Breast Cancer Prediction

Abstract:

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics. Here we are using the Logistic Regression,K – Nearest Neighbor Classifier,Random Forest Classifier and XGBoost Classifier algorithm are used to compare the performance of the model.The model with the best results will be used and then classify the cancer as malignant or benign.

Introduction:

According to the world health organization (WHO) Breast cancer is the most frequent cancer among women, impacting 2.1 million women each year, and also causes the greatest number of cancer-related deaths among women. In 2018, it is estimated that 627,000 women died from breast cancer-that is approximately 15% of all cancer deaths among women and the number is still increasing.While breast cancer rates are higher among women in more developed regions, rates are increasing in nearly every region globally. Breast cancer is a dangerous disease for women. If it does not identify in the early-stage then the result will be the death of the patient. It is a common cancer in women worldwide.The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Thus, the correct diagnosis of BC and classification of patients into malignant or benign groups is the subject of much research.

Mammography:The most important screening test for breast cancer is the mammogram. A mammogram is an X-ray of the breast. It can detect breast cancer up to two years before the tumor can be felt by you or your doctor. Women age 40–45 or older who are at average risk of breast cancer should have a mammogram once a year.The chance of getting breast cancer increases as women age. Nearly 80 percent of breast cancers are found in women over the age of 50. A woman has a higher risk of breast cancer if her mother, sister or daughter had breast cancer, especially at a young age (before 40). Having other relatives with breast cancer may also raise the risk.Women with certain genetic mutations, including changes to the BRCA1 and BRCA2 genes, are at higher risk of developing breast cancer during their lifetime. Other genetic changes may raise breast cancer risk as well.

In the healthcare industry, there is a lot of evidence that machine learning algorithms can provide effective models to solve problems in order to identify patients.Many researchers and scientists related to machine learning are also involved in solving this situation.Classification and data mining methods are an effective way to classify data. Especially in medical field, where those methods are widely used in diagnosis and analysis to make decisions. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML) is widely recognized

as the methodology of choice in BC pattern classification and forecast modelling.This study will help to understand and predict breast cancer accordingly.

Methodology:

The dataset used in this story is publicly available and was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA. To create the dataset Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt, which is capable of perform the analysis of cytological features based on a digital scan. The goal is to classify whether the breast cancer is benign or malignant.To build the best model, we have to train and test the dataset with multiple Machine Learning algorithms then we can find the best ML model.The XGB classifier is used here and compared with other three classifiers.To check the accuracy we need to import confusion_matrix method of metrics class. The confusion matrix is a way of tabulating the number of mis-classifications, i.e., the number of predicted classes which ended up in a wrong classification bin based on the true classes.We will use Classification Accuracy method to find the accuracy of our models. Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

Logistic Regression

Logistic regression was introduced by statistician DR Cox in 1958 and so predates the field of machine learning. It is a supervised machine learning technique, employed in classification jobs (for predictions based on training data). Logistic Regression uses an equation like Linear Regression,but the outcome of logistic regression is a categorical variable whereas it is a value for other regression models. Binary outcomes can be predicted from the independent variables. The steps followed are 1)Get a dataset 2)Train a classifier3)Make a prediction using such classifier.

k-Nearest Neighbour (k-NN)

K-Nearest Neighbour is a supervised machine learning algorithm as the data given to it is labelled. It is a nonparametric method as the classification of test data point relies upon the nearest training data points rather than considering the dimensions (parameters) of the dataset. (1) Input the dataset and split it into a training and testing set. (2) Pick an instance from the testing sets and calculate its distance with the training set. (3) List distances in ascending order. (4) The class of the instance is the most common class of the 3 first trainings instances (k=3).

