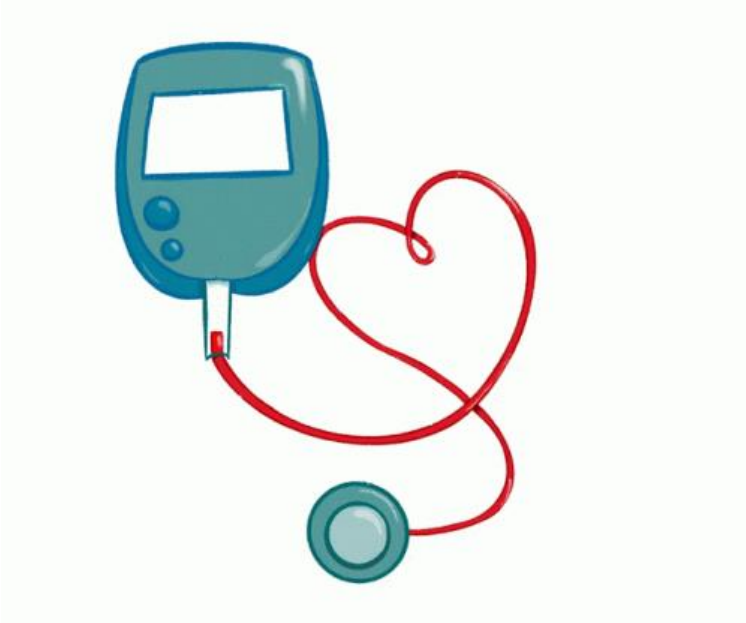
A
Project Report
On

" DIAPRO – Diabetes Prediction Application"

(Exposys Data Labs Internship Project)



Prepared by
Dhyey Joshi

Under the Supervision of
Vishnu Vardhan Sir

Submitted to

Exposys Data Labs

ABSTRACT

Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. According to the International Diabetes Federation 382 million people are living with diabetes across the whole world. By 2035, this will be doubled as 592 million. Diabetes mellitus or simply diabetes is a disease caused due to the increased level of blood glucose. Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. However, early prediction of diabetes is quite a challenging task for medical practitioners due to complex interdependence on various factors as diabetes affects human organs such as kidney, eye, heart, nerves, foot etc. Data science methods have the potential to benefit other scientific fields by shedding new light on common questions. One such task is to help make predictions on medical data. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. The aim of this project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by combining the results of different machine learning techniques. This project aims to predict diabetes via 10 different supervised & Ensemble Machine Learning methods including: SVM, K Nearest Neighbor, Naive Bayes, Logistic Regression, Random Forest Classifier, AdaBoost, XgBoost, Gradient Boost, LightGBM, Extra Tree Classifier. This project also aims to propose an effective technique for earlier detection of the diabetes disease.

Contents

List of Figures

Chapter 1

Project Definition

Diabetes Mellitus Diabetes is one of deadliest diseases in the world. It is not only a disease but also a creator of different kinds of diseases like heart attack, blindness, kidney diseases, etc. The normal identifying process is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day or more to get their reports. Moreover, every time they want to get their diagnosis report, they have to waste their money in vain.Diabetes Mellitus (DM) is defined as a group of metabolic disorders mainly caused by abnormal insulin secretion and/or action. Insulin deficiency results in elevated blood glucose levels (hyperglycemia) and impaired metabolism of carbohydrates, fat and proteins. DM is one of the most common endocrine disorders, affecting more than 200 million people worldwide. The onset of diabetes is estimated to rise dramatically in the upcoming years. DM can be divided into several distinct types. However, there are two major clinical types, type 1 diabetes (T1D) and type 2 diabetes (T2D), according to the etiopathology of the disorder. T2D appears to be the most common form of diabetes (90% of all diabetic patients), mainly characterized by insulin resistance. The main causes of T2D include lifestyle, physical activity, dietary habits and heredity, whereas T1D is thought to be due to autoimmune logical destruction of the Langerhans islets hosting pancreatic-β cells. T1D affects almost 10% of all diabetic patients worldwide, with 10% of them ultimately developing idiopathic diabetes. Other forms of DM, classified on the basis of insulin secretion profile and/or onset, include Gestational Diabetes, endocrinopathies, MODY (Maturity Onset Diabetes of the Young), neonatal, mitochondrial, and pregnancy

diabetes. The symptoms of DM include polyuria, polydipsia, and significant weight loss among others. Diagnosis depends on blood glucose levels (fasting plasma glucose = 7.0 mmol/L.

Supervised Learning: In supervised learning, the system must "learn" inductively a function called target function, which is an expression of a model describing the data. The objective function is used to predict the value of a variable, called dependent variable or output variable, from a set of variables, called independent variables or input variables or characteristics or features. The set of possible input values of the function, i.e. its domain, are called instances. Each case is described by a set of characteristics (attributes or features). A subset of all cases, for which the output variable value is known, is called training data or examples. In order to infer the best target function, the learning system, given a training set, takes into consideration alternative functions, called hypothesis and denoted by h. In supervised learning, there are two kinds of learning tasks: classification and regression. Classification models try to predict distinct classes, such as e.g. blood groups, while regression models predict numerical values. Some of the most common techniques are Decision Trees (DT), Rule Learning, and Instance Based Learning (IBL), such as k-Nearest Neighbours (k-NN), Genetic Algorithms (GA), Artificial Neural Networks (ANN), and Support Vector Machines (SVM).

Diabetes Prediction using Machine Learning & Ensemble Learning Algorithm's:

1.SVM
2.K Nearest Neighbor
3.Naive Bayes
4.Logistic Regression
5.RandomForest Classifier

6.AdaBoost
7.XGboost
8.Gradient Boost
9.LightGBM
10.Extra Tree Classifier

Model Deployment using Flask

Often we focus a lot in the EDA, Model Development part, however there is one more major aspect that we tend to miss out i.e creating an end to end application or deploying the model, after all that hard work and efforts that you had put in developing your model it is equally important that you give it in a usable form that can be consumed by the end user directly. Moreover building an application around the code you have developed helps you make you make your work more presentable and helps you showcase your work better.

Flask is a a web application framework written in python, in simple terms it helps end users interact with your python code (in this case our ML models) directly from their web browser without needing any libraries, code files, etc.
Flask enables you to create web applications very easily, hence enabling you to focus your energy more on other important parts of a ML lifecycle like EDA, feature engineering, etc. I will give you a walkthrough on how to build a simple web application out of your ML Model and deploying it eventually. It's a Python module that lets you develop web applications easily. It has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine. It is a WSGI web app framework.

Deployment on Heroku Platform

To deploy your app to Heroku, you typically use the git push command to push the code from your local repository's master or main branch to your Heroku remote, like so: $ git push Heroku main Initializing repository, done.

Proposed Method

The whole project will be completed in 3 complex steps
a. Creating a model using machine learning
b. Creating a web app using flask and connecting it with model
c. Now, uploading project to GitHub, then connect Heroku with your GitHub account. Name your application – Click on Deploy Branch. Wahoo!! our application on fly now.

Classification is one of the most important decision making techniques in many real world problems. In this work, the main objective is to classify the data as diabetic or non-diabetic and improve the classification accuracy. For many classification problems, the higher number of samples chosen doesn't leads to higher classification accuracy. In many cases, the performance of algorithms is high in the context of speed but the accuracy of data classification is low. The main objective of our model is to achieve high accuracy. Classification accuracy can be increased if we use much of the data set for training and few data sets for testing. This survey has analyzed various classification techniques for classification of diabetic and non-diabetic data. Thus, it is observed that techniques like Gradient Boosting & K nearest Neighbor are most suitable for implementing the Diabetes prediction system.

K-NEAREST NEIGHBOURS CLASSIFIER (K-NN)

Nearest neighbor classification are based on learning analogy i.e., by comparing given test tuple with training tuples that are similar. Each tuple represent a point in an n- dimensional space. Any training tuples are stored in an n- dimensional pattern space. It is a tuple-based classifier that can simply locate the nearest neighbor in tuple space and labelling the unknown tuple with the same class label as that 2 i of the known neighbor. The k-nearest neighbor classifier searches the pattern space for the k-training tuple that are closest to the unknown tuple. These training tuples are k- nearest neighbor classifier of the unknown tuple. Closeness can be defined as any distance metric such as Euclidean distance. Nearest neighbor classifiers are distance based comparisons intrinsically assign equal weight to each attribute. Therefore, they can suffer from poor accuracy if there is noisy or irrelevant attribute.

SUPPORT VECTOR MACHINES (SVM)

Support Vector Machine is a new generation learning system based on recent advances in statistical learning theory. It is an algorithm for both linear and non-linear data. It transforms the original data in a higher dimension, from where it can find a hyper plane for separation of the data using essential training tuples called support vectors. A Support Vector Machine is a discriminative classifier formally defined by a separating hyper plane. In other words, given labelled training Support vector machine constructs a hyper plane or set of hyper planes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyper plane that has largest distance to the nearest training data point of any class so called functional margin, since in general the larger the margin the lower the generalization error of the classifier.

NAÏVE BAYESIAN CLASSIFIER

Naïve Bayesian classification is called naïve because it assumes class condition independence. That is, the effect of an attribute value of given class is independence of the values of the other attributes. This assumption is made to reduce computational costs, and hence is considered naïve. A Naïve Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. The major idea behind naïve Bayesian classification is to try and classify data by maximizing P (Ci/X) = P (X/C) P (Ci)/P(X) (Where ian index of class, each tuple is represented n- dimensional vector, X = (x1, x2, x3 …… xn) depending n measurements made on the tuple from n attributes, respectivelyA1, A2, A3 …… An. We are also given a set of N classesC1, C2, C3 …… CN).

RANDOM FOREST

Random forest Leo Breiman (2001) is an ensemble of decision trees based classifiers. Each tree is constructed by bootstrap sample from the data, and it uses a candidate set of features selected from a random set. It uses both bagging and random variable selection for tree building. Once the forest is formed, test instances are percolated down each tree and trees make their respective class prediction. The error rate of a random forest depends on the strength of each tree and correlation between any two trees. It can be used to rank the importance of variables in a regression or classification problem in a natural way.

BAGGING

Bootstrap aggregating, also called bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid over fitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach. Main reason for error in learning is due to noise, bias and variance. Noise is error by the target function, Bias is where the algorithm cannot learn the target and Variance comes from the sampling, and how it affects

the learning algorithm. Bagging minimizes these errors. Averaging over bootstrap samples can reduce error from variance especially in case of unstable classifiers.

BOOSTING

Boosting is a machine learning ensemble meta- algorithm for reducing bias primarily and also variance in supervised learning, and a family of machine learning algorithms which convert weak learners to strong ones. Boosting is based on the question posed by Kearns and Valiant (1988, 1989): Can a set of weak learners create a single strong learner? A weak learner is defined to be a classifier which is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

Implementation

Creating a model using machine learning approaches:

a. Install all required packages all the packages can be installed using pip from Command (terminal)
   pip install pandas ,NumPy ,matplotlib ,scikit-learn ,seaborn , XgBoost , Lightgbm

   Also you can install all the required packages by using requirements.txt provided in the code folder.

   pip install -r requirements.txt

b. Install jupyter notebook
   pip install jupyter

c. Open jupyter notebook
   python -m notebook

   Open a new notebook in jupyter, follow the below steps along:

   a. Import the necessary libraries
   b. Load the dataset
   c. EDA on dataset
   d. Modeling and training
   e. Improve accuracy using Hyperparameter tuning
   f. Evaluate the model
   g. Save and load the model

   Creating a web app using flask and connecting it with model

   Create folder structure as follow:

```
flask(root)
|____templates
        |___main.html
        |___result.html
|____static
        |____Images
```

|____app.py
|_____GB_Classifier.pkl
|_____knn_Classifier.pkl

Write code in app.py

EXECUTION

a. Open command Prompt navigate to project > code > flask
b. Run the command

```
python app.py
```

Deploying webapp/model on Heroku Platform

Open Heroku Platform – Create new app – Name the application – Connect your GitHub account where all your project is uploaded - Click on connect – Click on Deploy Branch – Result: Give you domain name to visit your first DiaPro App online.

Chapter 2

Software and Hardware Requirement

2.1 Software Requirement

Following are the software requirement necessary of the project:

a) Python programming language.
b) Jupyter Notebook.
c) Google Colab.
d) Windows 7 / 10 Operating System.

2.2 Hardware Requirement

Following are the hardware requirement that is most important for the project:

