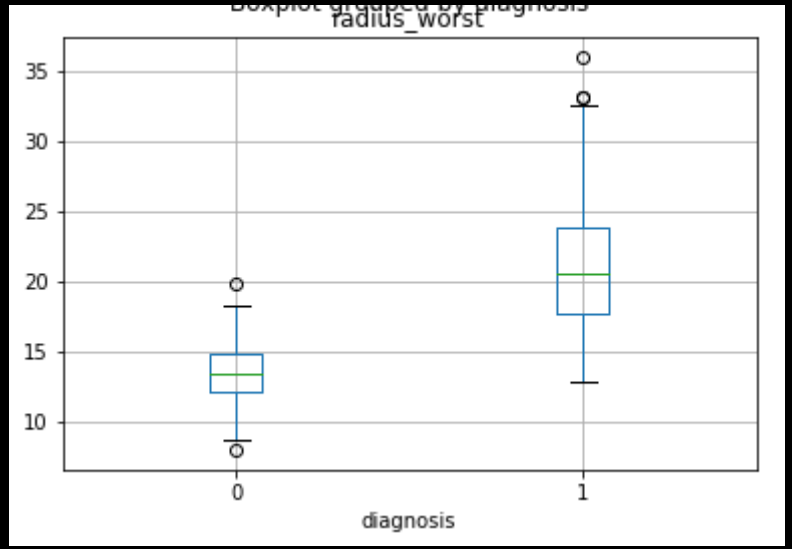
```
data.boxplot(column='concave points_worst',by='diagnosis')
```



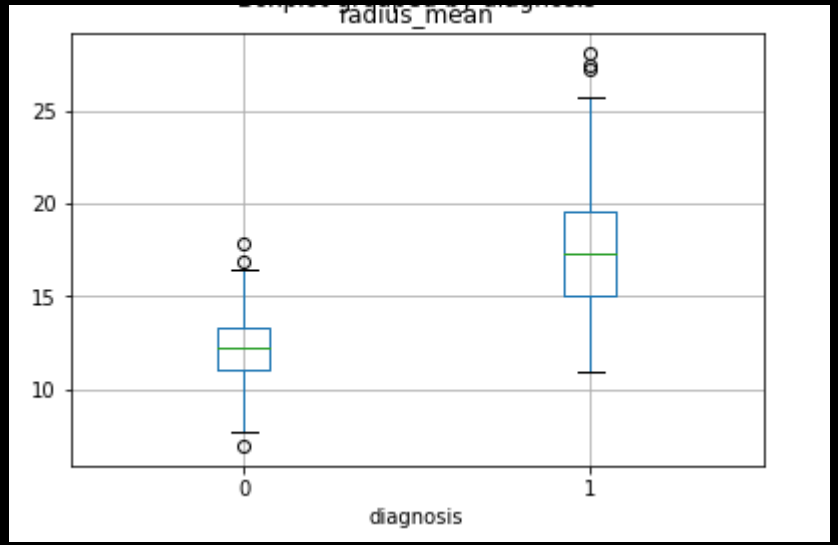



```
data.boxplot(column='concave points_mean',by='diagnosis')
```

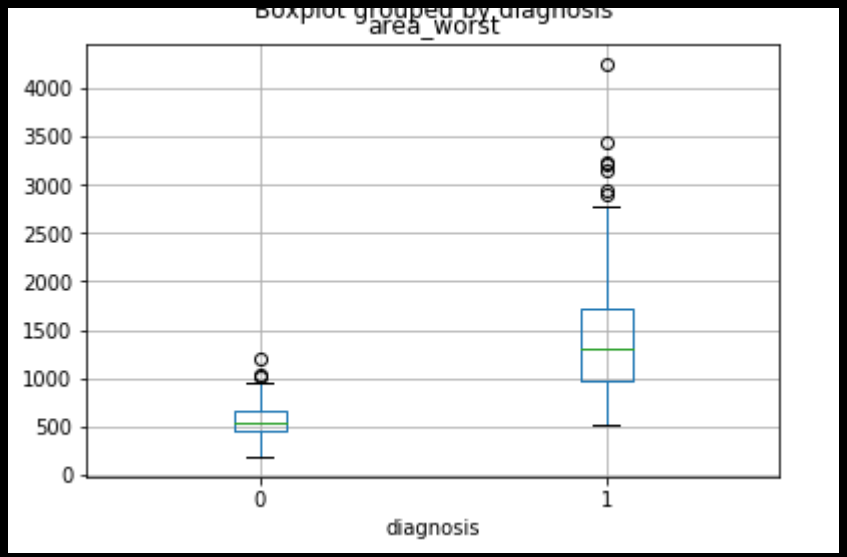
```
data.boxplot(column='radius_worst',by='diagnosis')
```