Random Forest

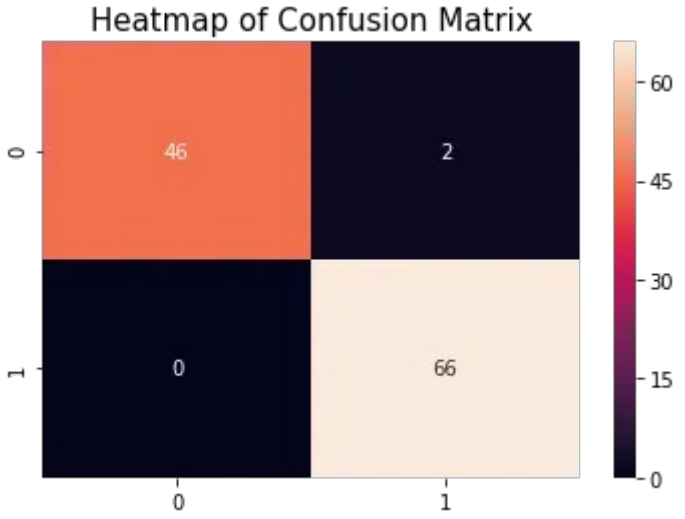
Random forests also known as random decision forestscreates a large number of trees that achieve their outputthrough ensemble learning methods for classification,regression. Bagging and feature randomness are thefeatures it uses to construct those trees. The random foresthas an advantage over the decision tree which, is that it does not overfit the data.

XGBoost Classifier

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance that is dominative competitive machine learning.XGBoost (Extreme Gradient

Boosting) belongs to a family of boosting algorithms and uses the gradient boosting (GBM) framework at its core. It is an optimized distributed gradient boosting library.This predictive models are designed to determine the class to which a given subject belongs. Since they use supervised learning, they require labeled training data that includes a column containing their class.

To check the correct prediction we have to check confusion matrix object and add the predicted results diagonally which will be number of correct prediction and then divide by total number of predictions.

