Support vector machine (SVM) classifier Implementation on Breast Cancer Dataset

Problem statement:

In the world Breast Cancer is the familiar cancer to women and more then 3 Million women affected from this cancer. Different Machine learning algorithms used for early diagnosing to detect disease at initial stage. In this project we are detecting tumors in to 2 categories malignant and benign mean cancerous or non-cancerous.

Dataset and Features:

Dataset is taken from official UCI website and the link of the data is given below.
http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29

These are the features of our dataset.

```
'feature_names': array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal dimension',
       'radius error', 'texture error', 'perimeter error', 'area error',
       'smoothness error', 'compactness error', 'concavity error',
       'concave points error', 'symmetry error',
       'fractal dimension error', 'worst radius', 'worst texture',
       'worst perimeter', 'worst area', 'worst smoothness',
       'worst compactness', 'worst concavity', 'worst concave points',
       'worst symmetry', 'worst fractal dimension'], dtype='<U23'),
```

Data is loaded directly using sklearn library:

```
#Pandas library for reading data and preprocessing data
import pandas as pd
import numpy as np
#Matplotlib library used for ploting graphs
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

#Import Cancer data from the Sklearn library

from sklearn.datasets import load_breast_cancer
Breastcancer = load_breast_cancer()
```

This is the data features view in Data frame format.

```
df_BreastCancer = pd.DataFrame(np.c_[Breastcancer['data'], Breastcancer['target']], columns = np.append(Breastcancer['feature_names'], ['target'])

df_BreastCancer.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	radius error	texture error	perimeter error	area error	smoothness error
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	1.0950	0.9053	8.589	153.40	0.006399
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.006150
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	0.4956	1.1560	3.445	27.23	0.009110
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.011490

Some explanation of features in our dataset

Mean radius feature:

Mean radius is the mean from center point to perimeter.

Mean texture feature:

Standard deviation

Mean perimeter feature:

Size of tumor (mean)

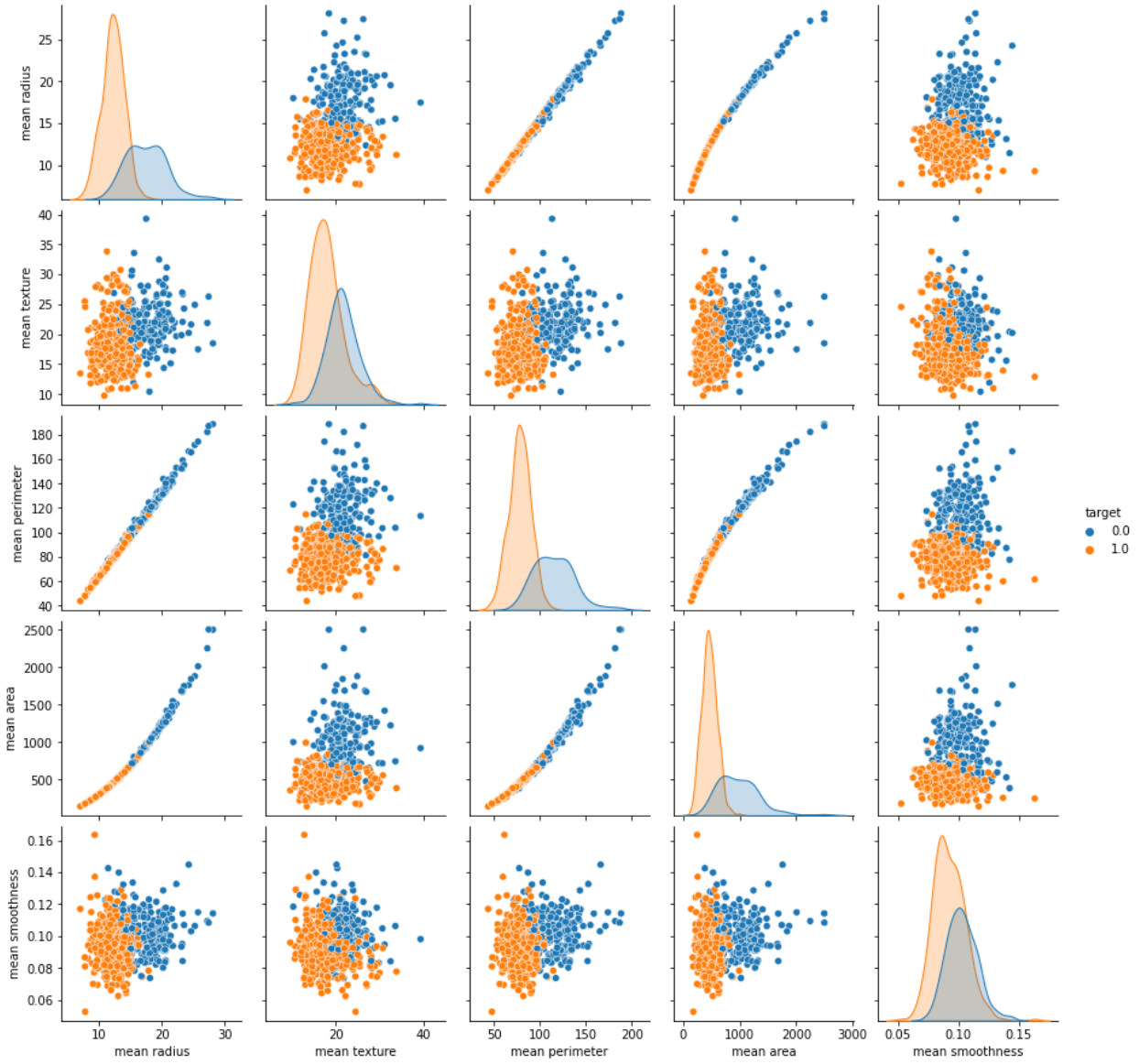
Mean smoothness feature:

Value of radius length(mean)

Mean compactness feature:

Perimeter means.

Features Visualization:



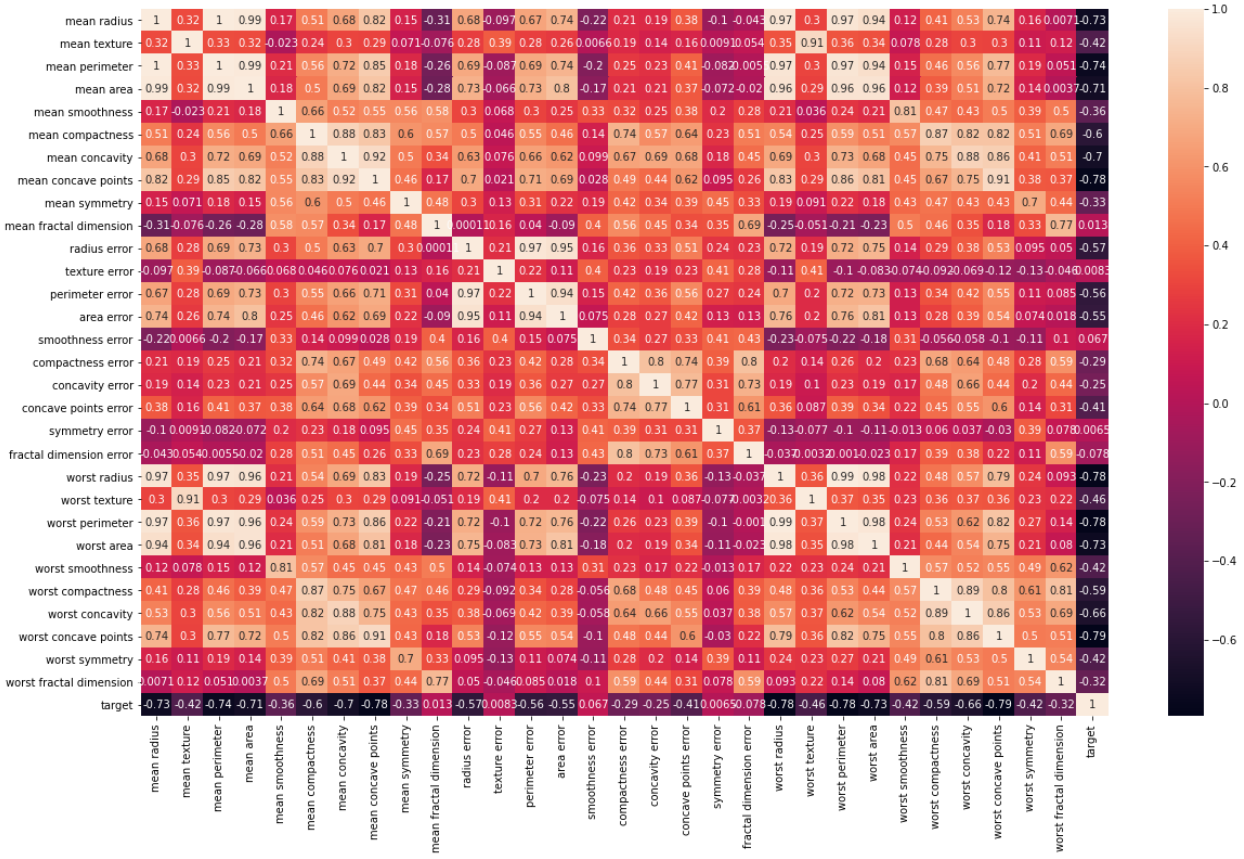
Orange data points represented as no cancer.

Blue data points represented as Cancer.

Now we will try to visualize the co relation between our data features.

Correlations:

```
1   plt.figure(figsize=(20,12))
2   sns.heatmap(df_BreastCancer.corr(), annot=True)
```



From this figure w can see that we have a strong relationship b/w mean perimeter and radius mean.

Now we have a bigger picture of the dataset and we can move into our main part which is classification using SVM.

SVM:

A SVM is famous classification model which deal with binary classification. I can classify both linear and nonlinear boundaries. SVM fits the linear boundary between datapoints and this way it can classify the data.

Data Training:

We will divide our data set into two groups. One is our training features and we will denote this to X and w have one target feature which is basically our output and we will assign into Y.

```
X = df_BreastCancer.drop(['target'], axis = 1)
X.head()
```

We dropped the target column from the data set and stored in a X variable.

```
y = df_BreastCancer['target']
y.head()
```

In the same way we have taken only Target feature from the whole data and saved into Y variable.

After assigning the data into X and Y now we are going to split the data into training and testing

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 20)
```

Data is divided 80% into training and 20% into testing.

Data is divided into testing and training now we will fit our model of SVM into training dataset.

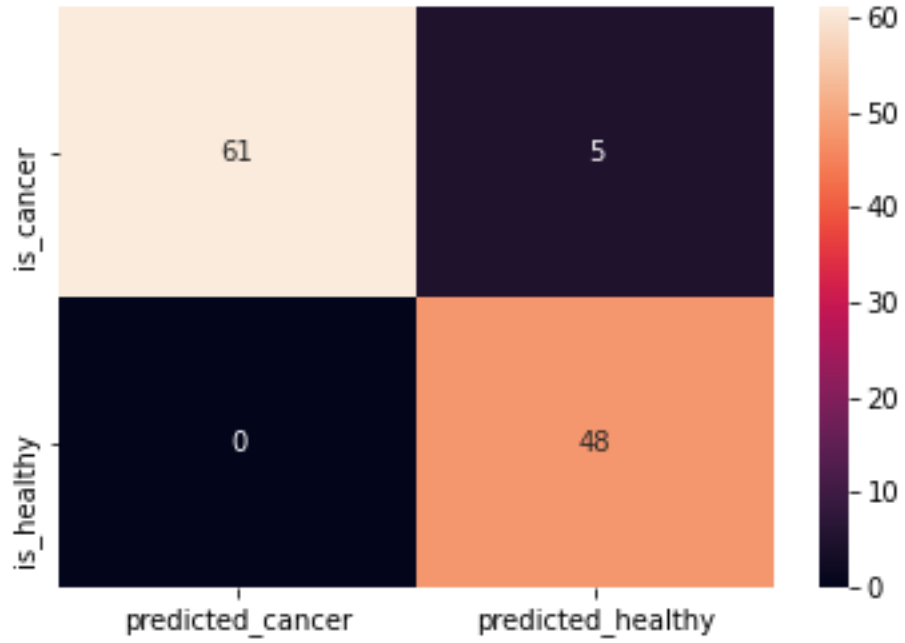
```
svc_model.fit(X_train, y_train)
```

Now its time to prediction using testing dataset.

```
y_predict = svc_model.predict(X_test)
```

Now we will compare the results using confusion matrix which is a very famous accuracy measure technique using test dataset.

	predicted_cancer	predicted_healthy
is_cancer	61	5
is_healthy	0	48



	precision	recall	f1-score	support
0.0	0.91	1.00	0.95	48
1.0	1.00	0.92	0.96	66
accuracy			0.96	114
macro avg	0.95	0.96	0.96	114
weighted avg	0.96	0.96	0.96	114

We can see we have achieved 96% accuracy and our model is working perfectly with high accuracy.