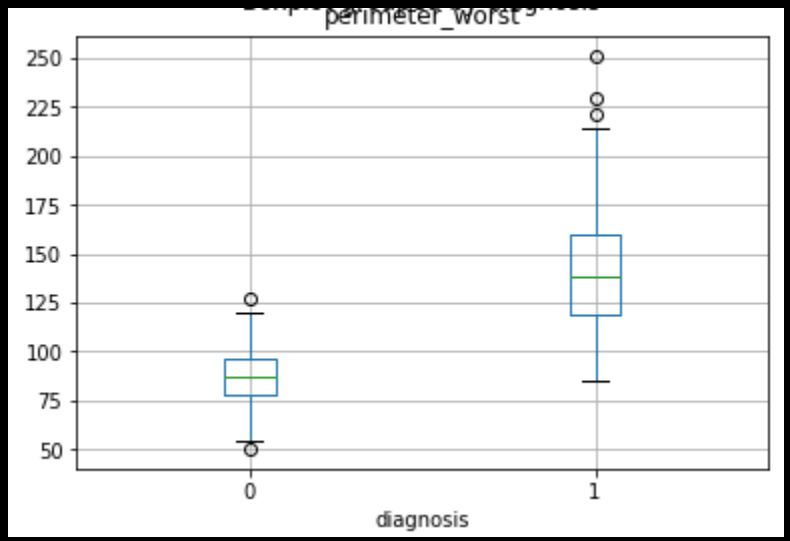

```
data.boxplot(column='radius_mean',by='diagnosis')
```