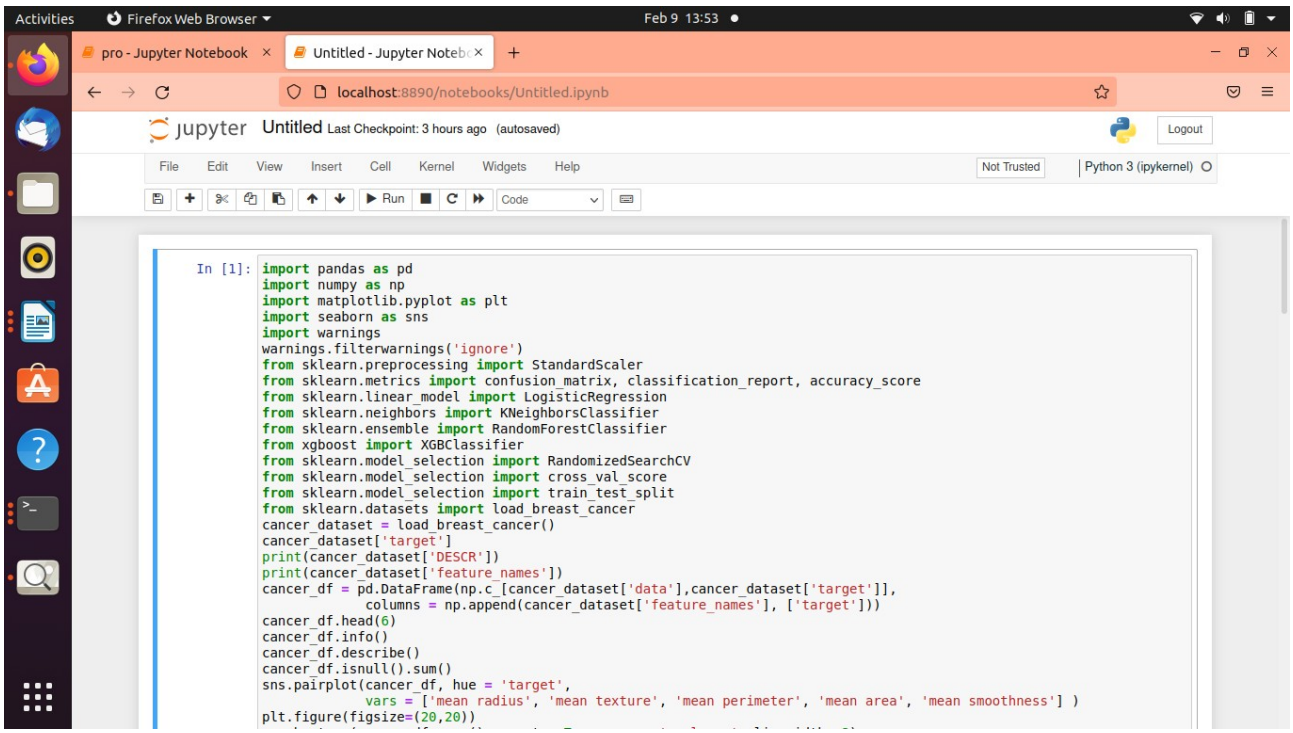


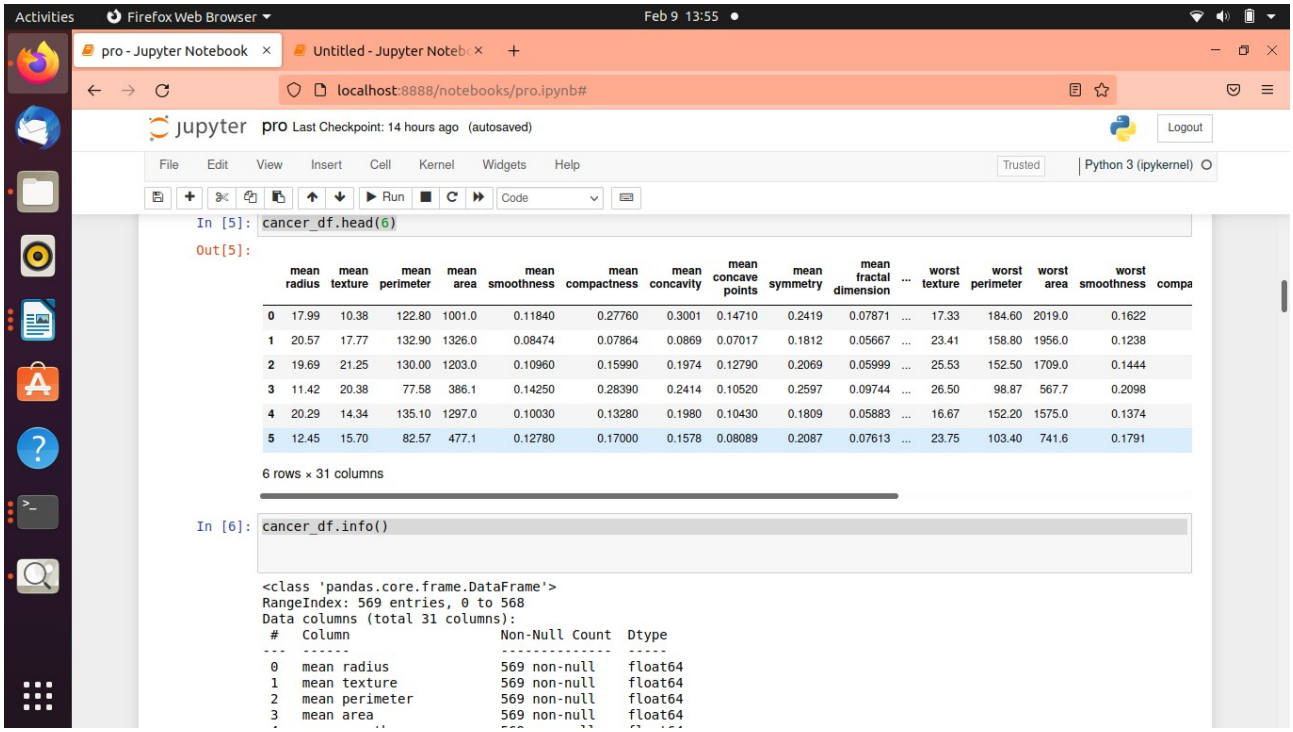
Here the matrices are of form

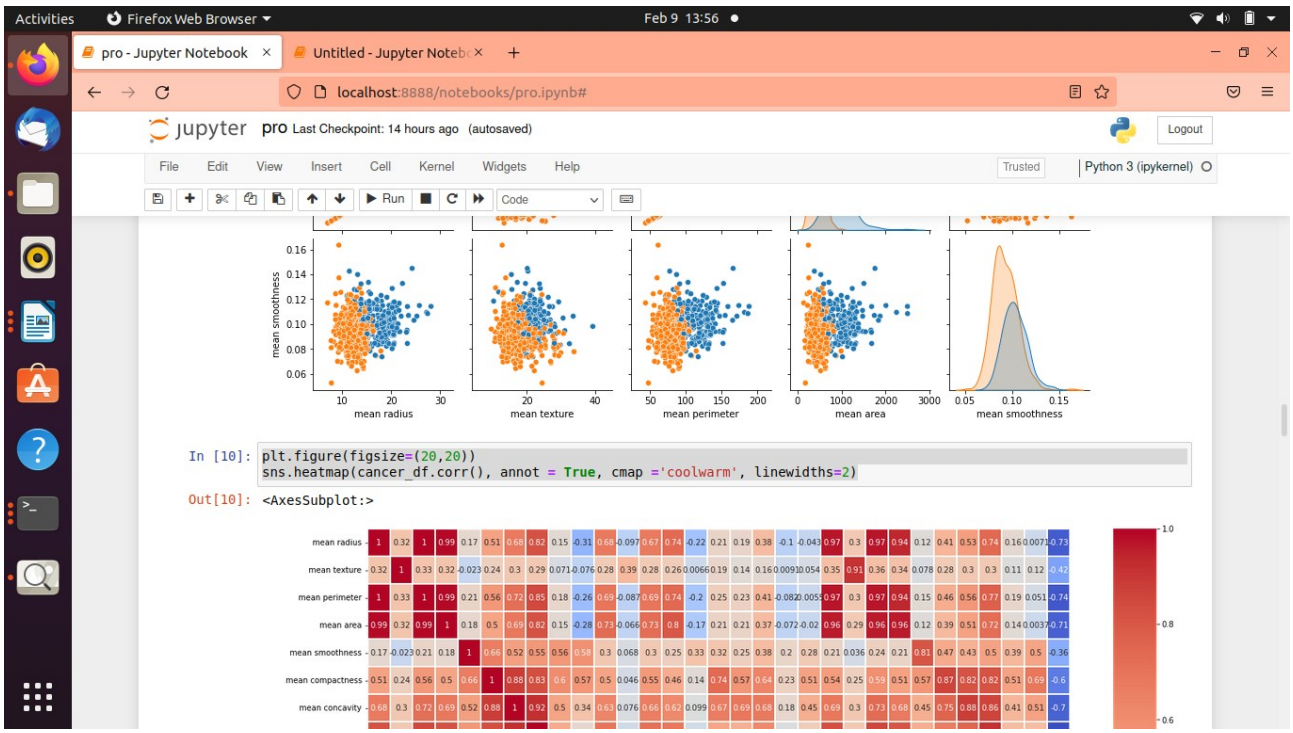
[TP FP]
[FN TP] where
TP is true positive: A true positive is an outcome where the model correctly predicts the positive class
TN is true negative: A true negative is an outcome where the model correctly predicts the negative class.
FN is false negative: A false negative is an outcome where the model incorrectly predicts the negative class.
FP is false positive: A false positive is an outcome where the model incorrectly predicts the positive class.
The model is giving 0% type II error and it is best.

