# import libraries   
import pandas as pd # Import Pandas for data manipulation using dataframes  
import numpy as np # Import Numpy for data statistical analysis   
import matplotlib.pyplot as plt # Import matplotlib for data visualisation  
import seaborn as sns # Statistical data visualization  
# %matplotlib inline

# Import Cancer data drom the Sklearn library  
from sklearn.datasets import load\_breast\_cancer  
cancer = load\_breast\_cancer()

sns.pairplot(df\_cancer, hue = 'target', vars = ['mean radius', 'mean texture', 'mean area', 'mean perimeter', 'mean smoothness'] )

# Let's check the correlation between the variables   
# Strong correlation between the mean radius and mean perimeter, mean area and mean primeter  
plt.figure(figsize=(20,10))   
sns.heatmap(df\_cancer.corr(), annot=True)

# Let's drop the target label coloumns  
X = df\_cancer.drop(['target'],axis=1)from sklearn.model\_selection import train\_test\_splitX\_train, X\_test, y\_train,

y\_test = train\_test\_split(X, y, test\_size = 0.20,

random\_state=5)

min\_train = X\_train.min()range\_train = (X\_train - min\_train).max()X\_train\_scaled = (X\_train - min\_train)/range\_trainsns.scatterplot(x = X\_train['mean area'], y = X\_train['mean smoothness'], hue = y\_train)from sklearn.svm import SVC   
from sklearn.metrics import classification\_report, confusion\_matrixsvc\_model = SVC()  
svc\_model.fit(X\_train\_scaled, y\_train)y\_predict = svc\_model.predict(X\_test\_scaled)  
cm = confusion\_matrix(y\_test, y\_predict)sns.heatmap(cm,annot=True,fmt="

#Using Logistic Regression Algorithm to the Training Setfrom sklearn.linear\_model import LogisticRegression  
classifier = LogisticRegression(random\_state = 0)  
classifier.fit(X\_train, Y\_train)#Using KNeighborsClassifier Method of neighbors class to use Nearest Neighbor algorithm*from sklearn.neighbors import KNeighborsClassifier  
classifier = KNeighborsClassifier(n\_neighbors = 5, metric = 'minkowski', p = 2)  
classifier.fit(X\_train, Y\_train)*  
#Using SVC method of svm class to use Support Vector Machine Algorithm  
*from sklearn.svm import SVC  
classifier = SVC(kernel = 'linear', random\_state = 0)  
classifier.fit(X\_train, Y\_train)*  
#Using SVC method of svm class to use Kernel SVM Algorithm  
*from sklearn.svm import SVC  
classifier = SVC(kernel = 'rbf', random\_state = 0)  
classifier.fit(X\_train, Y\_train)*  
#Using GaussianNB method of naïve\_bayes class to use Naïve Bayes Algorithm  
*from sklearn.naive\_bayes import GaussianNB  
classifier = GaussianNB()  
classifier.fit(X\_train, Y\_train)*  
#Using DecisionTreeClassifier of tree class to use Decision Tree Algorithm  
*from sklearn.tree import DecisionTreeClassifier  
classifier = DecisionTreeClassifier(criterion = 'entropy', random\_state = 0)  
classifier.fit(X\_train, Y\_train)*  
#Using RandomForestClassifier method of ensemble class to use Random Forest Classification algorithm  
*from sklearn.ensemble import RandomForestClassifier  
classifier = RandomForestClassifier(n\_estimators = 10, criterion = 'entropy', random\_state = 0)  
classifier.fit(X\_train, Y\_train)*

t