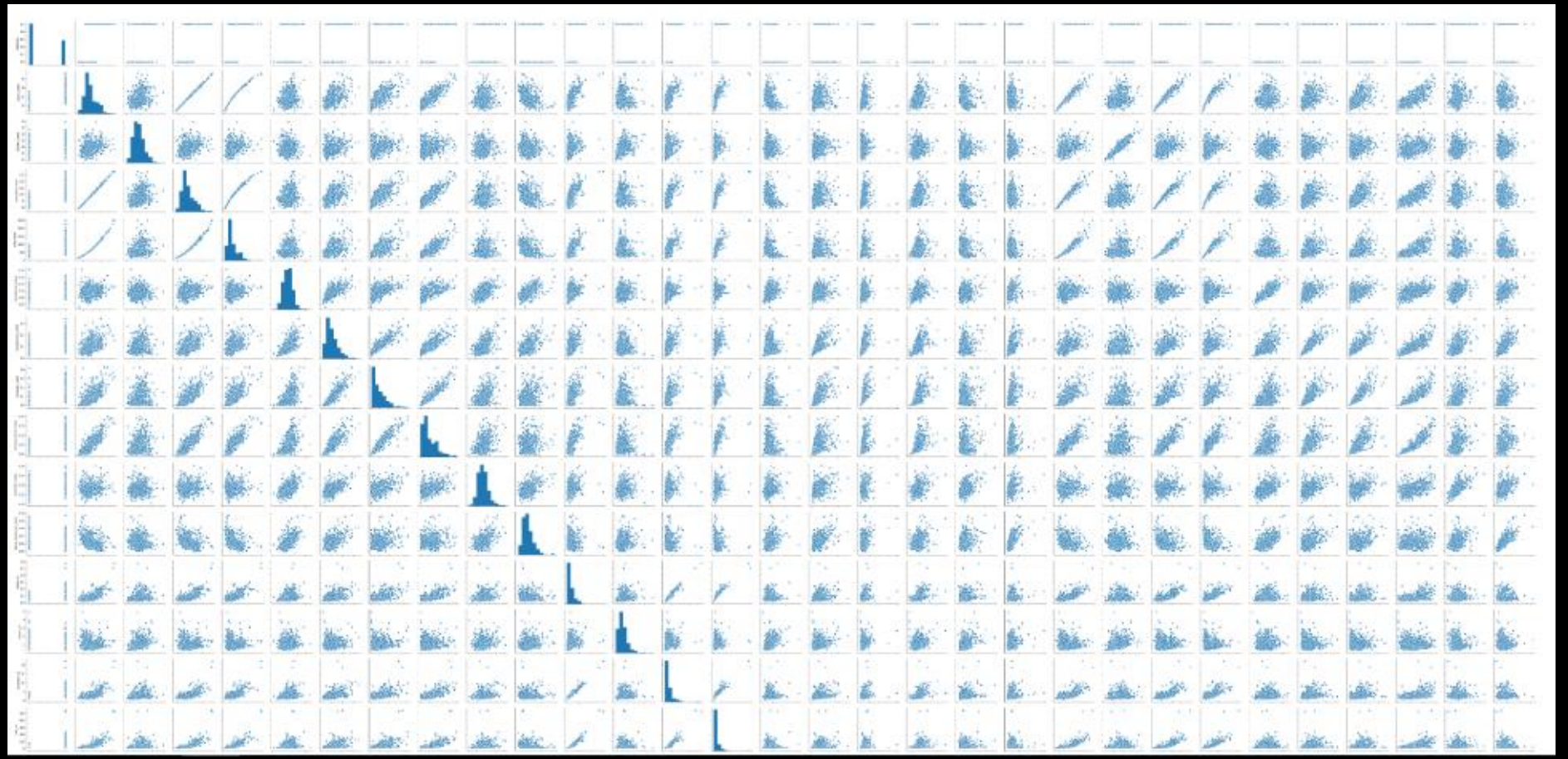
```
data.boxplot(column='area_worst',by='diagnosis')
```




```
data.boxplot(column='perimeter_worst',by='diagnosis')
```