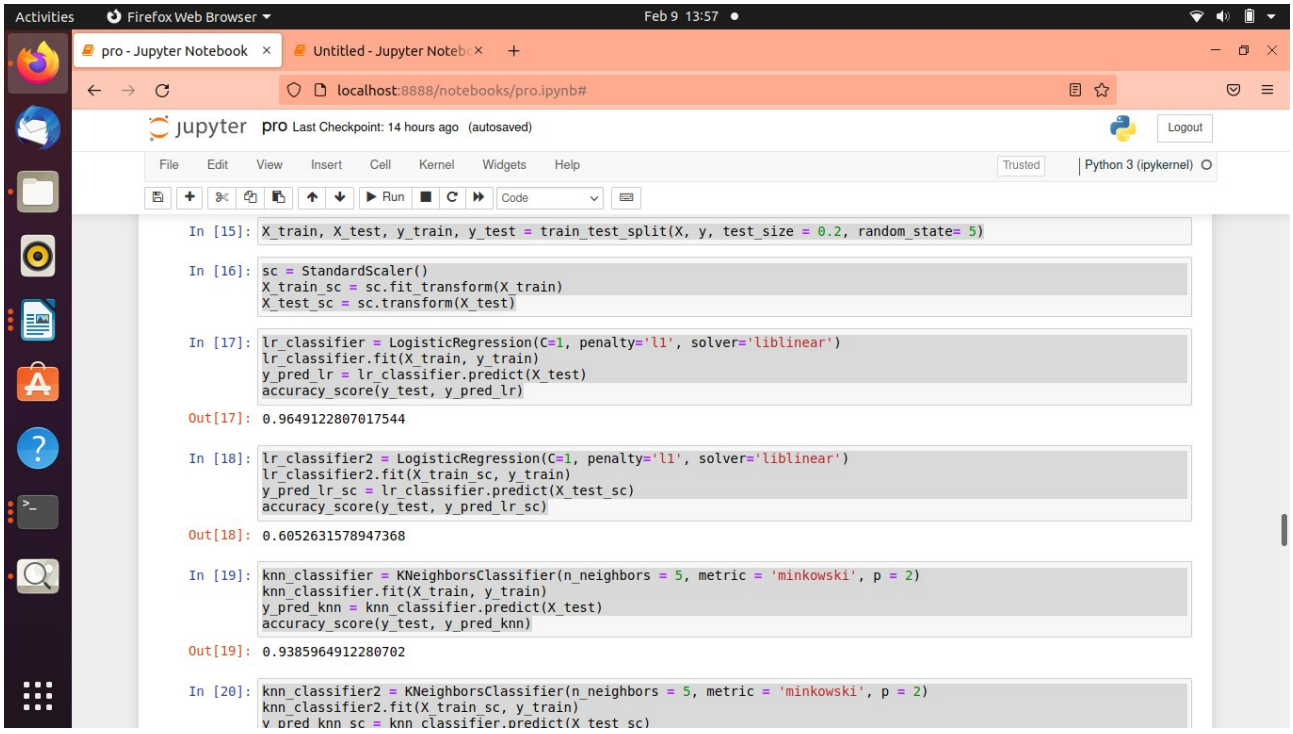
To find the ML model is overfitted, under fitted or generalize doing cross-validation.the mean accuracy value of **cross-validation is 96.24%** and **XGBoost model accuracy is 98.24%**. To get more accuracy, we trained all supervised classification algorithms but you can try out a few of them which are always popular. After training all algorithms, we found that Logistic Regression, Random Forest and XGBoost classifiers are given high accuracy than remaining but we have chosen XGBoost.

Code:


Activities
Firefox Web Browser
Feb 9 13:53
pro - Jupyter Notebook
Untitled - Jupyter Noteb
localhost:8890/notebooks/Untitled.ipynb
jupyter Untitled Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_breast_cancer
cancer_dataset = load_breast_cancer()
cancer_dataset['target']
print(cancer_dataset['DESCR'])
print(cancer_dataset['feature_names'])
cancer_df = pd.DataFrame(np.c_[cancer_dataset['data'],cancer_dataset['target']],
columns = np.append(cancer_dataset['feature_names'], ['target']))
cancer_df.head(6)
cancer_df.info()
cancer_df.describe()
cancer_df.isnull().sum()
sns.pairplot(cancer_df, hue = 'target',
vars = ['mean radius', 'mean texture', 'mean perimeter', 'mean area', 'mean smoothness'])
plt.figure(figsize=(20,20))