2.2.1 Fluently working Laptops.
2.2.2 RAM minimum 4Gb.
2.2.3 Web Camera

Chapter 3

System Flow Chart

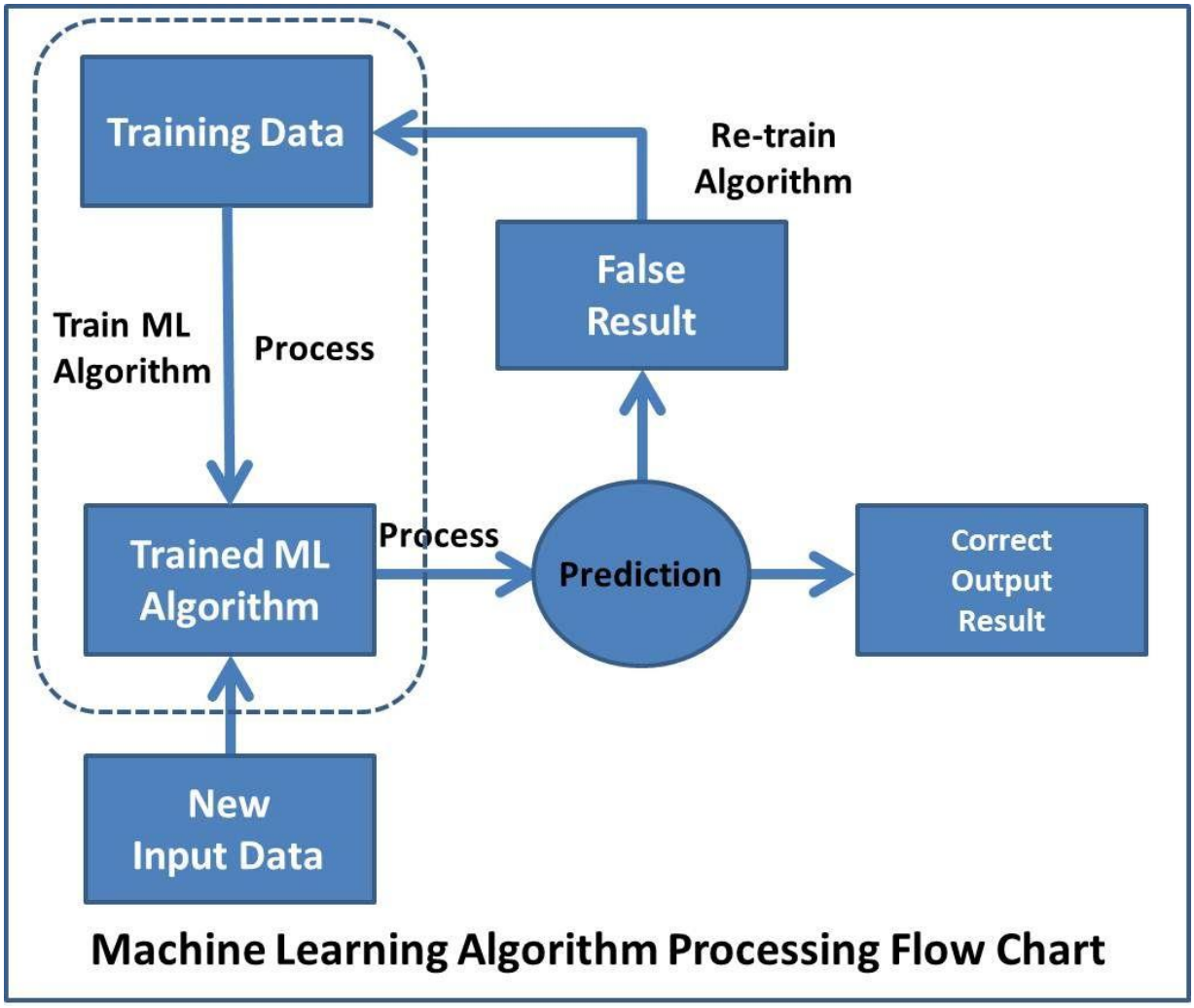
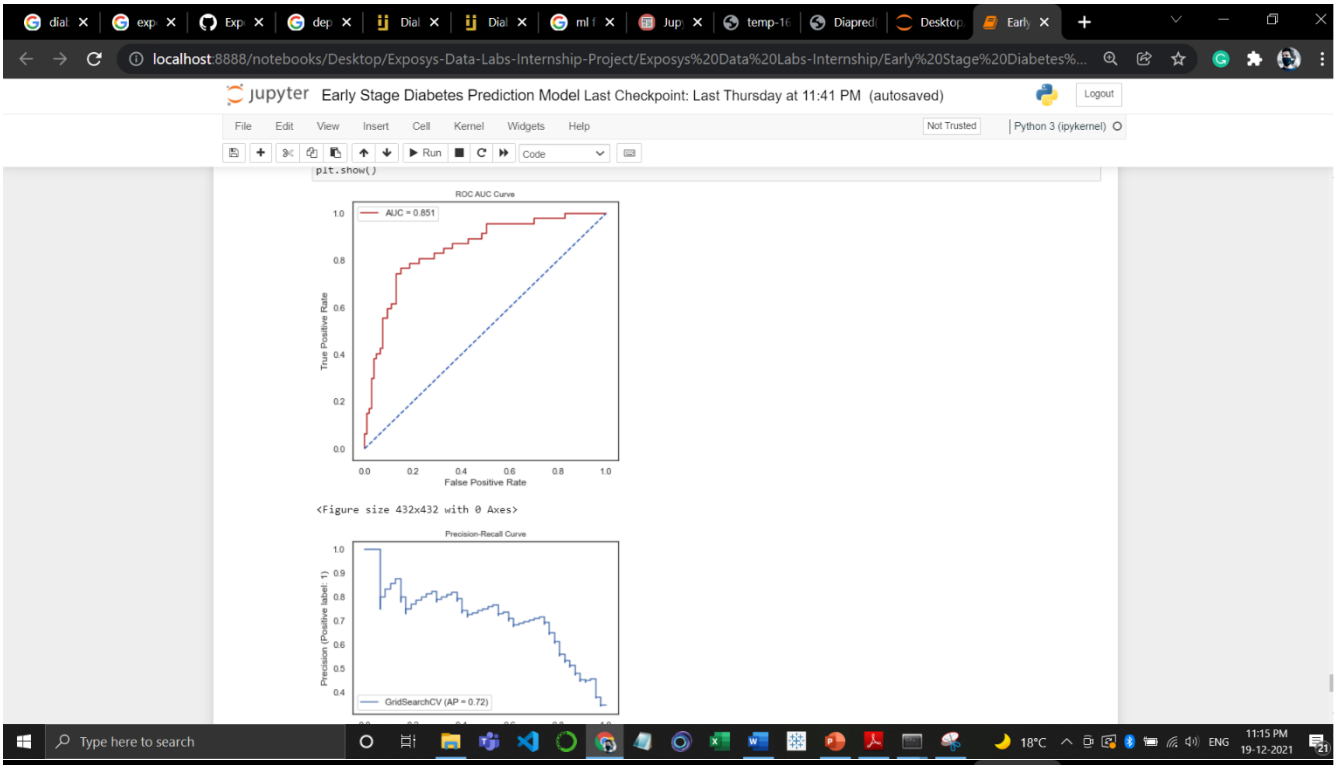
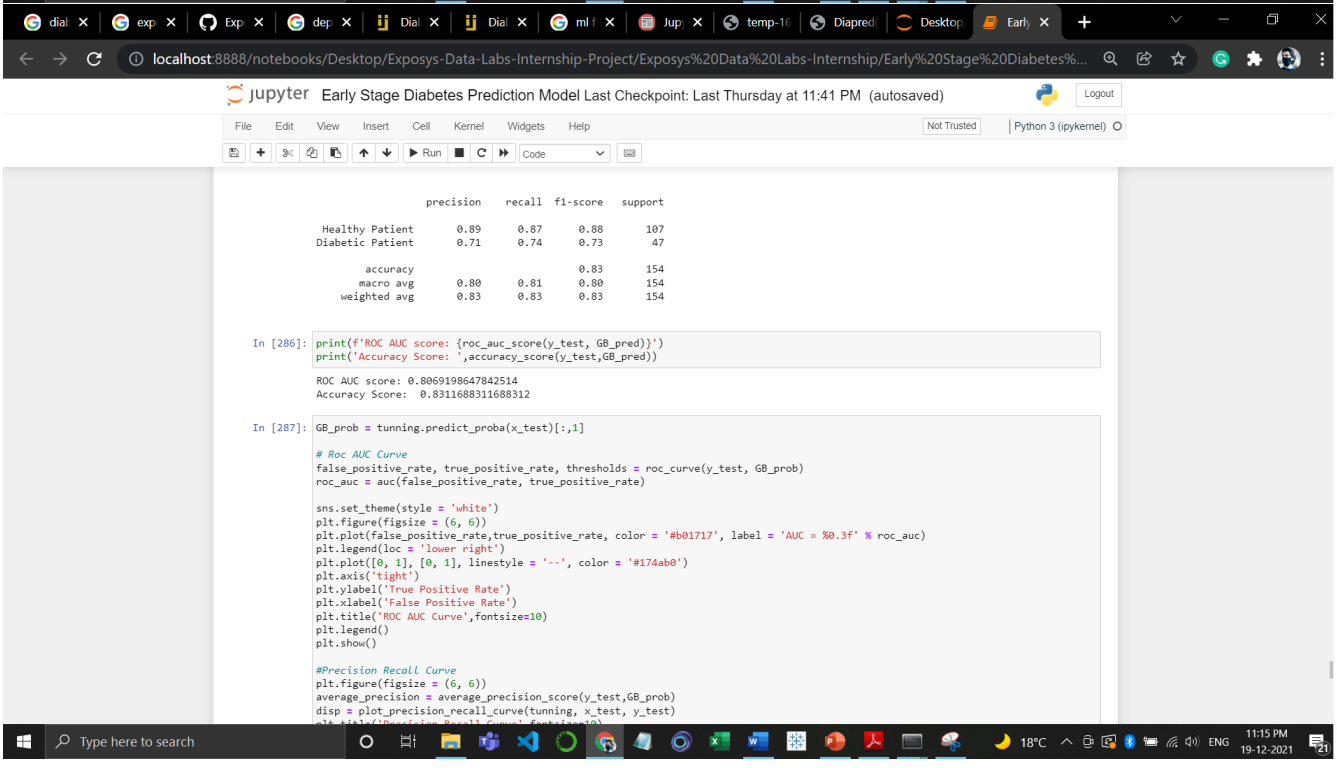


Figure (3.1) – Flow Chart

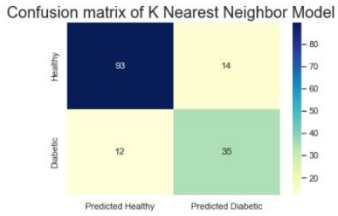
Chapter 4

Screenshots Of Project





In [285]:
```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
print('\n')
corr_matrix = confusion_matrix(y_test,GB_pred)
sns.heatmap(corr_matrix ,cmap="YlGnBu",annot=True,robust=True,linecolor='black',yticklabels = ['Healthy', 'Diabetic'], xticklabe
plt.title("Confusion matrix of K Nearest Neighbor Model " ,fontsize=20)
plt.show()
print('\n')
target_names = ['Healthy Patient','Diabetic Patient']
print(classification_report(y_test,GB_pred,target_names=target_names))
```




```
                  precision    recall  f1-score   support

 Healthy Patient       0.89      0.87      0.88       107
Diabetic Patient       0.71      0.74      0.73        47

        accuracy                           0.83       154
```


Gradient Boosting Model

In [278]:
```
GB_pred = tunning.predict(x_test)

print("Actual Outcomes : " , y_test[0:5])
print(" ")
print("Predicted Outcomes : ", GB_pred[0:5])
```

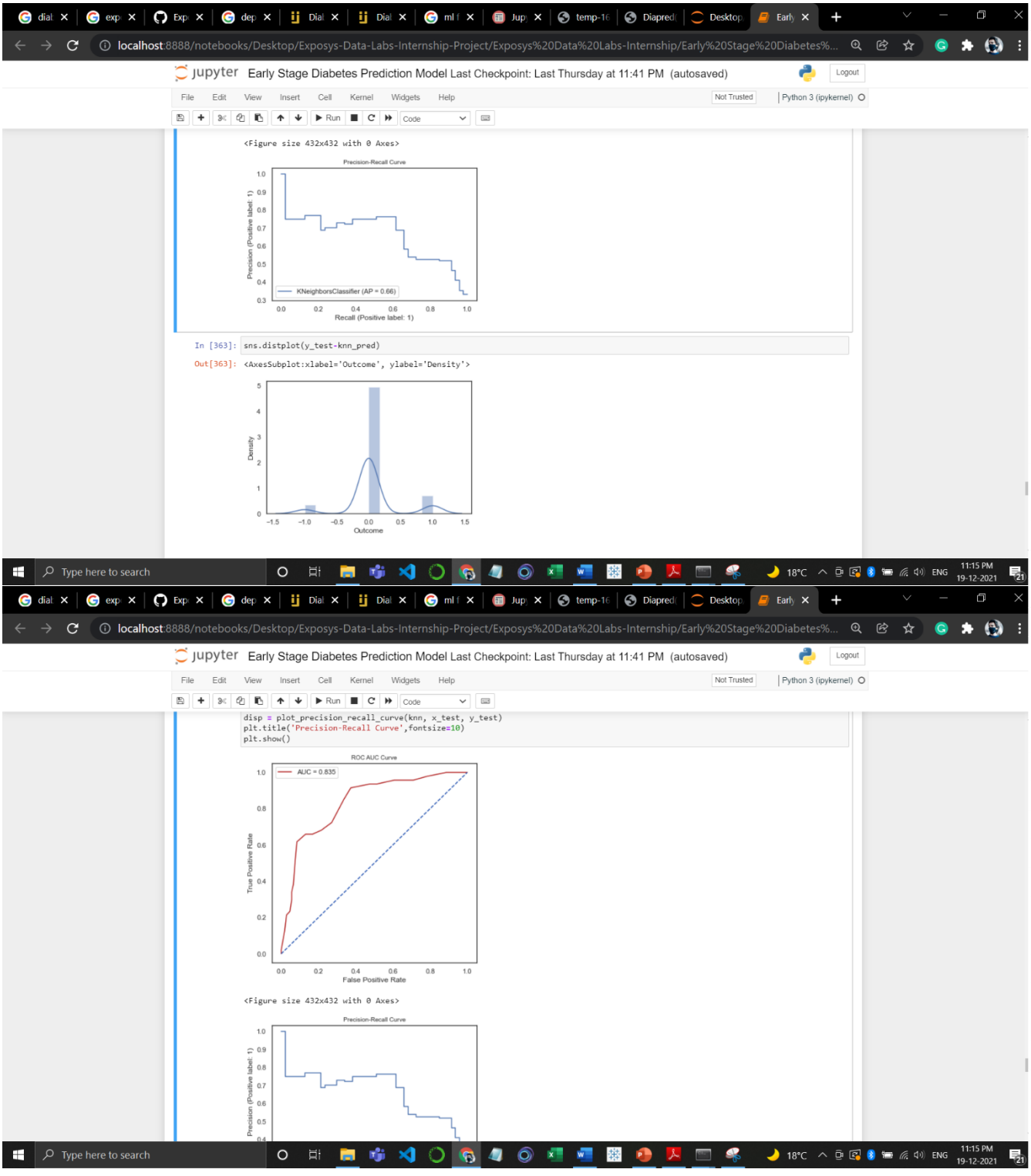
```
Actual Outcomes :  661    1
122    0
113    0
14     1
529    0
Name: Outcome, dtype: int64

Predicted Outcomes :  [1 0 0 1 0]
```

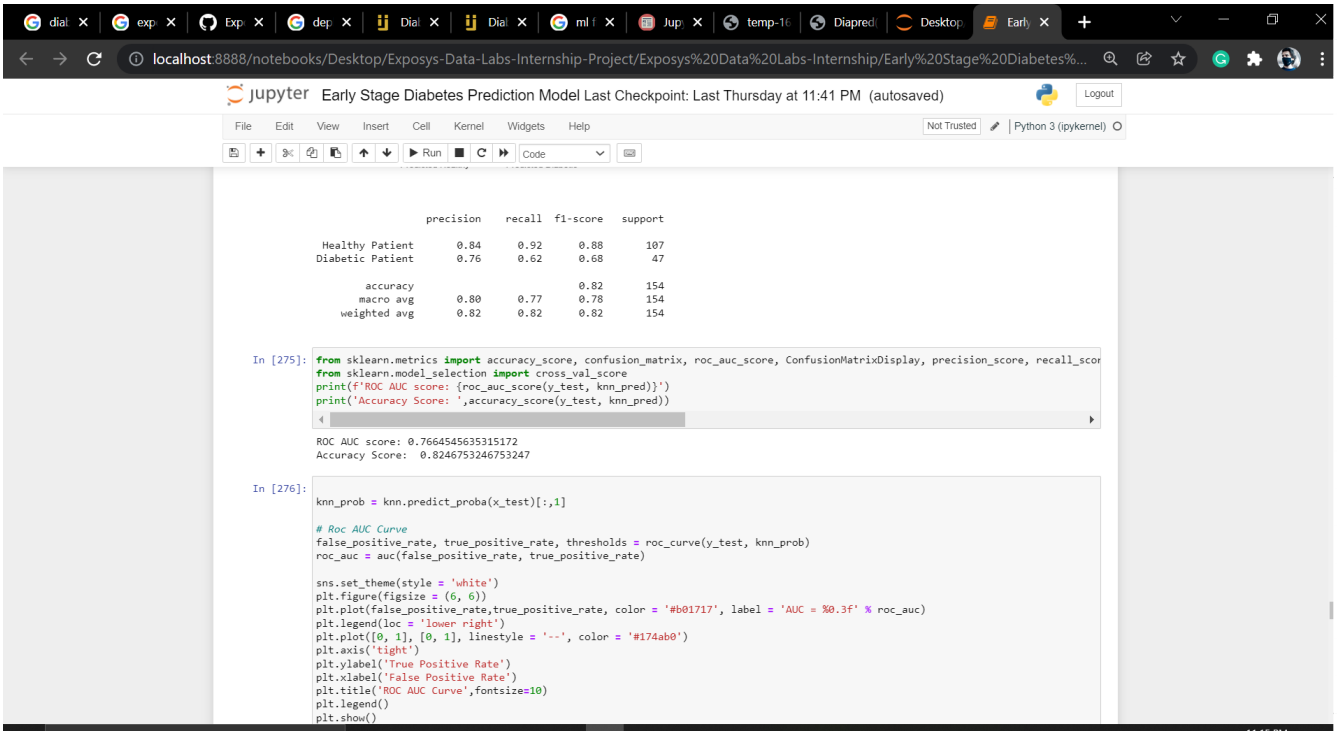
In [279]:
```
misclassified = np.where(y_test!=GB_pred)
misclassified
```

Out[279]:
```
(array([  9,  15,  21,  27,  39,  47,  48,  50,  53,  58,  59,  61,  68,
         73,  75,  86,  94,  96, 105, 111, 117, 129, 133, 137, 141, 149],
       dtype=int64),)
```

In [284]:
```
print("Total Misclassified Samples: ",len(misclassified[0]))
print(" ")
print("Actual first row outcome :",y_test[:3])
print("Predicted first row outcome :",knn_pred[:3])
```

```
Total Misclassified Samples:  26

Actual first row outcome : 661    1
122    0
113    0
Name: Outcome, dtype: int64
Predicted first row outcome : [1 0 0]
```


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
<Figure size 432x432 with 0 Axes>
Precision-Recall Curve
Precision (Positive label: 1)
KNeighborsClassifier (AP = 0.66)
Recall (Positive label: 1)
In [363]: sns.distplot(y_test-knn_pred)
Out[363]: <AxesSubplot:xlabel='Outcome', ylabel='Density'>
Density
Outcome
Type here to search
18°C
ENG
11:15 PM
19-12-2021
localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
disp = plot_precision_recall_curve(knn, x_test, y_test)
plt.title('Precision-Recall Curve',fontsize=10)
plt.show()
ROC AUC Curve
AUC = 0.835
True Positive Rate
False Positive Rate
<Figure size 432x432 with 0 Axes>
Precision-Recall Curve
Precision (Positive label: 1)
Type here to search
18°C
ENG
11:15 PM
19-12-2021