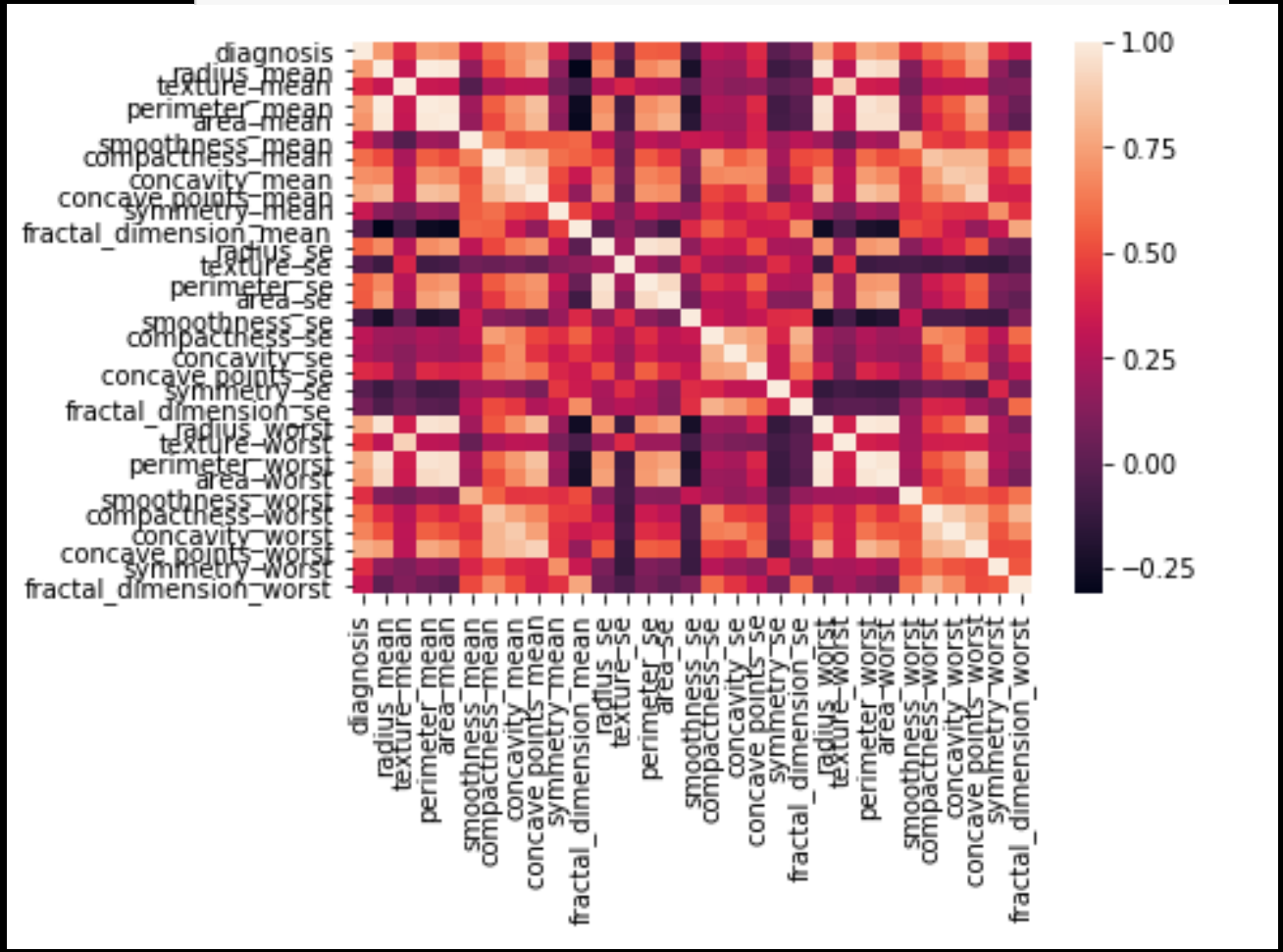


Pair plots



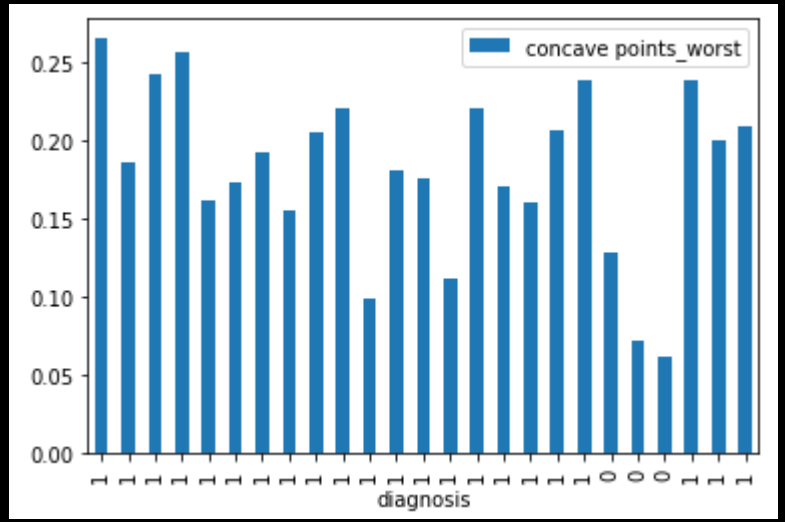
Heatmap

```
corr=data.corr()
sns.heatmap(corr,xticklabels=corr.columns,yticklabels=corr.columns)
```