Activities
Firefox Web Browser
Feb 9 13:55
pro - Jupyter Notebook
Untitled - Jupyter Noteb
localhost:8888/notebooks/pro.ipynb#
jupyter pro Last Checkpoint: 14 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [5]:
cancer_df.head(6)
Out[5]:
mean radius mean texture mean perimeter mean area mean smoothness mean compactness mean concavity mean concave points mean symmetry mean fractal dimension ... worst texture worst perimeter worst area worst smoothness compa
0 17.99 10.38 122.80 1001.0 0.11840 0.27760 0.3001 0.14710 0.2419 0.07871 ... 17.33 184.60 2019.0 0.1622
1 20.57 17.77 132.90 1326.0 0.08474 0.07864 0.0869 0.07017 0.1812 0.05667 ... 23.41 158.80 1956.0 0.1238
2 19.69 21.25 130.00 1203.0 0.10960 0.15990 0.1974 0.12790 0.2069 0.05999 ... 25.53 152.50 1709.0 0.1444
3 11.42 20.38 77.58 386.1 0.14250 0.28390 0.2414 0.10520 0.2597 0.09744 ... 26.50 98.87 567.7 0.2098
4 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.1980 0.10430 0.1809 0.05883 ... 16.67 152.20 1575.0 0.1374
5 12.45 15.70 82.57 477.1 0.12780 0.17000 0.1578 0.08089 0.2087 0.07613 ... 23.75 103.40 741.6 0.1791
6 rows × 31 columns
In [6]:
cancer_df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
Column Non-Null Count Dtype
0 mean radius 569 non-null float64
1 mean texture 569 non-null float64
2 mean perimeter 569 non-null float64
3 mean area 569 non-null float64


Activities
Firefox Web Browser
Feb 9 13:56
pro - Jupyter Notebook
Untitled - Jupyter Noteb
localhost:8888/notebooks/pro.ipynb#
jupyter pro Last Checkpoint: 14 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
mean smoothness
mean radius
mean texture
mean perimeter
mean area
mean smoothness
In [10]:
plt.figure(figsize=(20,20))
sns.heatmap(cancer_df.corr(), annot = True, cmap ='coolwarm', linewidths=2)
Out[10]:
<AxesSubplot:>


Activities
Firefox Web Browser
Feb 9 13:57
pro - Jupyter Notebook
Untitled - Jupyter Noteb
localhost:8888/notebooks/pro.ipynb#
jupyter pro Last Checkpoint: 14 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [15]:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state= 5)
In [16]:
sc = StandardScaler()
X_train_sc = sc.fit_transform(X_train)
X_test_sc = sc.transform(X_test)
In [17]:
lr_classifier = LogisticRegression(C=1, penalty='l1', solver='liblinear')
lr_classifier.fit(X_train, y_train)
y_pred_lr = lr_classifier.predict(X_test)
accuracy_score(y_test, y_pred_lr)
Out[17]:
0.9649122807017544
In [18]:
lr_classifier2 = LogisticRegression(C=1, penalty='l1', solver='liblinear')
lr_classifier2.fit(X_train_sc, y_train)
y_pred_lr_sc = lr_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_lr_sc)
Out[18]:
0.6052631578947368
In [19]:
knn_classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
knn_classifier.fit(X_train, y_train)
y_pred_knn = knn_classifier.predict(X_test)
accuracy_score(y_test, y_pred_knn)
Out[19]:
0.9385964912280702
In [20]:
knn_classifier2 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
knn_classifier2.fit(X_train_sc, y_train)

Activities Firefox Web Browser Feb 9 13:57

pro - Jupyter Notebook | Untitled - Jupyter Noteb | localhost:8888/notebooks/pro.ipynb#

jupyter pro Last Checkpoint: 14 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [20]:
```
knn_classifier2 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
knn_classifier2.fit(X_train_sc, y_train)
y_pred_knn_sc = knn_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_knn_sc)
```

Out[20]: 0.5789473684210527

In [21]:
```
rf_classifier = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', random_state = 51)
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)
accuracy_score(y_test, y_pred_rf)
```

Out[21]: 0.9736842105263158

In [22]:
```
rf_classifier2 = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', random_state = 51)
rf_classifier2.fit(X_train_sc, y_train)
y_pred_rf_sc = rf_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_rf_sc)
```

Out[22]: 0.7543859649122807

In [23]:
```
xgb_classifier = XGBClassifier()
xgb_classifier.fit(X_train, y_train)
y_pred_xgb = xgb_classifier.predict(X_test)
accuracy_score(y_test, y_pred_xgb)
```

```
[23:00:29] WARNING: ../src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the o
bjective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restor
e the old behavior.
```