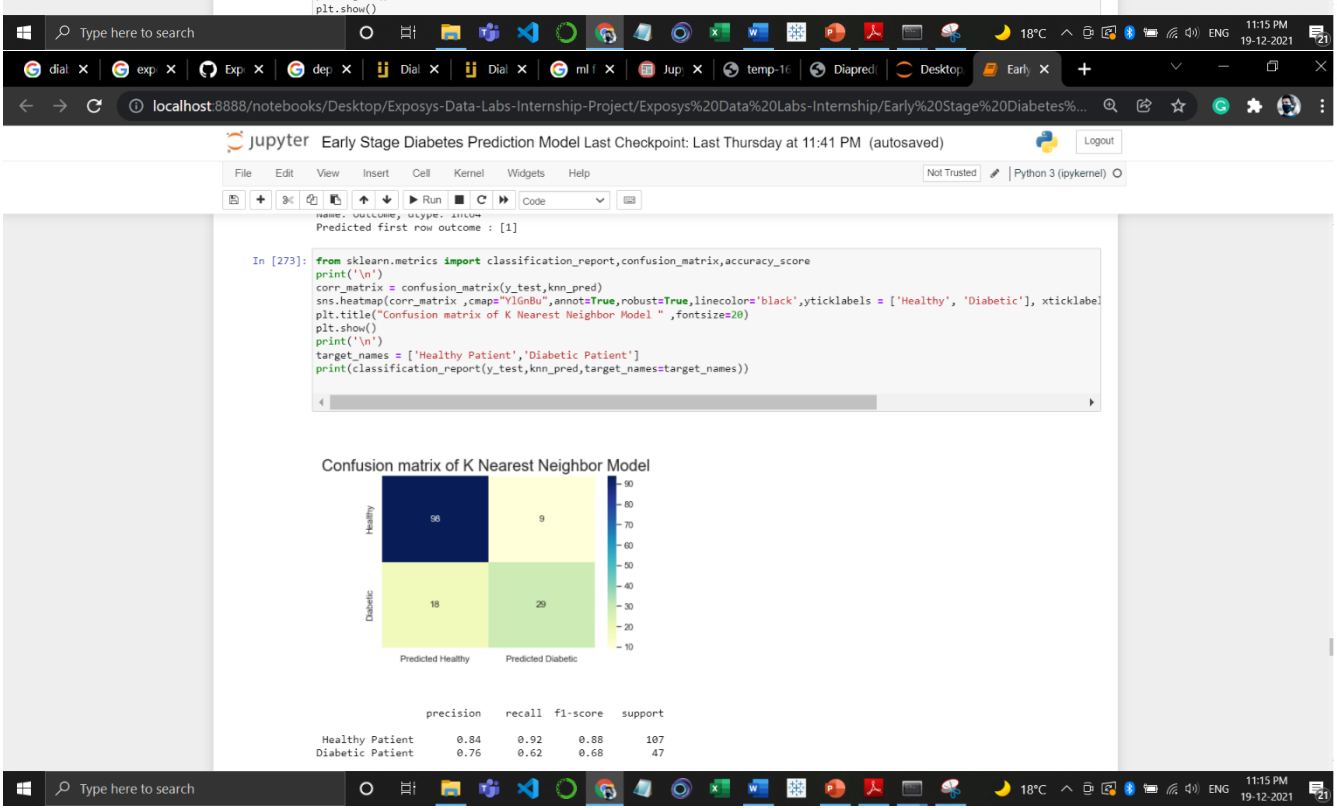


	precision	recall	f1-score	support
Healthy Patient	0.84	0.92	0.88	107
Diabetic Patient	0.76	0.62	0.68	47
accuracy			0.82	154
macro avg	0.80	0.77	0.78	154
weighted avg	0.82	0.82	0.82	154

In [275]:
```
from sklearn.metrics import accuracy_score, confusion_matrix, roc_auc_score, ConfusionMatrixDisplay, precision_score, recall_scor
from sklearn.model_selection import cross_val_score
print(f'ROC AUC score: {roc_auc_score(y_test, knn_pred)}')
print('Accuracy Score: ',accuracy_score(y_test, knn_pred))
```

```
ROC AUC score: 0.7664545635315172
Accuracy Score:  0.8246753246753247
```

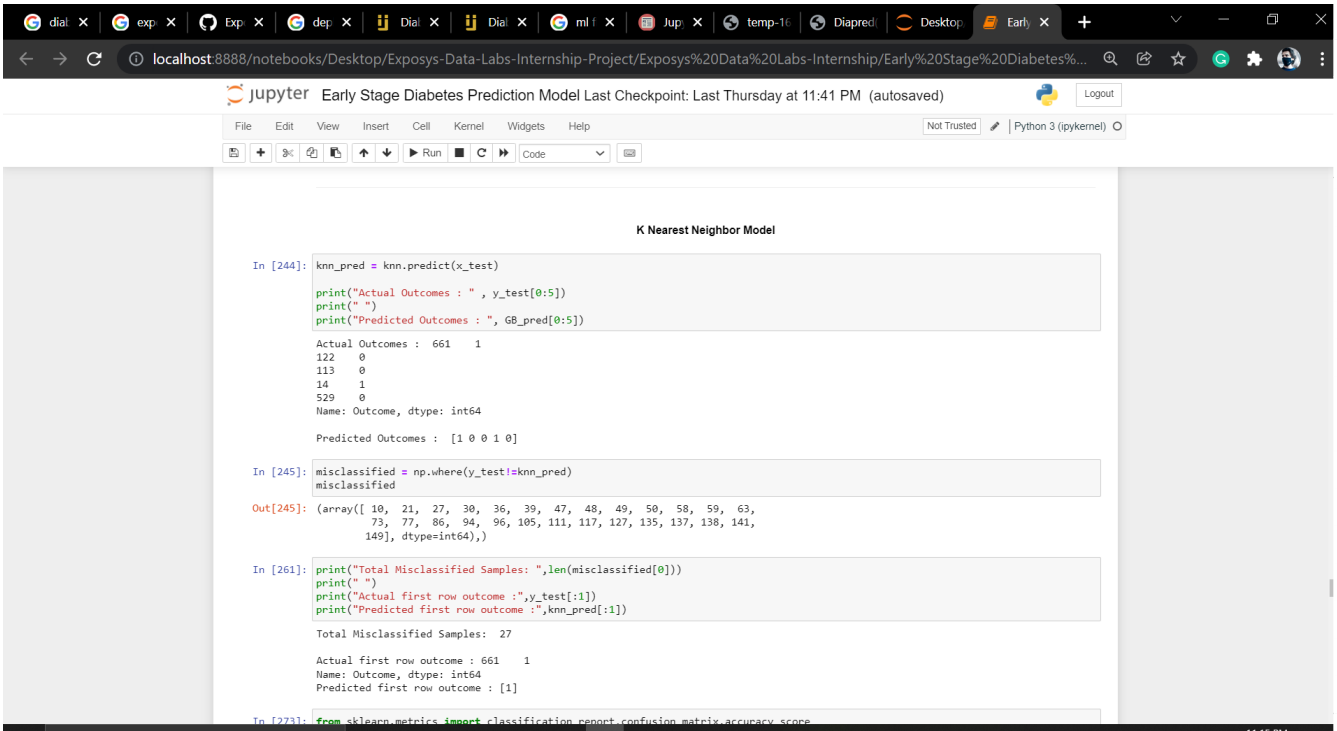
In [276]:
```
knn_prob = knn.predict_proba(x_test)[:,1]

# Roc AUC Curve
false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, knn_prob)
roc_auc = auc(false_positive_rate, true_positive_rate)

sns.set_theme(style = 'white')
plt.figure(figsize = (6, 6))
plt.plot(false_positive_rate,true_positive_rate, color = '#b01717', label = 'AUC = %0.3f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1], linestyle = '--', color = '#174ab0')
plt.axis('tight')
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.title('ROC AUC Curve',fontsize=10)
plt.legend()
plt.show()
```

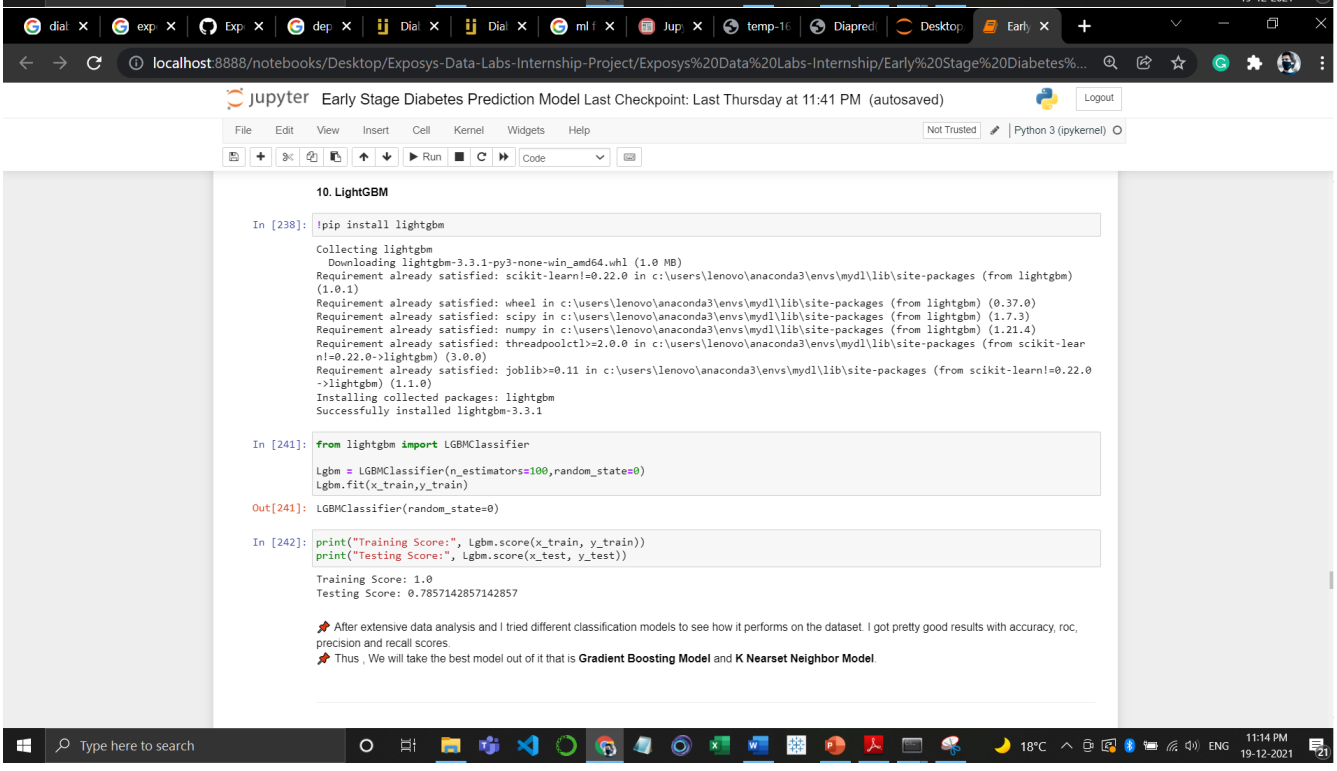


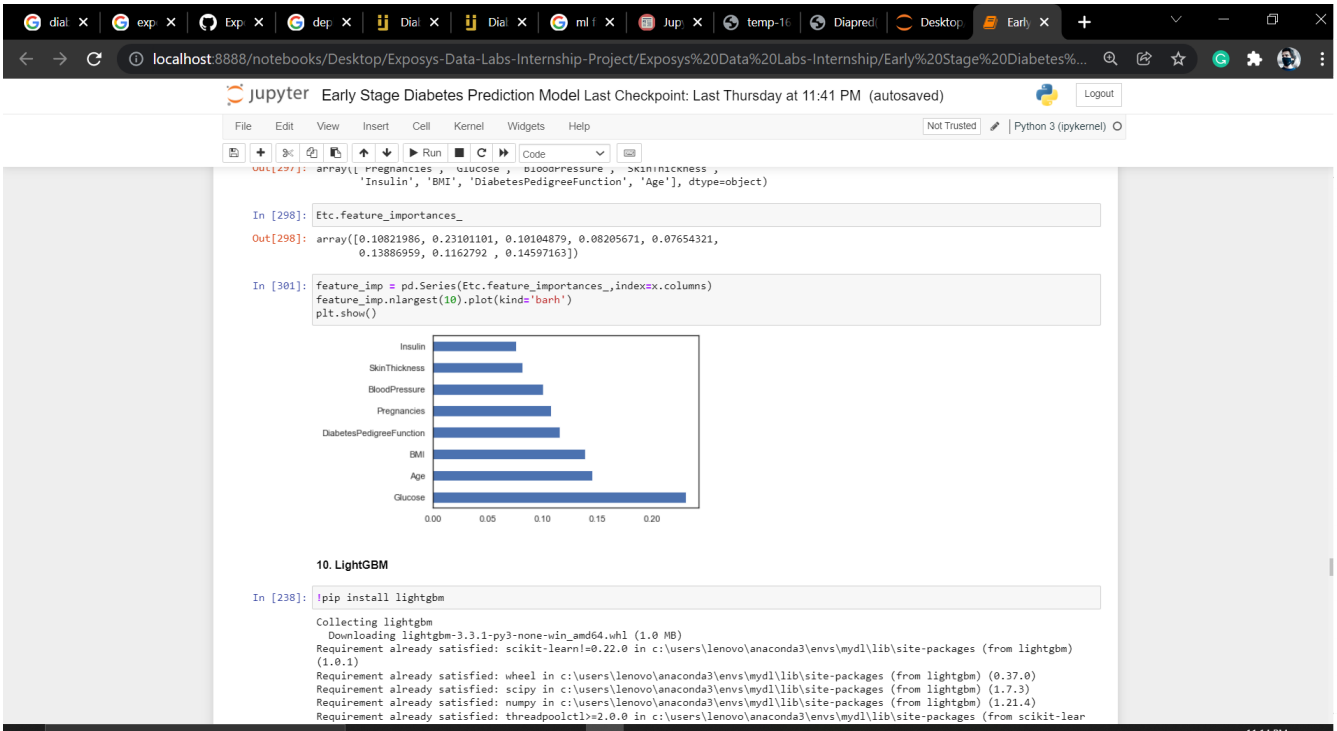
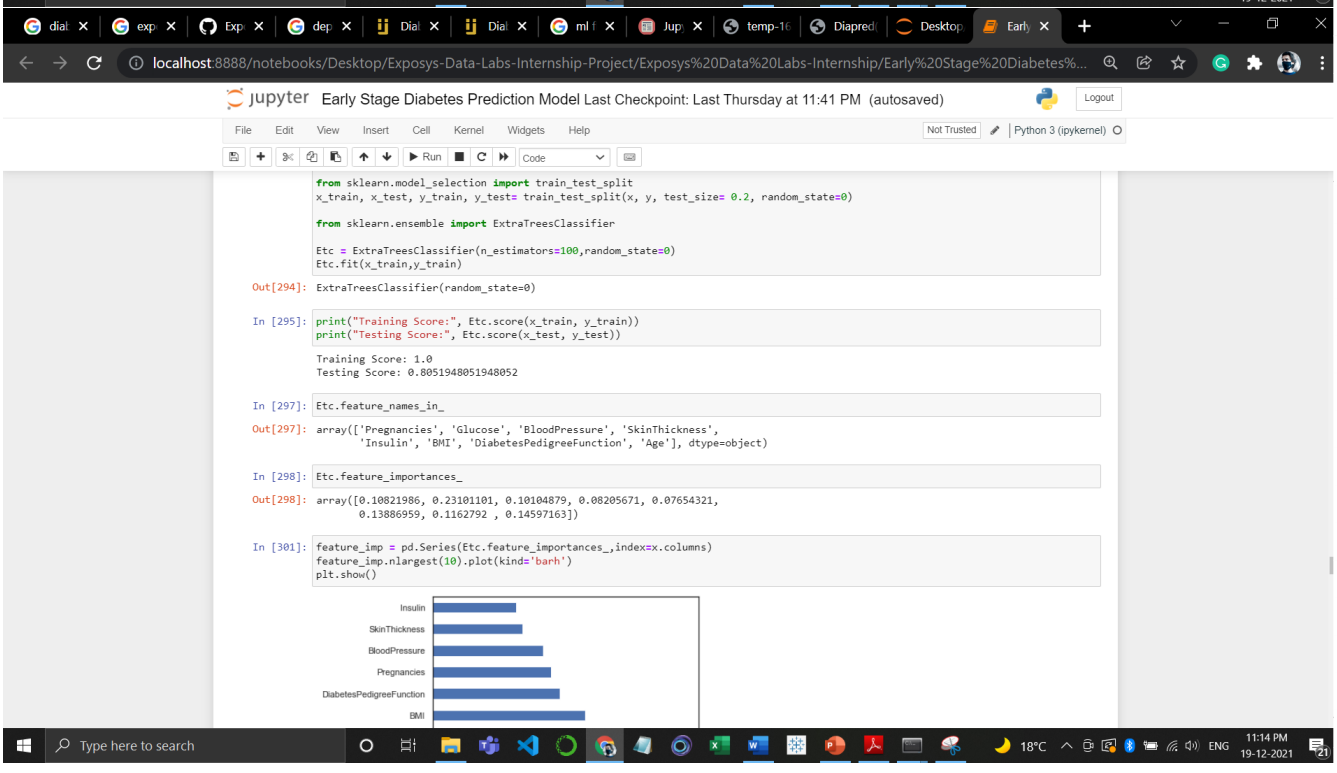

```
Predicted first row outcome : [1]
```