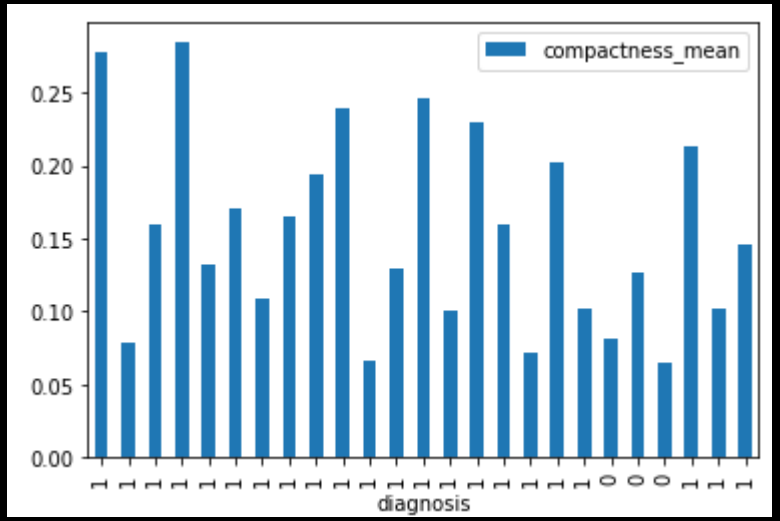


Bar graphs

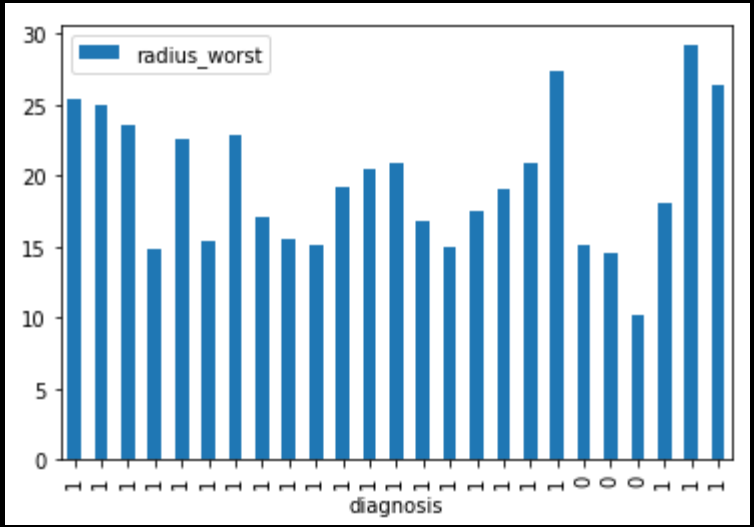
```
dt=data.head(25)
dt.plot.bar(y='concave points_worst',x='diagnosis')
```



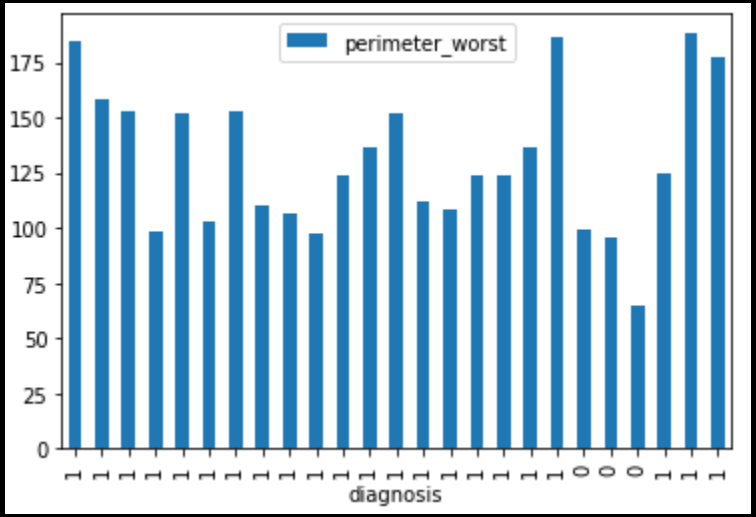

```
dt=data.head(25)
dt.plot.bar(y='compactness_mean',x='diagnosis')
```



```
dt=data.head(25)
dt.plot.bar(y='radius_worst',x='diagnosis')
```




```
dt=data.head(25)
dt.plot.bar(y='perimeter_worst',x='diagnosis')
```