Out[23]: 0.9824561403508771

Activities Firefox Web Browser Feb 9 14:00

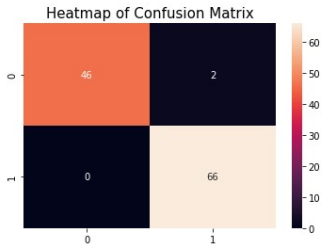
pro - Jupyter Notebook | Untitled - Jupyter Noteb | localhost:8888/notebooks/pro.ipynb#

jupyter pro Last Checkpoint: 14 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [30]:
```
cm = confusion_matrix(y_test, y_pred_xgb_pt)
plt.title('Heatmap of Confusion Matrix', fontsize = 15)
sns.heatmap(cm, annot = True)
plt.show()
```

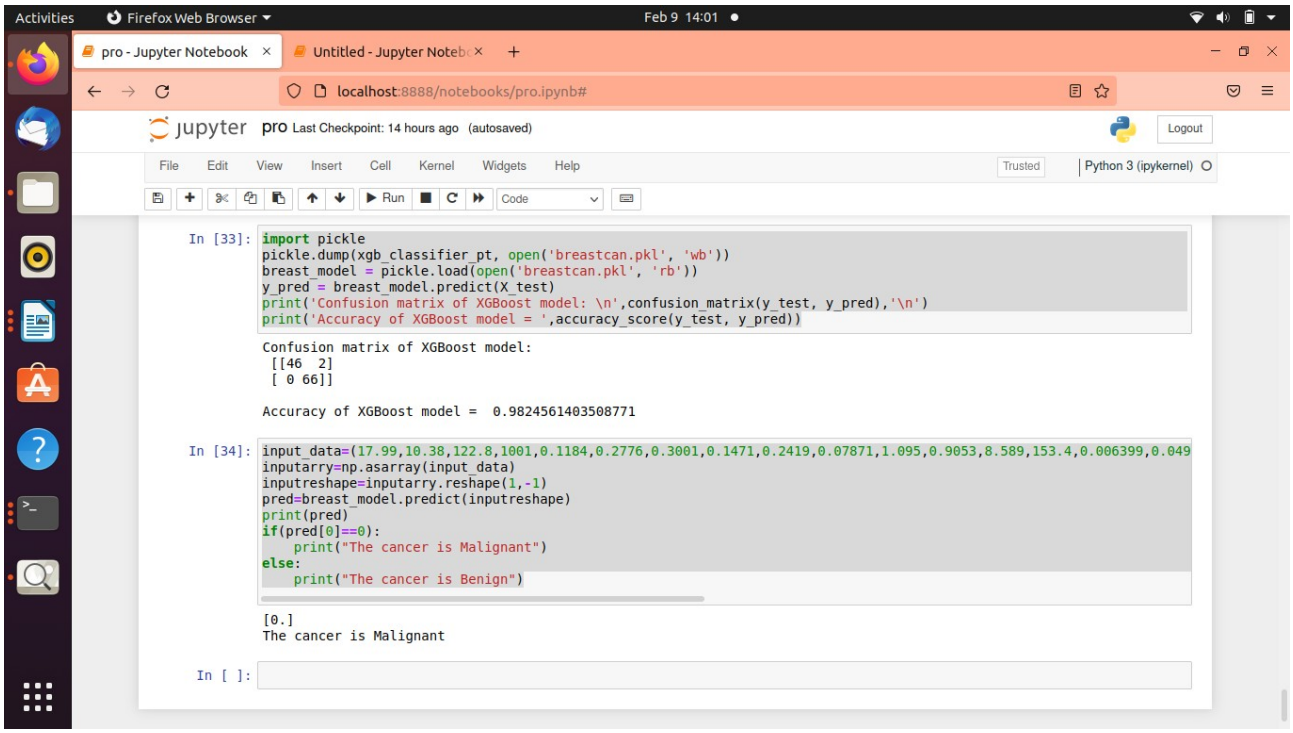


In [31]:
```
print(classification_report(y_test, y_pred_xgb_pt))
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.96      0.98        48
         1.0       0.97      1.00      0.99        66

    accuracy                           0.98       114
   macro avg       0.99      0.98      0.98       114
```



Refernces:

1) https://indianaiproduction.com/breast-cancer-detection-using-machine-learning-classifier

2) http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29

3) Wang, D. Zhang and Y. H. Huang “Breast Cancer Prediction Using Machine Learning” (2018), Vol. 66, NO. 7.

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

Presented By

Anjali Maria V