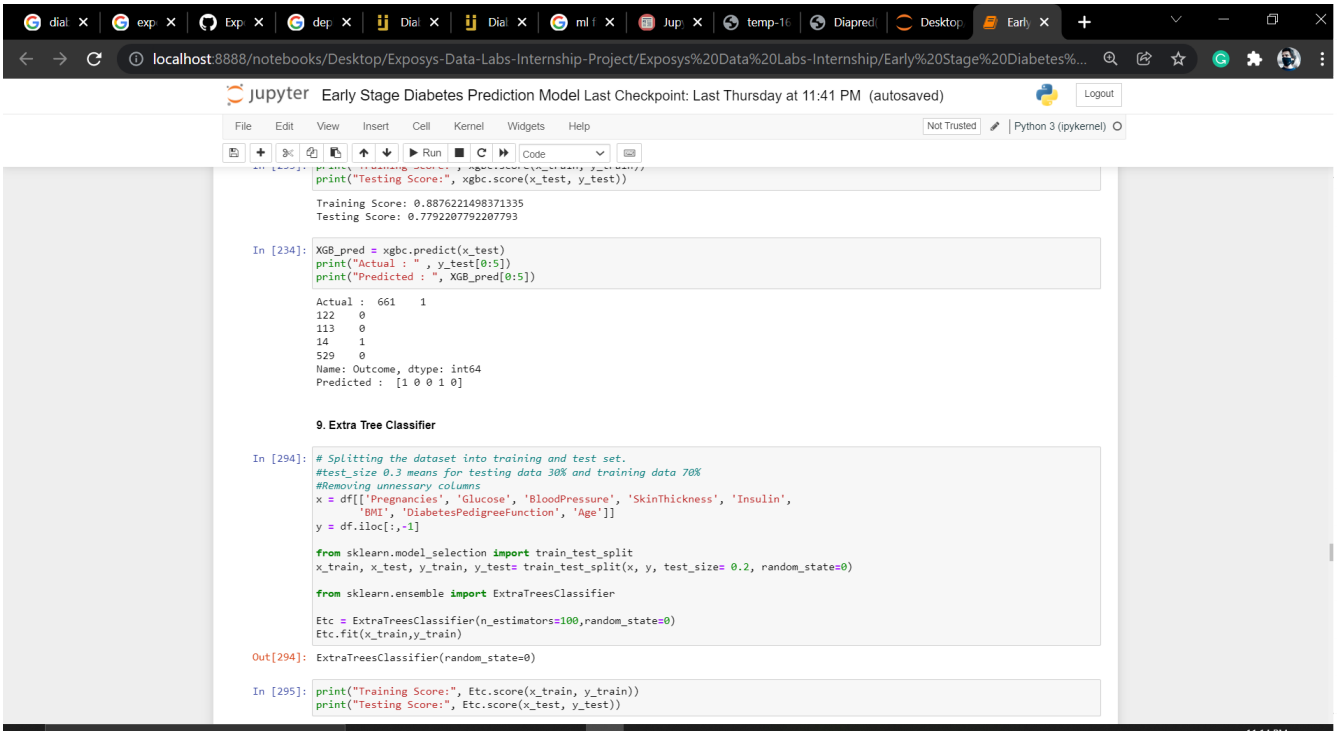
In [273]:
```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
print('\n')
corr_matrix = confusion_matrix(y_test,knn_pred)
sns.heatmap(corr_matrix ,cmap="YlGnBu",annot=True,robust=True,linecolor='black',yticklabels = ['Healthy', 'Diabetic'], xticklabel
plt.title("Confusion matrix of K Nearest Neighbor Model " ,fontsize=20)
plt.show()
print('\n')
target_names = ['Healthy Patient','Diabetic Patient']
print(classification_report(y_test,knn_pred,target_names=target_names))
```


	precision	recall	f1-score	support
Healthy Patient	0.84	0.92	0.88	107
Diabetic Patient	0.76	0.62	0.68	47


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
K Nearest Neighbor Model
In [244]:
knn_pred = knn.predict(x_test)
print("Actual Outcomes : " , y_test[0:5])
print(" ")
print("Predicted Outcomes : ", GB_pred[0:5])
Actual Outcomes : 661 1
122 0
113 0
14 1
529 0
Name: Outcome, dtype: int64
Predicted Outcomes : [1 0 0 1 0]
In [245]:
misclassified = np.where(y_test!=knn_pred)
misclassified
Out[245]:
(array([10, 21, 27, 30, 36, 39, 47, 48, 49, 50, 58, 59, 63,
73, 77, 86, 94, 96, 105, 111, 117, 127, 135, 137, 138, 141,
149], dtype=int64),)
In [261]:
print("Total Misclassified Samples: ",len(misclassified[0]))
print(" ")
print("Actual first row outcome :",y_test[:1])
print("Predicted first row outcome :",knn_pred[:1])
Total Misclassified Samples: 27
Actual first row outcome : 661 1
Name: Outcome, dtype: int64
Predicted first row outcome : [1]
In [273]:
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
Type here to search
18°C
ENG
11:15 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
10. LightGBM
In [238]:
!pip install lightgbm
Collecting lightgbm
Downloading lightgbm-3.3.1-py3-none-win_amd64.whl (1.0 MB)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm) (1.0.1)
Requirement already satisfied: wheel in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm) (0.37.0)
Requirement already satisfied: scipy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm) (1.7.3)
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm) (1.21.4)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (3.0.0)
Requirement already satisfied: joblib>=0.11 in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (1.1.0)
Installing collected packages: lightgbm
Successfully installed lightgbm-3.3.1
In [241]:
from lightgbm import LGBMClassifier
Lgbm = LGBMClassifier(n_estimators=100,random_state=0)
Lgbm.fit(x_train,y_train)
Out[241]:
LGBMClassifier(random_state=0)
In [242]:
print("Training Score:", Lgbm.score(x_train, y_train))
print("Testing Score:", Lgbm.score(x_test, y_test))
Training Score: 1.0
Testing Score: 0.7857142857142857
After extensive data analysis and I tried different classification models to see how it performs on the dataset. I got pretty good results with accuracy, roc, precision and recall scores.
Thus , We will take the best model out of it that is Gradient Boosting Model and K Nearset Neighbor Model.
Type here to search
18°C
ENG
11:14 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age'], dtype=object)
In [298]:
Etc.feature_importances_
Out[298]:
array([0.10821986, 0.23101101, 0.10104879, 0.08205671, 0.07654321,
0.13886959, 0.1162792 , 0.14597163])
In [301]:
feature_imp = pd.Series(Etc.feature_importances_,index=x.columns)
feature_imp.nlargest(10).plot(kind='barh')
plt.show()
Insulin
SkinThickness
BloodPressure
Pregnancies
DiabetesPedigreeFunction
BMI
Age
Glucose
0.00
0.05
0.10
0.15
0.20
10. LightGBM
In [238]:
!pip install lightgbm
Collecting lightgbm
Downloading lightgbm-3.3.1-py3-none-win_amd64.whl (1.0 MB)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm)
(1.0.1)
Requirement already satisfied: wheel in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm) (0.37.0)
Requirement already satisfied: scipy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm) (1.7.3)
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from lightgbm) (1.21.4)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from scikit-lear
Type here to search
18°C
ENG
11:14 PM
19-12-2021

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
from sklearn.ensemble import ExtraTreesClassifier
Etc = ExtraTreesClassifier(n_estimators=100,random_state=0)
Etc.fit(x_train,y_train)
Out[294]:
ExtraTreesClassifier(random_state=0)
In [295]:
print("Training Score:", Etc.score(x_train, y_train))
print("Testing Score:", Etc.score(x_test, y_test))
Training Score: 1.0
Testing Score: 0.8051948051948052
In [297]:
Etc.feature_names_in_
Out[297]:
array(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age'], dtype=object)
In [298]:
Etc.feature_importances_
Out[298]:
array([0.10821986, 0.23101101, 0.10104879, 0.08205671, 0.07654321,
0.13886959, 0.1162792 , 0.14597163])
In [301]:
feature_imp = pd.Series(Etc.feature_importances_,index=x.columns)
feature_imp.nlargest(10).plot(kind='barh')
plt.show()
Insulin
SkinThickness
BloodPressure
Pregnancies
DiabetesPedigreeFunction
BMI
Type here to search
18°C
ENG
11:14 PM
19-12-2021



jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)

```
print("Testing Score:", xgbc.score(x_test, y_test))
```

```
Training Score: 0.8876221498371335
Testing Score: 0.7792207792207793
```

In [234]:
```
XGB_pred = xgbc.predict(x_test)
print("Actual : " , y_test[0:5])
print("Predicted : ", XGB_pred[0:5])
```

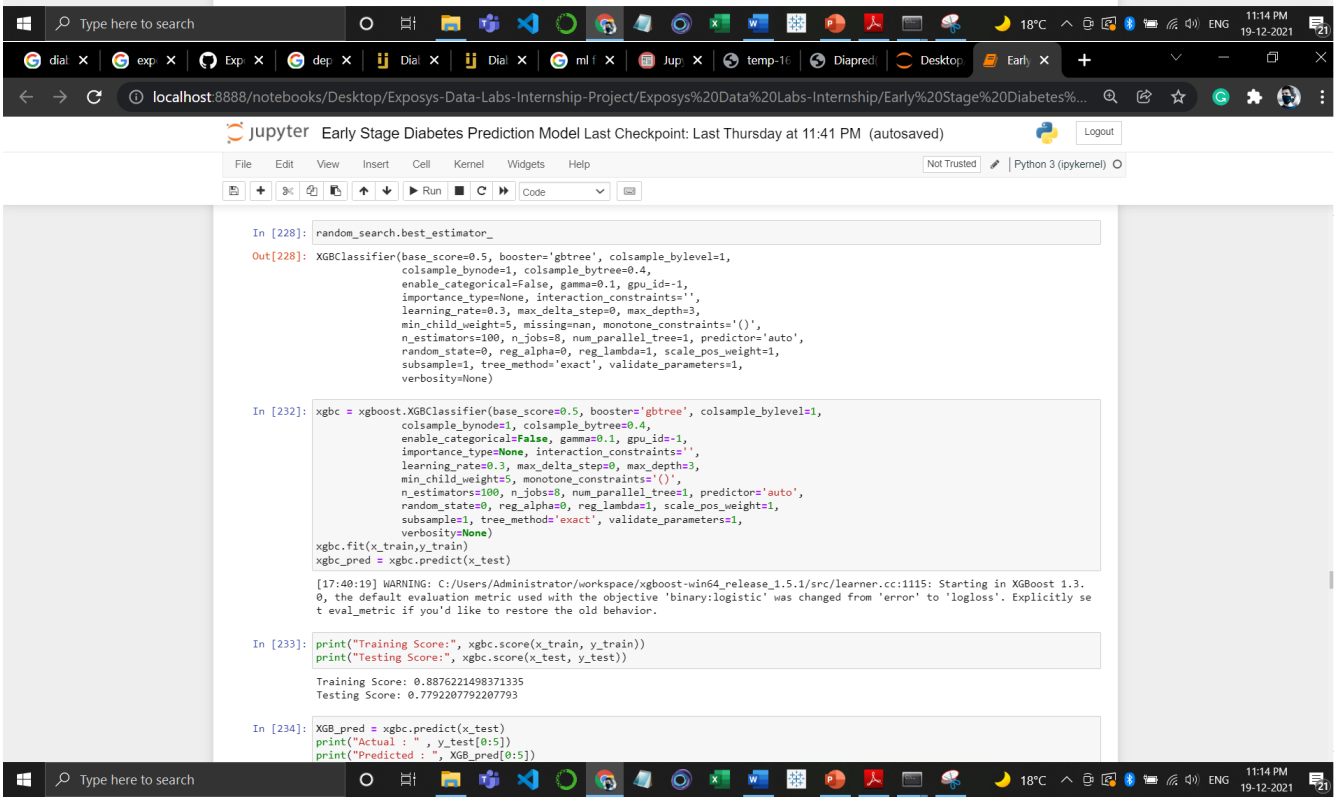
```
Actual :  661    1
122    0
113    0
14     1
529    0
Name: Outcome, dtype: int64
Predicted :  [1 0 0 1 0]
```

9. Extra Tree Classifier

In [294]:
```
# Splitting the dataset into training and test set.
#test_size 0.3 means for testing data 30% and training data 70%
#Removing unnessary columns
x = df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age']]
y = df.iloc[:,-1]

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)

from sklearn.ensemble import ExtraTreesClassifier

Etc = ExtraTreesClassifier(n_estimators=100,random_state=0)
Etc.fit(x_train,y_train)
```

Out[294]:
```
ExtraTreesClassifier(random_state=0)
```

In [295]:
```
print("Training Score:", Etc.score(x_train, y_train))
print("Testing Score:", Etc.score(x_test, y_test))
```