Splitting Attributes

```
In [26]: X = data.iloc[:,1:]

In [27]: y = data.iloc[:,0]

In [28]: y.name

Out[28]: 'diagnosis'

In [29]: X.columns

Out[29]: Index(['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', 'smoothness_se',
                'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
                'fractal_dimension_se', 'radius_worst', 'texture_worst',
```

splitting Dataset into Training and

```
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)
```

```
X_train.shape[0],X_test.shape[0]
```

[31]:
```
(398, 171)
```

```
X_train
```

[32]:

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_m
478	11.490	14.59	73.99	404.9	0.10460	0.08228	0.053080	0.019690	0.1
303	10.490	18.61	66.86	334.3	0.10680	0.06678	0.022970	0.017800	0.1
155	12.250	17.94	78.27	460.3	0.08654	0.06679	0.038850	0.023310	0.1
186	18.310	18.58	118.60	1041.0	0.08588	0.08468	0.081690	0.058140	0.1
101	6.981	13.43	43.79	143.5	0.11700	0.07568	0.000000	0.000000	0.1

Why did we choose all the Attributes as Predictors?

We choose all the attributes as predictors because

- All the attributes given are at a good correlation with diagnosis of breast cancer
- These values are from cancer reports

\.

prediction

S. No	Algorithms	Accuracy1
1	Logistic Regression	Train - 94% Test – 92%
2	Decision Tree	Train – 100% Test – 88%
3	Random Forest	Train – 99% Test – 93%
4	Support Vector Classification	Train – 100% Test – 58%
5	Naive Bayes	Train – 94% Test – 92%
6	K- Nearest Neighbours	Train – 94% Test – 93%

Evaluating the Algorithm

For KNN

Data Splitting	Train(%)	Test(%)	Accuracy(%)
90-10	95	89	89
80-20	94	93	93
70-30	93	94	94
60-40	93	95	95
50-50	94	93	93

Precision and Recall

Precision	Recall
91.6%	93.6%

Results

SmartBridge

Regression

radius_mean **54**

Prediction **1**

texture_mean **54**

perimeter_mean **{"rm":54,"tm":54,"pm":54,"am":54,"sm":5,"cm":6,"ccm":56,"cpm":765,"smm":65,"fdm":654,"rs":56567,"ts":656,"ps":6**

area_mean **{"rm":54,"tm":54,"pm":54,"am":54,"sm":5,"cm":6,"ccm":56,"cpm":765,"smm":65,"fdm":654,"rs":56567,"ts":656,"ps":68,"as"**

smoothness_mean **{"rm":54,"tm":54,"pm":54,"am":54,"sm":5,"cm":6,"ccm":56,"cpm":765,"smm":65,"fdm":654,"rs":56567,"ts":656,"ps"**

compactness_mean **{"rm":54,"tm":54,"pm":54,"am":54,"sm":5,"cm":6,"ccm":56,"cpm":765,"smm":65,"fdm":654,"rs":56567,"ts":656,"ps**

concavity_mean **{"rm":54,"tm":54,"pm":54,"am":54,"sm":5,"cm":6,"ccm":56,"cpm":765,"smm":65,"fdm":654,"rs":56567,"ts":656,"ps":68**

concave points_mean **{"rm":54,"tm":54,"pm":54,"am":54,"sm":5,"cm":6,"ccm":56,"cpm":765,"smm":65,"fdm":654,"rs":56567,"ts":656,"ps":68,"a**

Thank You

Work by
Team Cosmos

- Madhuri
- Sahil Shaik
- Vamsi Krishna
- Sai Kiran
- Sarath Reddy