In [228]:
```
random_search.best_estimator_
```

Out[228]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.4,
              enable_categorical=False, gamma=0.1, gpu_id=-1,
              importance_type=None, interaction_constraints='',
              learning_rate=0.3, max_delta_step=0, max_depth=3,
              min_child_weight=5, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=8, num_parallel_tree=1, predictor='auto',
              random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
              subsample=1, tree_method='exact', validate_parameters=1,
              verbosity=None)
```

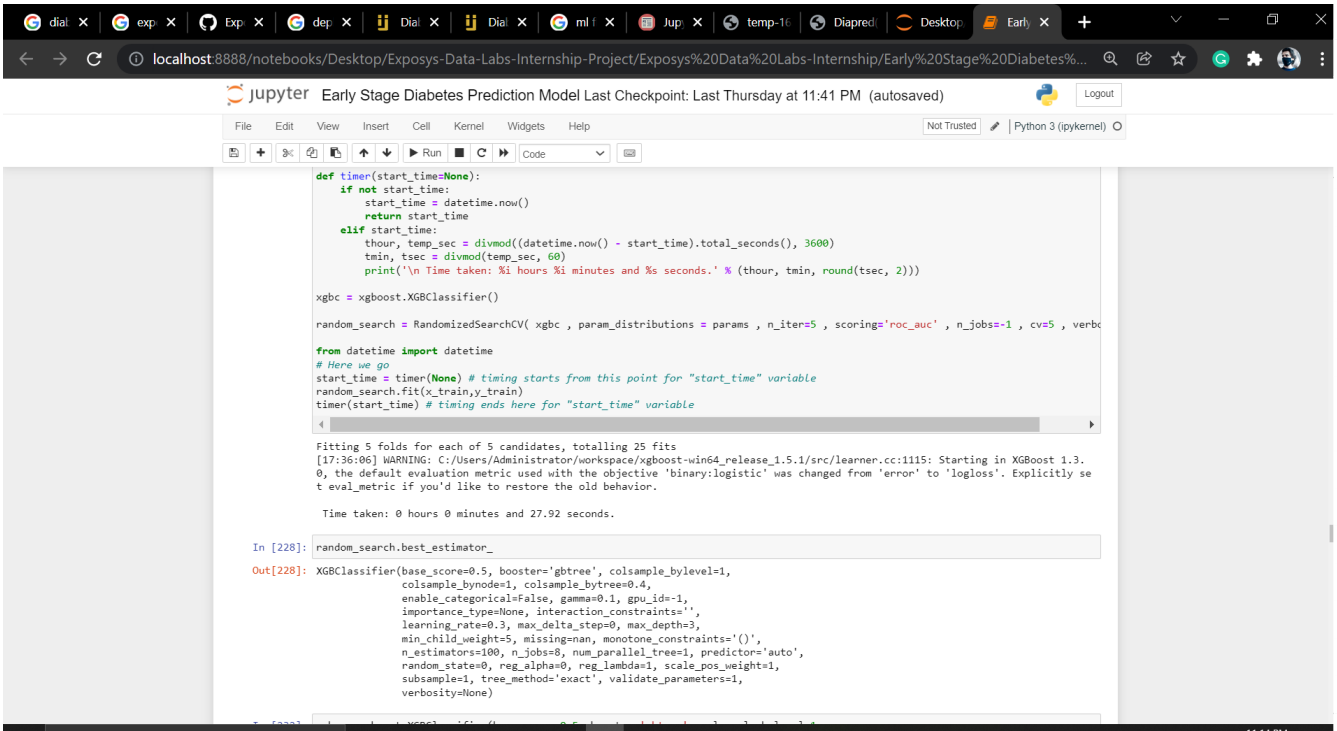
In [232]:
```
xgbc = xgboost.XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.4,
              enable_categorical=False, gamma=0.1, gpu_id=-1,
              importance_type=None, interaction_constraints='',
              learning_rate=0.3, max_delta_step=0, max_depth=3,
              min_child_weight=5, monotone_constraints='()',
              n_estimators=100, n_jobs=8, num_parallel_tree=1, predictor='auto',
              random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
              subsample=1, tree_method='exact', validate_parameters=1,
              verbosity=None)
xgbc.fit(x_train,y_train)
xgbc_pred = xgbc.predict(x_test)
```

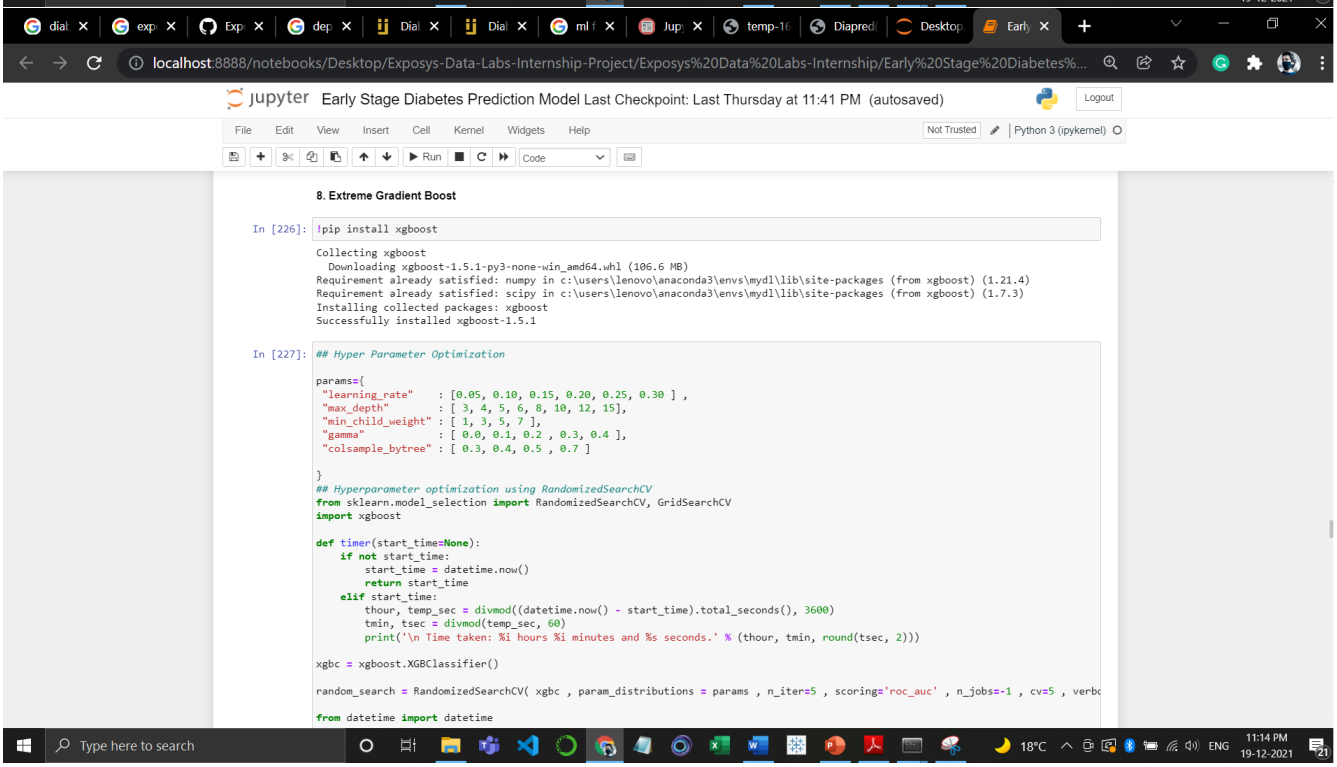
```
[17:40:19] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.
0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly se
t eval_metric if you'd like to restore the old behavior.
```

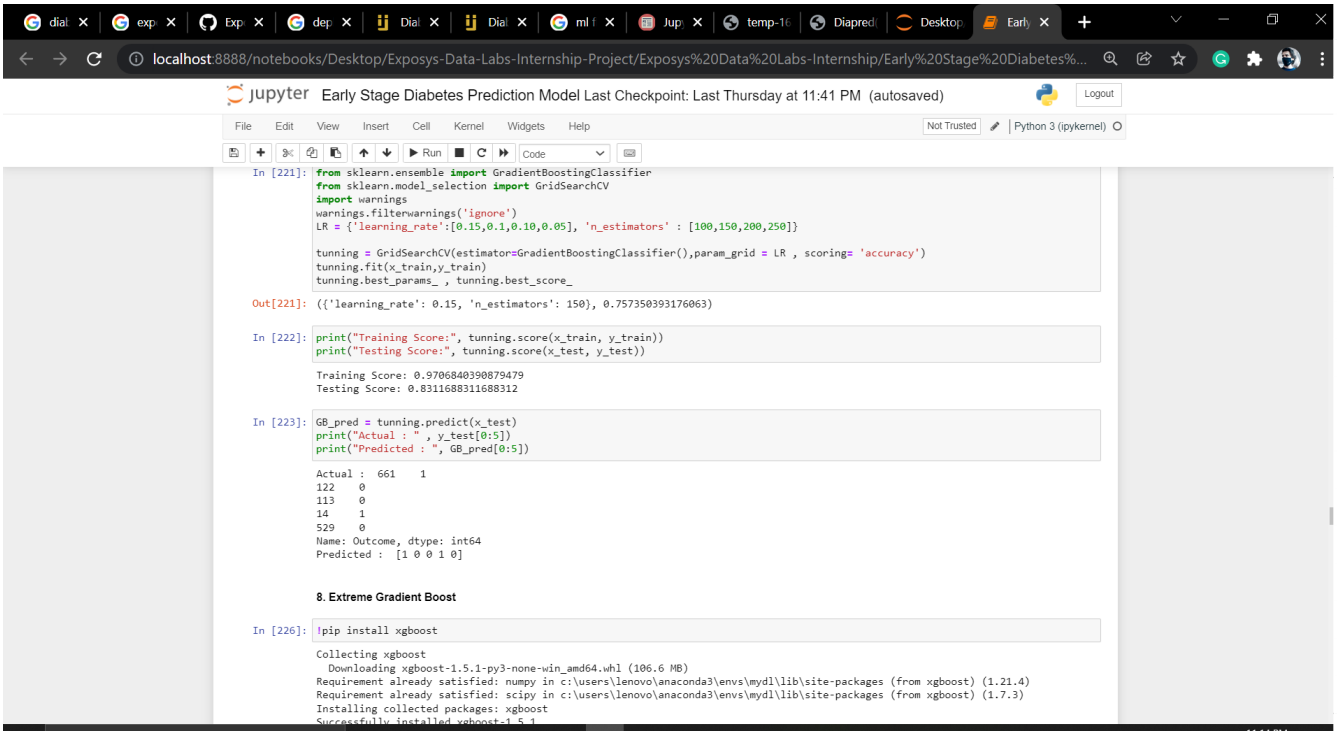
In [233]:
```
print("Training Score:", xgbc.score(x_train, y_train))
print("Testing Score:", xgbc.score(x_test, y_test))
```

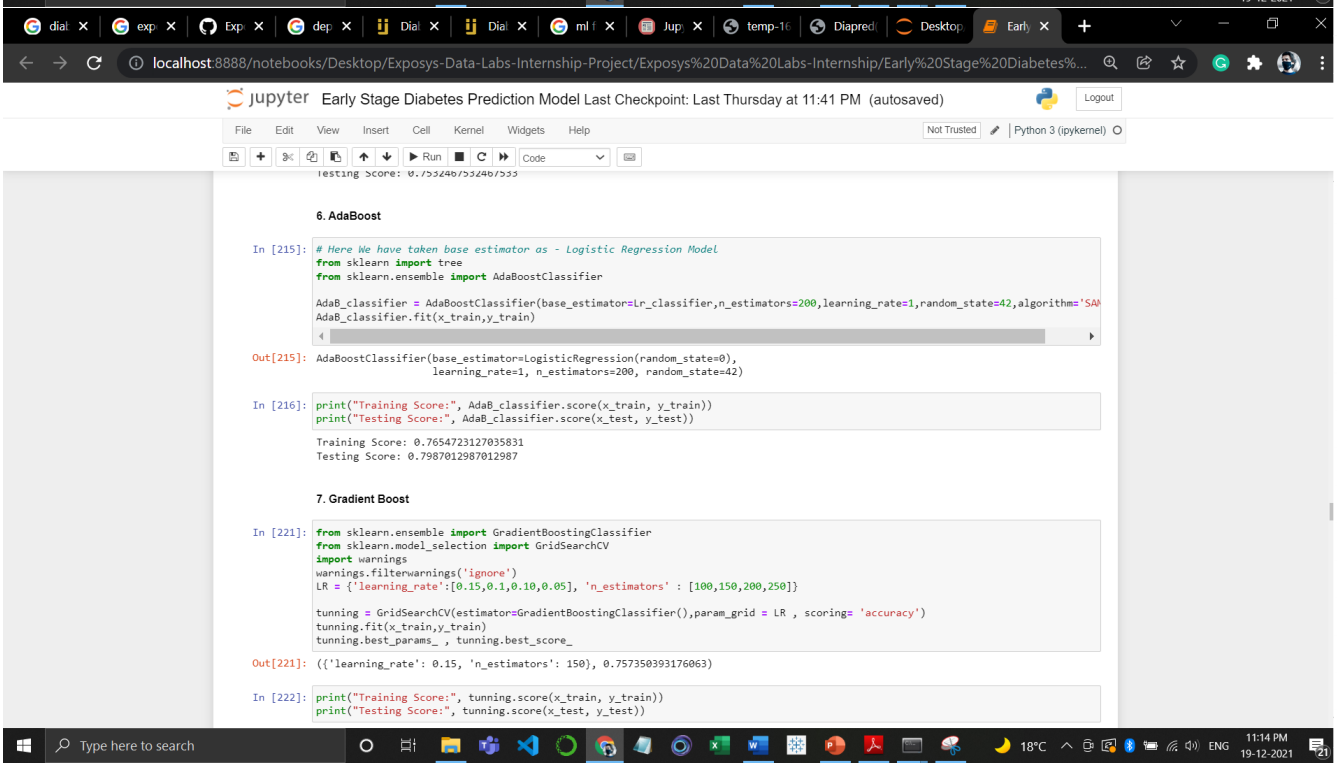
```
Training Score: 0.8876221498371335
Testing Score: 0.7792207792207793
```

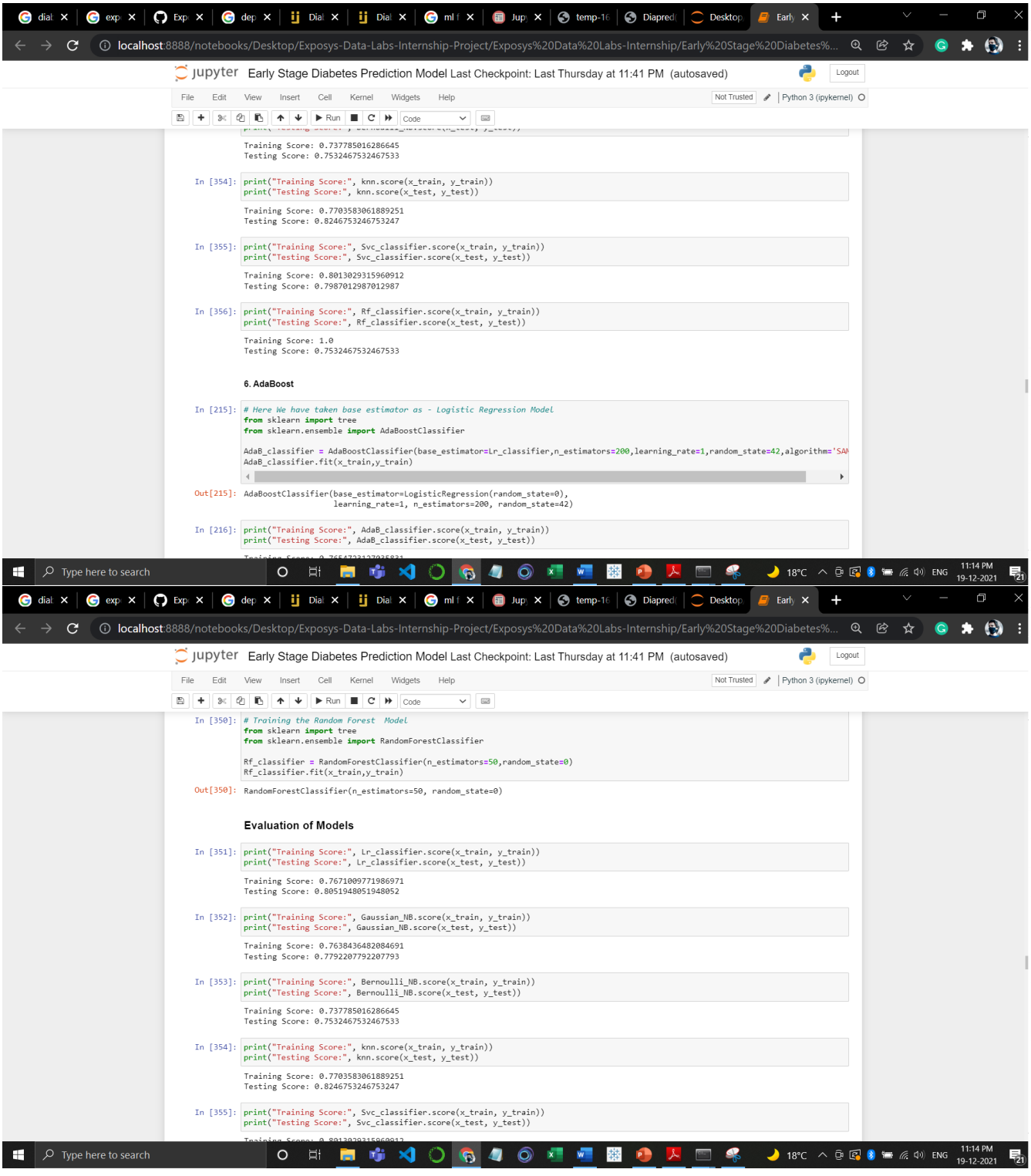
In [234]:
```
XGB_pred = xgbc.predict(x_test)
print("Actual : " , y_test[0:5])
print("Predicted : ", XGB_pred[0:5])
```


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
def timer(start_time=None):
if not start_time:
start_time = datetime.now()
return start_time
elif start_time:
thour, temp_sec = divmod((datetime.now() - start_time).total_seconds(), 3600)
tmin, tsec = divmod(temp_sec, 60)
print('\n Time taken: %i hours %i minutes and %s seconds.' % (thour, tmin, round(tsec, 2)))
xgbc = xgboost.XGBClassifier()
random_search = RandomizedSearchCV(xgbc , param_distributions = params , n_iter=5 , scoring='roc_auc' , n_jobs=-1 , cv=5 , verbo
from datetime import datetime
# Here we go
start_time = timer(None) # timing starts from this point for "start_time" variable
random_search.fit(x_train,y_train)
timer(start_time) # timing ends here for "start_time" variable
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[17:36:06] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.
0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly se
t eval_metric if you'd like to restore the old behavior.
Time taken: 0 hours 0 minutes and 27.92 seconds.
In [228]:
random_search.best_estimator_
Out[228]:
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
colsample_bynode=1, colsample_bytree=0.4,
enable_categorical=False, gamma=0.1, gpu_id=-1,
importance_type=None, interaction_constraints='',
learning_rate=0.3, max_delta_step=0, max_depth=3,
min_child_weight=5, missing=nan, monotone_constraints='()',
n_estimators=100, n_jobs=8, num_parallel_tree=1, predictor='auto',
random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
subsample=1, tree_method='exact', validate_parameters=1,
verbosity=None)
Type here to search
18°C
ENG
11:14 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
8. Extreme Gradient Boost
In [226]:
!pip install xgboost
Collecting xgboost
Downloading xgboost-1.5.1-py3-none-win_amd64.whl (106.6 MB)
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from xgboost) (1.21.4)
Requirement already satisfied: scipy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from xgboost) (1.7.3)
Installing collected packages: xgboost
Successfully installed xgboost-1.5.1
In [227]:
Hyper Parameter Optimization
params={
"learning_rate" : [0.05, 0.10, 0.15, 0.20, 0.25, 0.30] ,
"max_depth" : [3, 4, 5, 6, 8, 10, 12, 15],
"min_child_weight" : [1, 3, 5, 7],
"gamma" : [0.0, 0.1, 0.2 , 0.3, 0.4],
"colsample_bytree" : [0.3, 0.4, 0.5 , 0.7]
}
Hyperparameter optimization using RandomizedSearchCV
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
import xgboost
def timer(start_time=None):
if not start_time:
start_time = datetime.now()
return start_time
elif start_time:
thour, temp_sec = divmod((datetime.now() - start_time).total_seconds(), 3600)
tmin, tsec = divmod(temp_sec, 60)
print('\n Time taken: %i hours %i minutes and %s seconds.' % (thour, tmin, round(tsec, 2)))
xgbc = xgboost.XGBClassifier()
random_search = RandomizedSearchCV(xgbc , param_distributions = params , n_iter=5 , scoring='roc_auc' , n_jobs=-1 , cv=5 , verbo
from datetime import datetime
Type here to search
18°C
ENG
11:14 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [221]:
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV
import warnings
warnings.filterwarnings('ignore')
LR = {'learning_rate':[0.15,0.1,0.10,0.05], 'n_estimators' : [100,150,200,250]}
tunning = GridSearchCV(estimator=GradientBoostingClassifier(),param_grid = LR , scoring= 'accuracy')
tunning.fit(x_train,y_train)
tunning.best_params_ , tunning.best_score_
Out[221]:
({'learning_rate': 0.15, 'n_estimators': 150}, 0.757350393176063)
In [222]:
print("Training Score:", tunning.score(x_train, y_train))
print("Testing Score:", tunning.score(x_test, y_test))
Training Score: 0.9706840390879479
Testing Score: 0.8311688311688312
In [223]:
GB_pred = tunning.predict(x_test)
print("Actual : " , y_test[0:5])
print("Predicted : ", GB_pred[0:5])
Actual : 661 1
122 0
113 0
14 1
529 0
Name: Outcome, dtype: int64
Predicted : [1 0 0 1 0]
8. Extreme Gradient Boost
In [226]:
!pip install xgboost
Collecting xgboost
Downloading xgboost-1.5.1-py3-none-win_amd64.whl (106.6 MB)
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from xgboost) (1.21.4)
Requirement already satisfied: scipy in c:\users\lenovo\anaconda3\envs\mydl\lib\site-packages (from xgboost) (1.7.3)
Installing collected packages: xgboost
Type here to search
18°C
ENG
11:14 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
6. AdaBoost
In [215]:
# Here We have taken base estimator as - Logistic Regression Model
from sklearn import tree
from sklearn.ensemble import AdaBoostClassifier
AdaB_classifier = AdaBoostClassifier(base_estimator=Lr_classifier,n_estimators=200,learning_rate=1,random_state=42,algorithm='SA
AdaB_classifier.fit(x_train,y_train)
Out[215]:
AdaBoostClassifier(base_estimator=LogisticRegression(random_state=0),
learning_rate=1, n_estimators=200, random_state=42)
In [216]:
print("Training Score:", AdaB_classifier.score(x_train, y_train))
print("Testing Score:", AdaB_classifier.score(x_test, y_test))
Training Score: 0.7654723127035831
Testing Score: 0.7987012987012987
7. Gradient Boost
In [221]:
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV
import warnings
warnings.filterwarnings('ignore')
LR = {'learning_rate':[0.15,0.1,0.10,0.05], 'n_estimators' : [100,150,200,250]}
tunning = GridSearchCV(estimator=GradientBoostingClassifier(),param_grid = LR , scoring= 'accuracy')
tunning.fit(x_train,y_train)
tunning.best_params_ , tunning.best_score_
Out[221]:
({'learning_rate': 0.15, 'n_estimators': 150}, 0.757350393176063)
In [222]:
print("Training Score:", tunning.score(x_train, y_train))
print("Testing Score:", tunning.score(x_test, y_test))
Type here to search
18°C
ENG
11:14 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Training Score: 0.737785016286645
Testing Score: 0.7532467532467533
In [354]:
print("Training Score:", knn.score(x_train, y_train))
print("Testing Score:", knn.score(x_test, y_test))
Training Score: 0.7703583061889251
Testing Score: 0.8246753246753247
In [355]:
print("Training Score:", Svc_classifier.score(x_train, y_train))
print("Testing Score:", Svc_classifier.score(x_test, y_test))
Training Score: 0.8013029315960912
Testing Score: 0.798701298701298
In [356]:
print("Training Score:", Rf_classifier.score(x_train, y_train))
print("Testing Score:", Rf_classifier.score(x_test, y_test))
Training Score: 1.0
Testing Score: 0.7532467532467533
6. AdaBoost
In [215]:
# Here We have taken base estimator as - Logistic Regression Model
from sklearn import tree
from sklearn.ensemble import AdaBoostClassifier
AdaB_classifier = AdaBoostClassifier(base_estimator=Lr_classifier,n_estimators=200,learning_rate=1,random_state=42,algorithm='SA
AdaB_classifier.fit(x_train,y_train)
Out[215]:
AdaBoostClassifier(base_estimator=LogisticRegression(random_state=0),
learning_rate=1, n_estimators=200, random_state=42)
In [216]:
print("Training Score:", AdaB_classifier.score(x_train, y_train))
print("Testing Score:", AdaB_classifier.score(x_test, y_test))
Type here to search
18°C
ENG
11:14 PM
19-12-2021
localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [350]:
# Training the Random Forest Model
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
Rf_classifier = RandomForestClassifier(n_estimators=50,random_state=0)
Rf_classifier.fit(x_train,y_train)
Out[350]:
RandomForestClassifier(n_estimators=50, random_state=0)
Evaluation of Models
In [351]:
print("Training Score:", Lr_classifier.score(x_train, y_train))
print("Testing Score:", Lr_classifier.score(x_test, y_test))
Training Score: 0.7671009771986971
Testing Score: 0.8051948051948052
In [352]:
print("Training Score:", Gaussian_NB.score(x_train, y_train))
print("Testing Score:", Gaussian_NB.score(x_test, y_test))
Training Score: 0.7638436482084691
Testing Score: 0.7792207792207793
In [353]:
print("Training Score:", Bernoulli_NB.score(x_train, y_train))
print("Testing Score:", Bernoulli_NB.score(x_test, y_test))
Training Score: 0.737785016286645
Testing Score: 0.7532467532467533
In [354]:
print("Training Score:", knn.score(x_train, y_train))
print("Testing Score:", knn.score(x_test, y_test))
Training Score: 0.7703583061889251
Testing Score: 0.8246753246753247
In [355]:
print("Training Score:", Svc_classifier.score(x_train, y_train))
print("Testing Score:", Svc_classifier.score(x_test, y_test))
Type here to search
18°C
ENG
11:14 PM
19-12-2021

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...

jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [348]:
```
# NOW WITH K=30
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=24, metric = 'minkowski')

knn.fit(x_train,y_train)
```

Out[348]:
```
KNeighborsClassifier(n_neighbors=24)
```

4. Support Vector Machine

In [349]:
```
# Training the SVM Model
from sklearn.svm import SVC
Svc_classifier = SVC()
Svc_classifier.fit(x_train,y_train)
```

Out[349]:
```
SVC()
```

Ensemble Learning -

5. Random Forest Classifier

In [350]:
```
# Training the Random Forest  Model
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier

Rf_classifier = RandomForestClassifier(n_estimators=50,random_state=0)
Rf_classifier.fit(x_train,y_train)
```

Out[350]:
```
RandomForestClassifier(n_estimators=50, random_state=0)
```

Evaluation of Models

Type here to search

18°C ENG 11:14 PM 19-12-2021

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...

jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

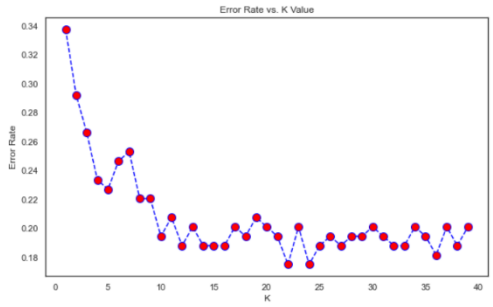
Not Trusted | Python 3 (ipykernel)

Run Code

```
    error_rate.append(np.mean(pred_i != y_test))
```

In [346]:
```
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

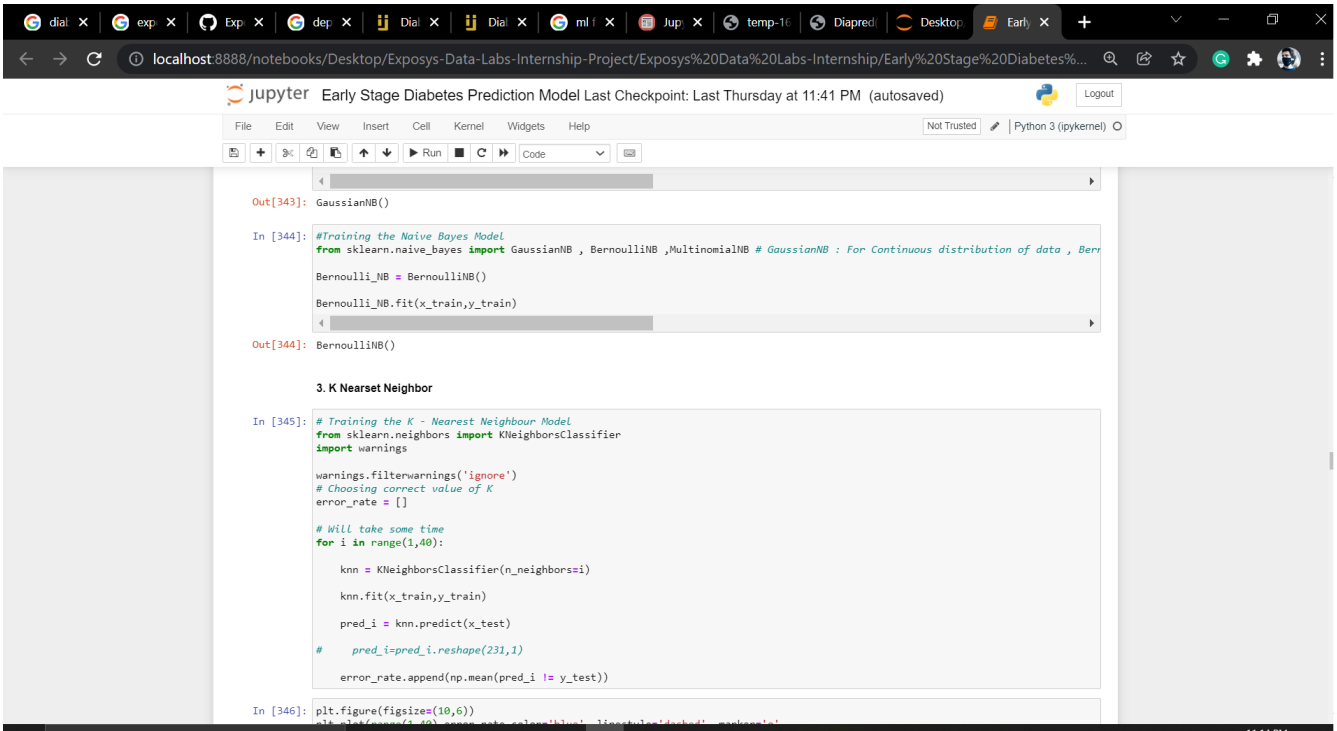
Out[346]:
```
Text(0, 0.5, 'Error Rate')
```

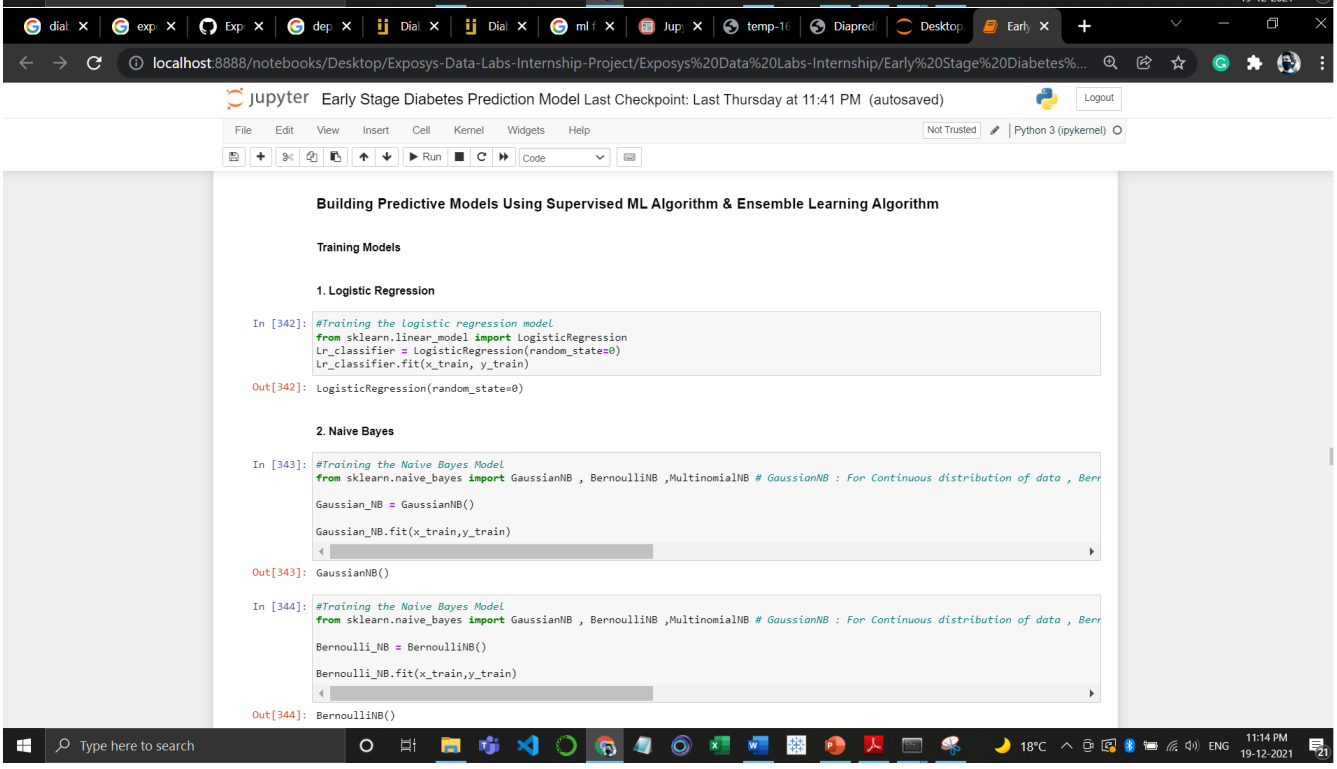


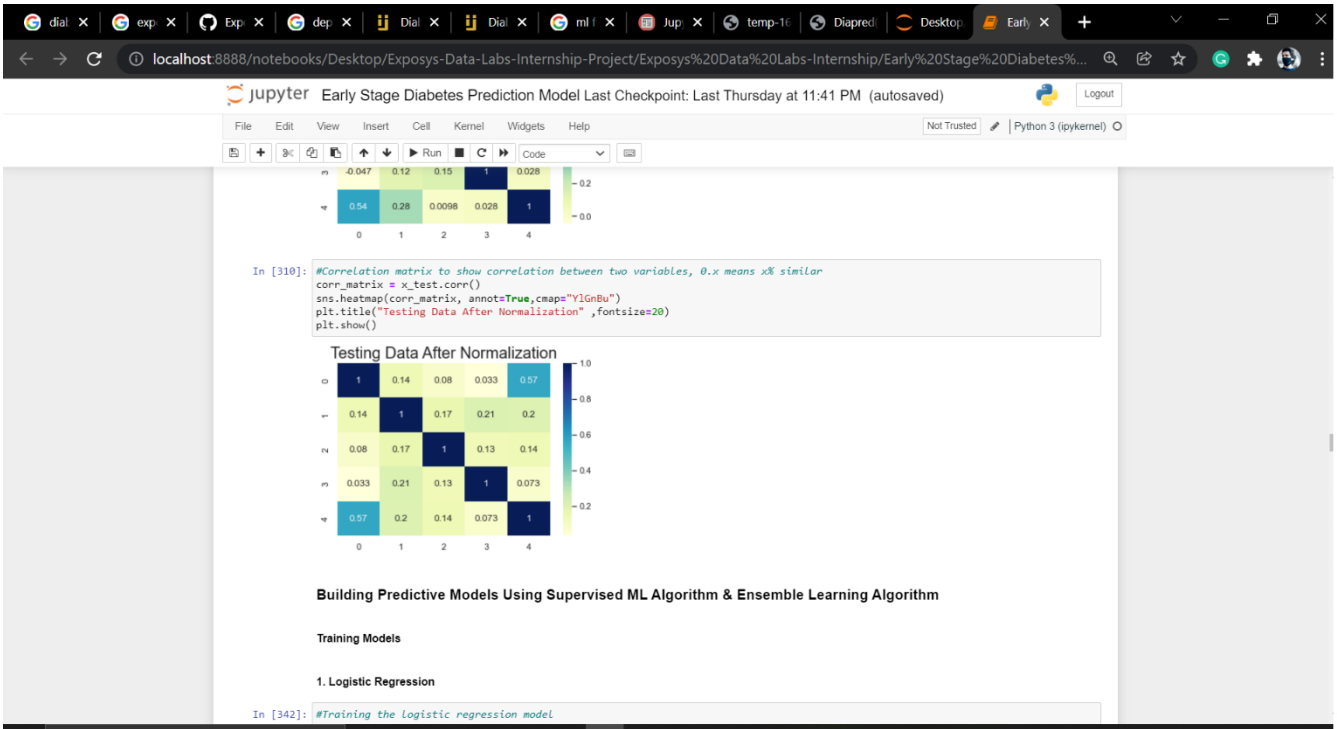
In [348]:
```
# NOW WITH K=30
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=24, metric = 'minkowski')
```

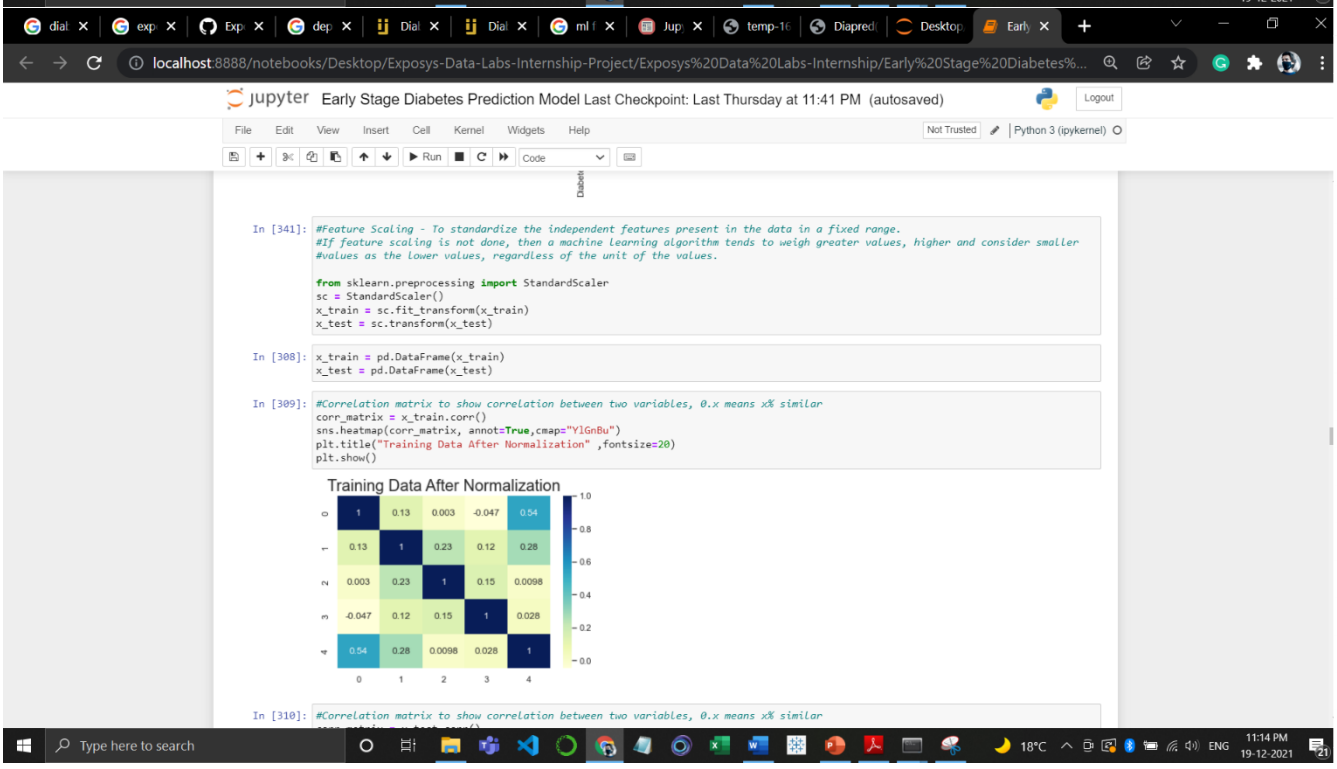
Type here to search

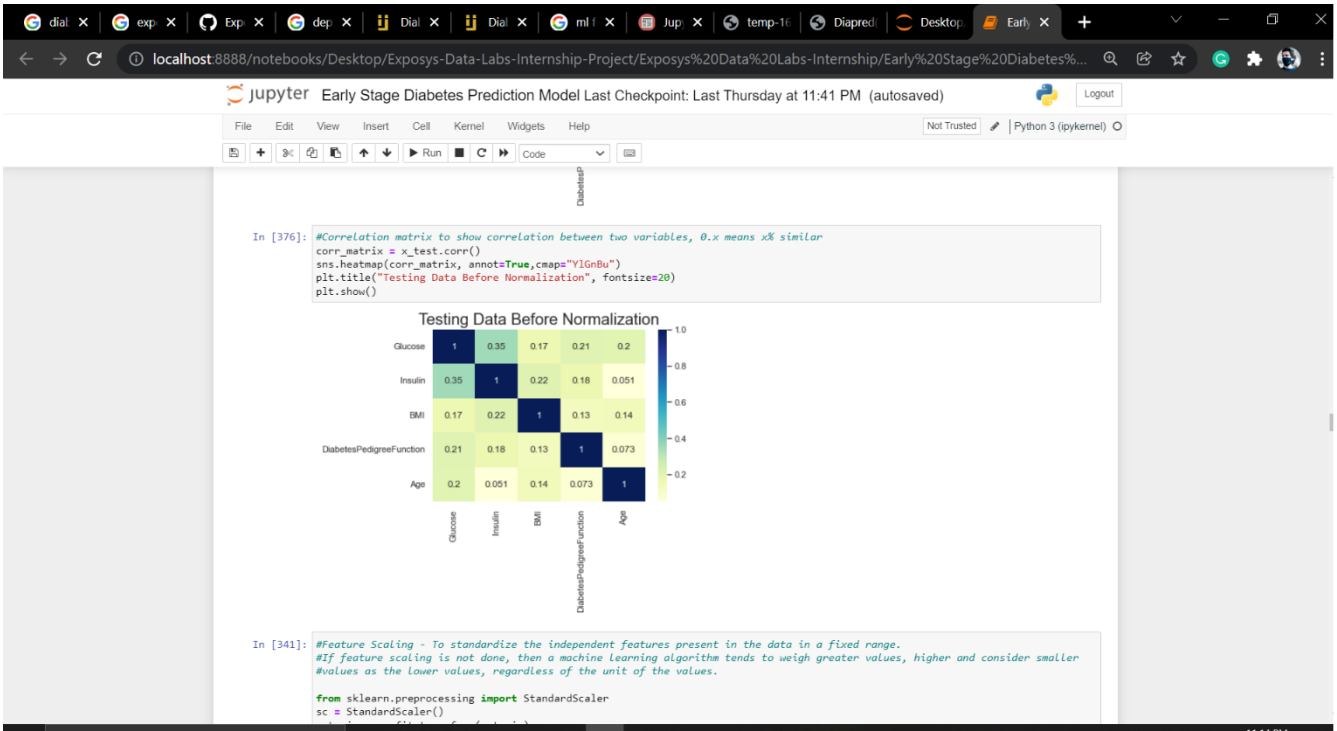
18°C ENG 11:14 PM 19-12-2021

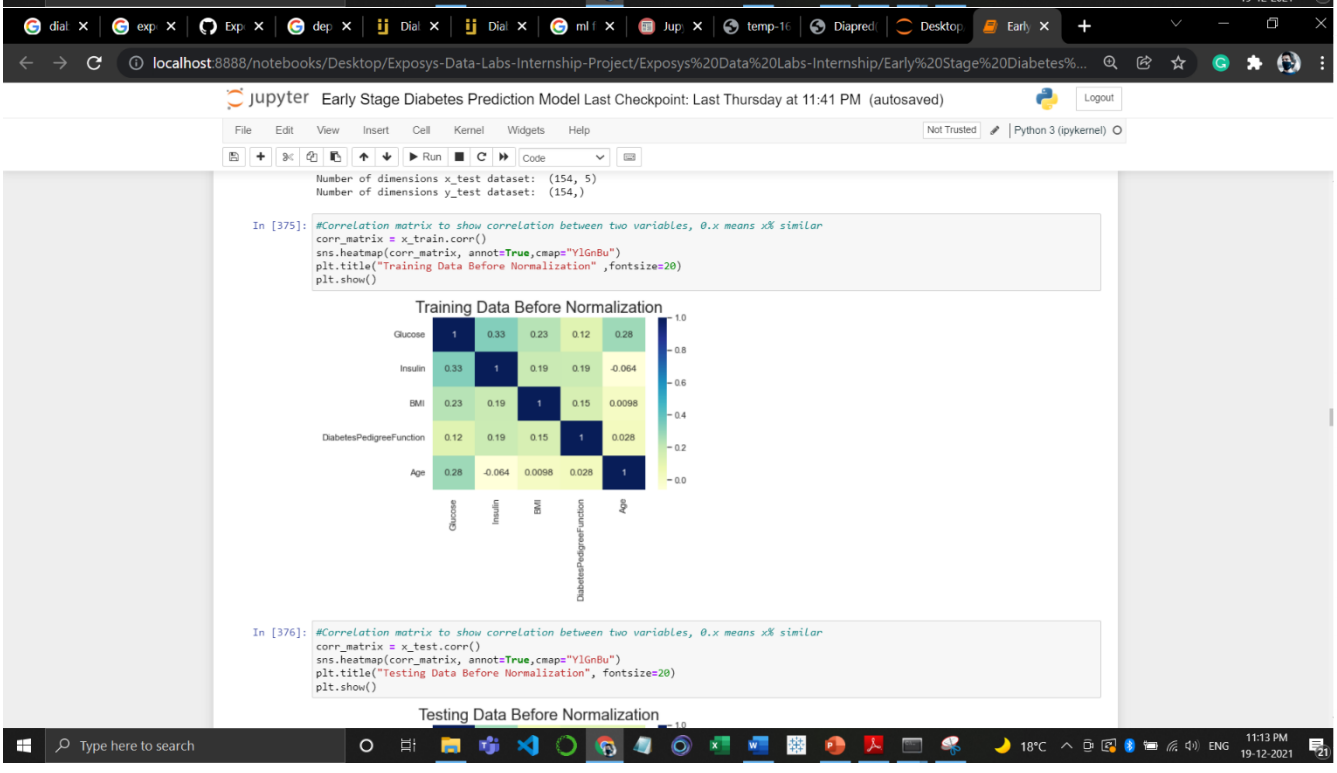

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Out[343]: GaussianNB()
In [344]:
#Training the Naive Bayes Model
from sklearn.naive_bayes import GaussianNB , BernoulliNB ,MultinomialNB # GaussianNB : For Continuous distribution of data , Ber
Bernoulli_NB = BernoulliNB()
Bernoulli_NB.fit(x_train,y_train)
Out[344]: BernoulliNB()
3. K Nearset Neighbor
In [345]:
# Training the K - Nearest Neighbour Model
from sklearn.neighbors import KNeighborsClassifier
import warnings
warnings.filterwarnings('ignore')
# Choosing correct value of K
error_rate = []
# Will take some time
for i in range(1,40):
knn = KNeighborsClassifier(n_neighbors=i)
knn.fit(x_train,y_train)
pred_i = knn.predict(x_test)
# pred_i=pred_i.reshape(231,1)
error_rate.append(np.mean(pred_i != y_test))
In [346]:
plt.figure(figsize=(10,6))
Type here to search
18°C
ENG
11:14 PM
19-12-2021

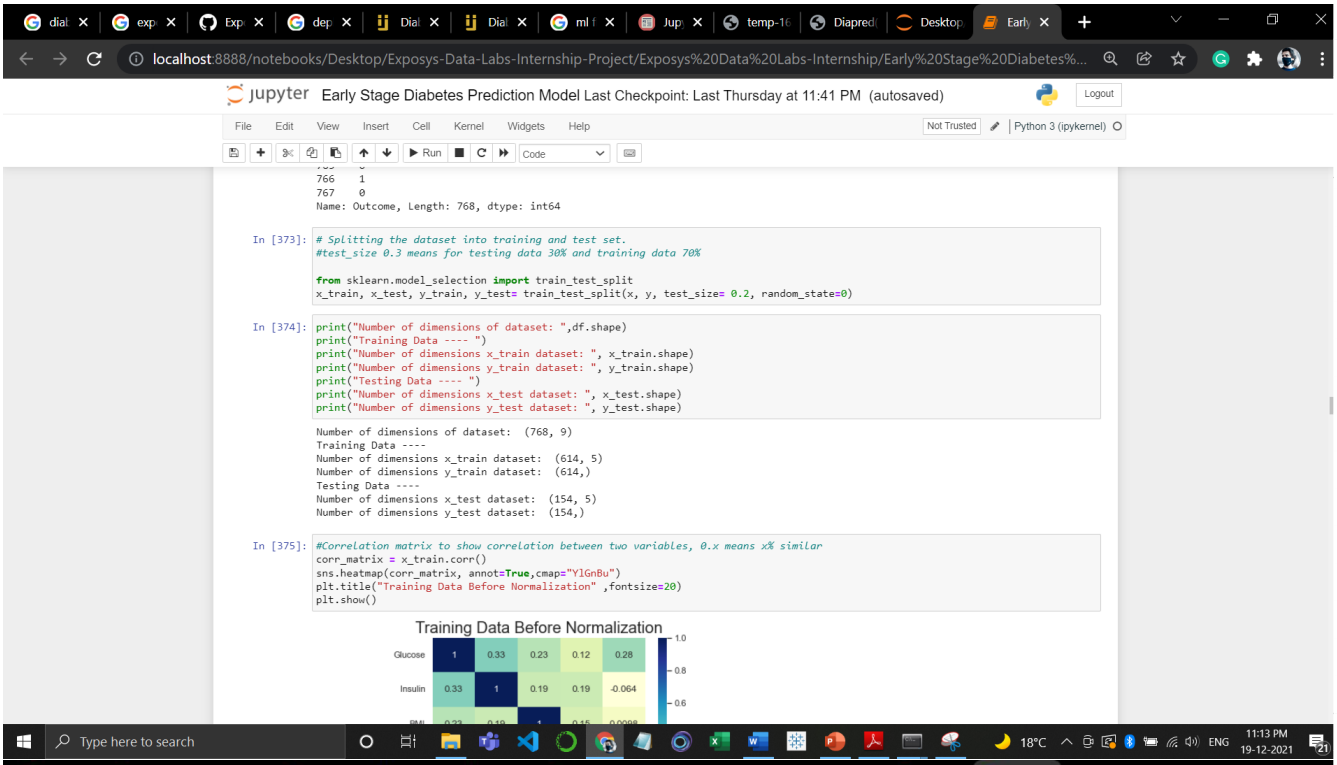

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Building Predictive Models Using Supervised ML Algorithm & Ensemble Learning Algorithm
Training Models
1. Logistic Regression
In [342]:
#Training the Logistic regression model
from sklearn.linear_model import LogisticRegression
Lr_classifier = LogisticRegression(random_state=0)
Lr_classifier.fit(x_train, y_train)
Out[342]: LogisticRegression(random_state=0)
2. Naive Bayes
In [343]:
#Training the Naive Bayes Model
from sklearn.naive_bayes import GaussianNB , BernoulliNB ,MultinomialNB # GaussianNB : For Continuous distribution of data , Ber
Gaussian_NB = GaussianNB()
Gaussian_NB.fit(x_train,y_train)
Out[343]: GaussianNB()
In [344]:
#Training the Naive Bayes Model
from sklearn.naive_bayes import GaussianNB , BernoulliNB ,MultinomialNB # GaussianNB : For Continuous distribution of data , Ber
Bernoulli_NB = BernoulliNB()
Bernoulli_NB.fit(x_train,y_train)
Out[344]: BernoulliNB()
Type here to search
18°C
ENG
11:14 PM
19-12-2021

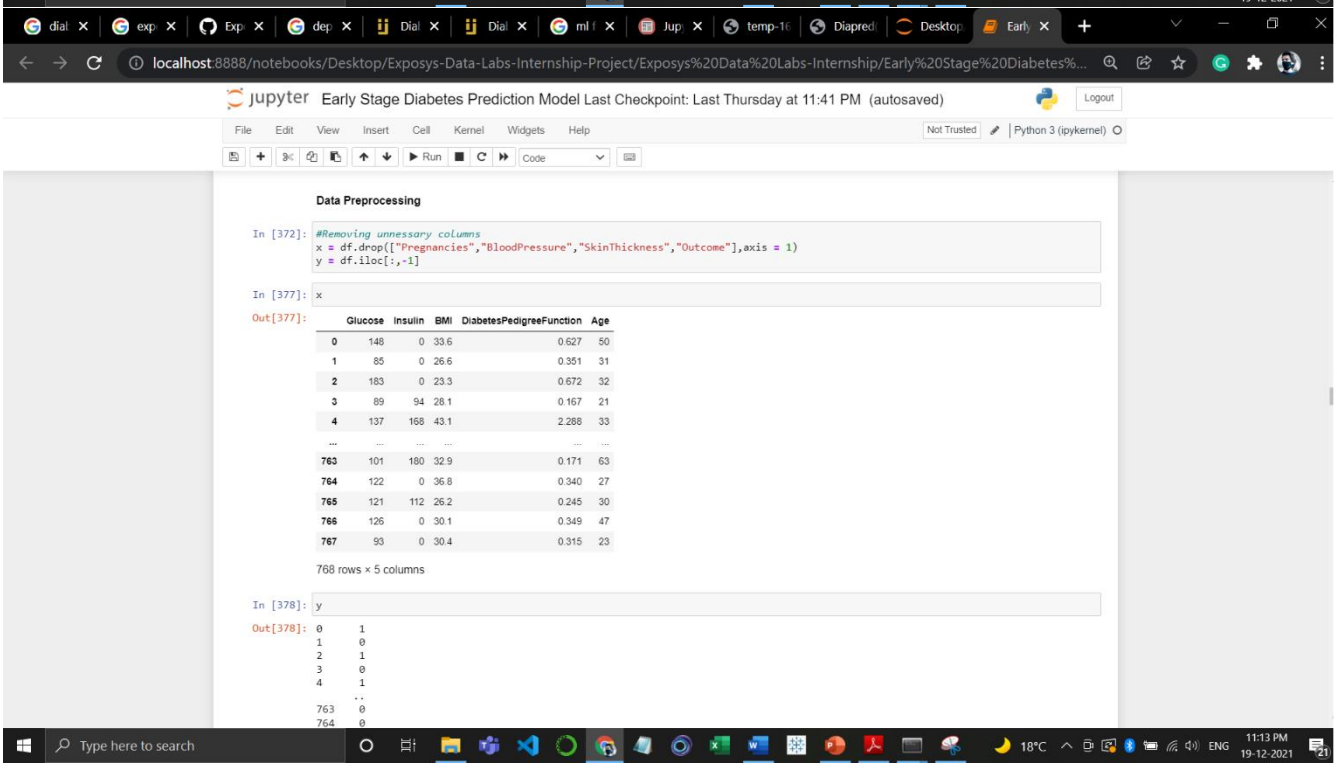

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0.54 0.28 0.0098 0.028 1
0 1 2 3 4
0.2
0.0
In [310]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
corr_matrix = x_test.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.title("Testing Data After Normalization" ,fontsize=20)
plt.show()
Testing Data After Normalization
1 0.14 0.08 0.033 0.57
0.14 1 0.17 0.21 0.2
0.08 0.17 1 0.13 0.14
0.033 0.21 0.13 1 0.073
0.57 0.2 0.14 0.073 1
0 1 2 3 4
1.0
0.8
0.6
0.4
0.2
Building Predictive Models Using Supervised ML Algorithm & Ensemble Learning Algorithm
Training Models
1. Logistic Regression
In [342]:
#Training the logistic regression model
Type here to search
18°C
ENG
11:14 PM
19-12-2021

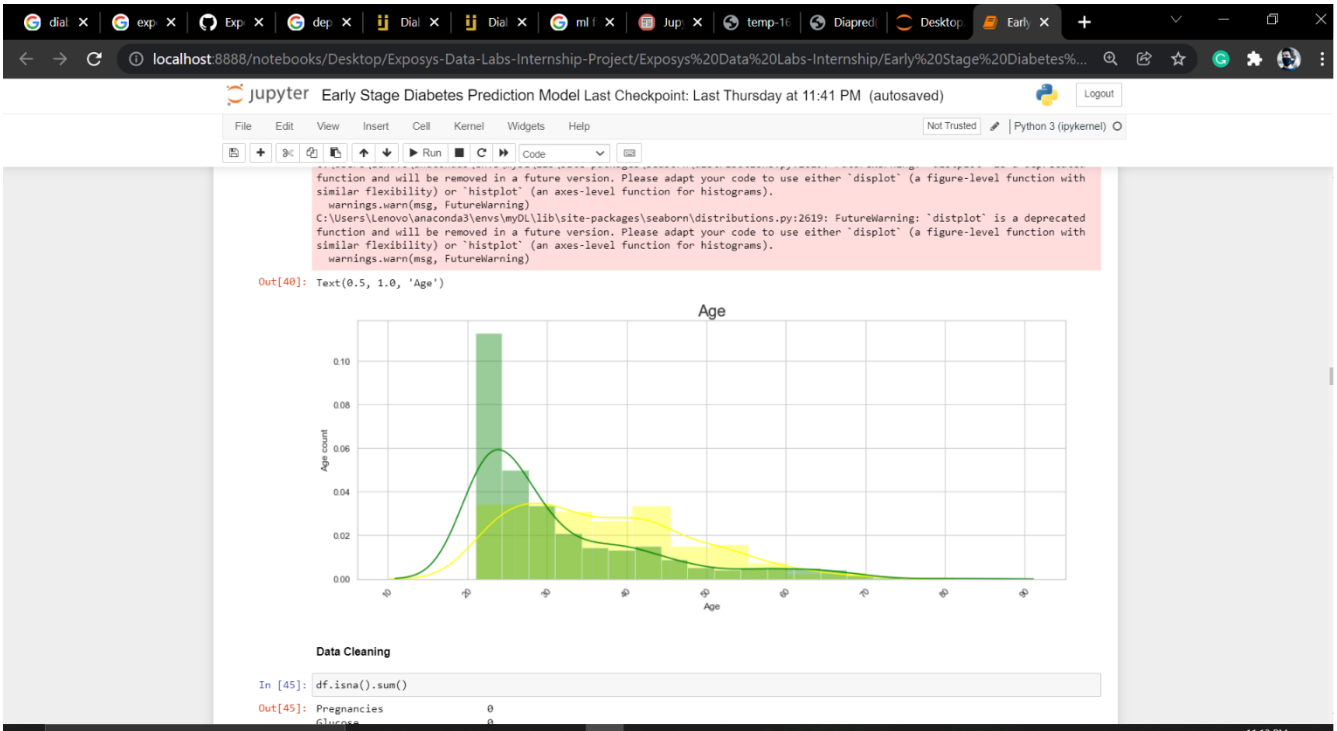

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [341]:
#Feature Scaling - To standardize the independent features present in the data in a fixed range.
#If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller
#values as the lower values, regardless of the unit of the values.
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
In [308]:
x_train = pd.DataFrame(x_train)
x_test = pd.DataFrame(x_test)
In [309]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
corr_matrix = x_train.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.title("Training Data After Normalization" ,fontsize=20)
plt.show()
Training Data After Normalization
1 0.13 0.003 -0.047 0.54
0.13 1 0.23 0.12 0.28
0.003 0.23 1 0.15 0.0098
-0.047 0.12 0.15 1 0.028
0.54 0.28 0.0098 0.028 1
0 1 2 3 4
1.0
0.8
0.6
0.4
0.2
0.0
In [310]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
Type here to search
18°C
ENG
11:14 PM
19-12-2021

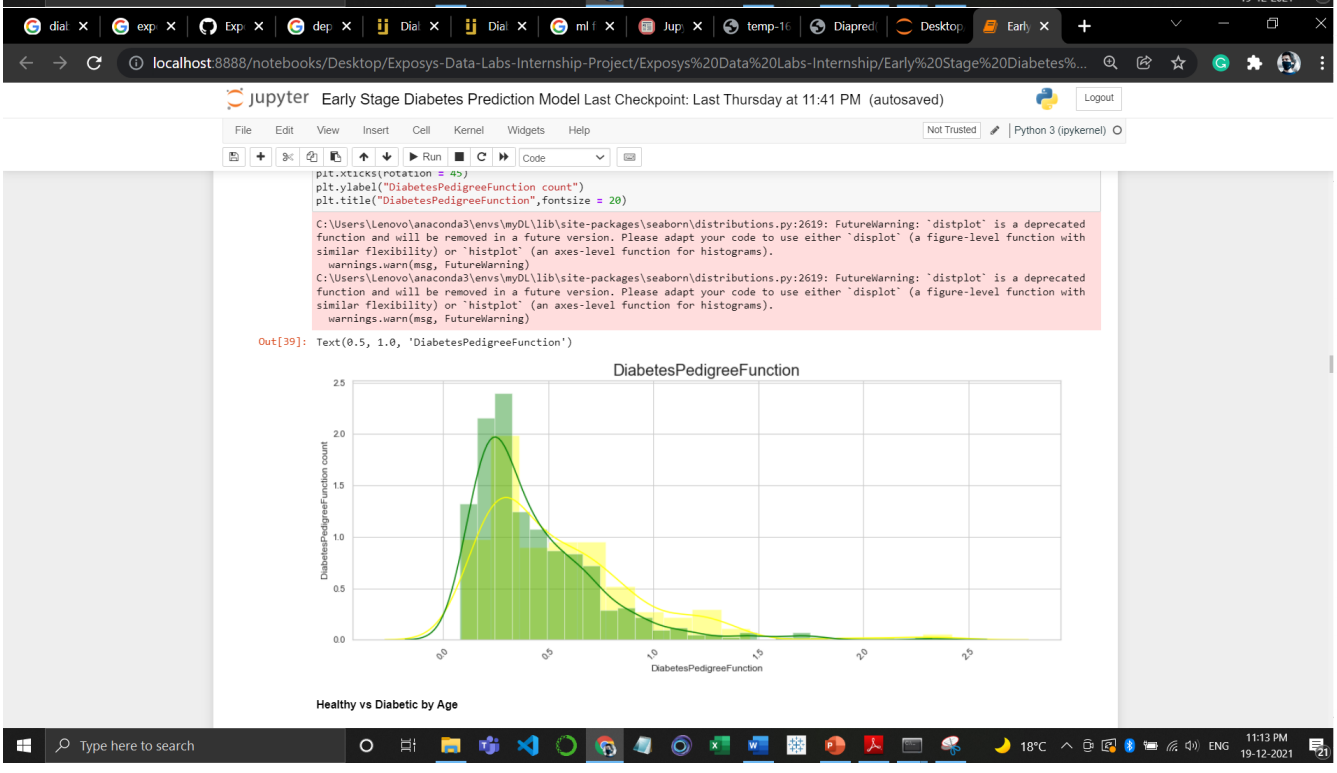

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
DiabetesP
In [376]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
corr_matrix = x_test.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.title("Testing Data Before Normalization", fontsize=20)
plt.show()
Testing Data Before Normalization
Glucose 1 0.35 0.17 0.21 0.2
Insulin 0.35 1 0.22 0.18 0.051
BMI 0.17 0.22 1 0.13 0.14
DiabetesPedigreeFunction 0.21 0.18 0.13 1 0.073
Age 0.2 0.051 0.14 0.073 1
Glucose Insulin BMI DiabetesPedigreeFunction Age
1.0 0.8 0.6 0.4 0.2
In [341]:
#Feature Scaling - To standardize the independent features present in the data in a fixed range.
#If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller
#values as the lower values, regardless of the unit of the values.
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
Type here to search
18°C
ENG
11:14 PM
19-12-2021

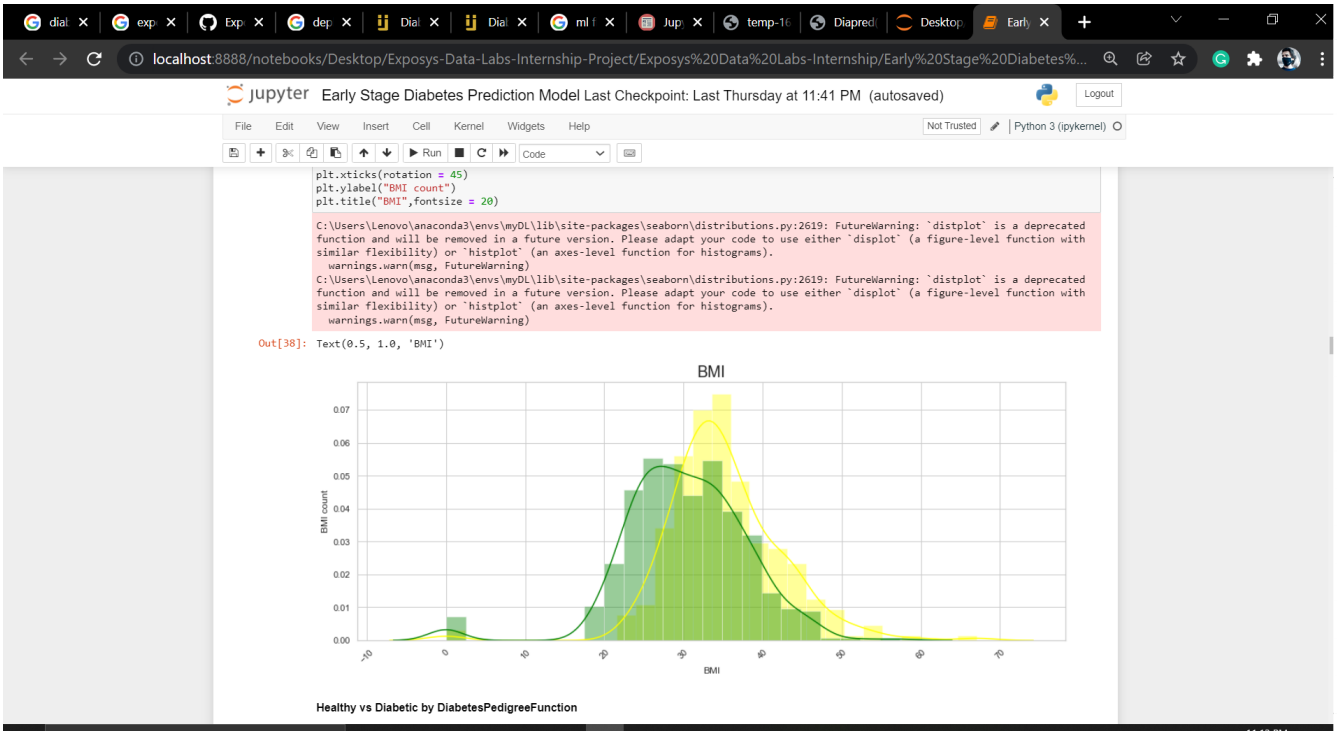

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Number of dimensions x_test dataset: (154, 5)
Number of dimensions y_test dataset: (154,)
In [375]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
corr_matrix = x_train.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.title("Training Data Before Normalization" ,fontsize=20)
plt.show()
Training Data Before Normalization
Glucose 1 0.33 0.23 0.12 0.28
Insulin 0.33 1 0.19 0.19 -0.064
BMI 0.23 0.19 1 0.15 0.0098
DiabetesPedigreeFunction 0.12 0.19 0.15 1 0.028
Age 0.28 -0.064 0.0098 0.028 1
Glucose Insulin BMI DiabetesPedigreeFunction Age
1.0 0.8 0.6 0.4 0.2 0.0
In [376]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
corr_matrix = x_test.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.title("Testing Data Before Normalization", fontsize=20)
plt.show()
Testing Data Before Normalization
Type here to search
18°C
ENG
11:13 PM
19-12-2021

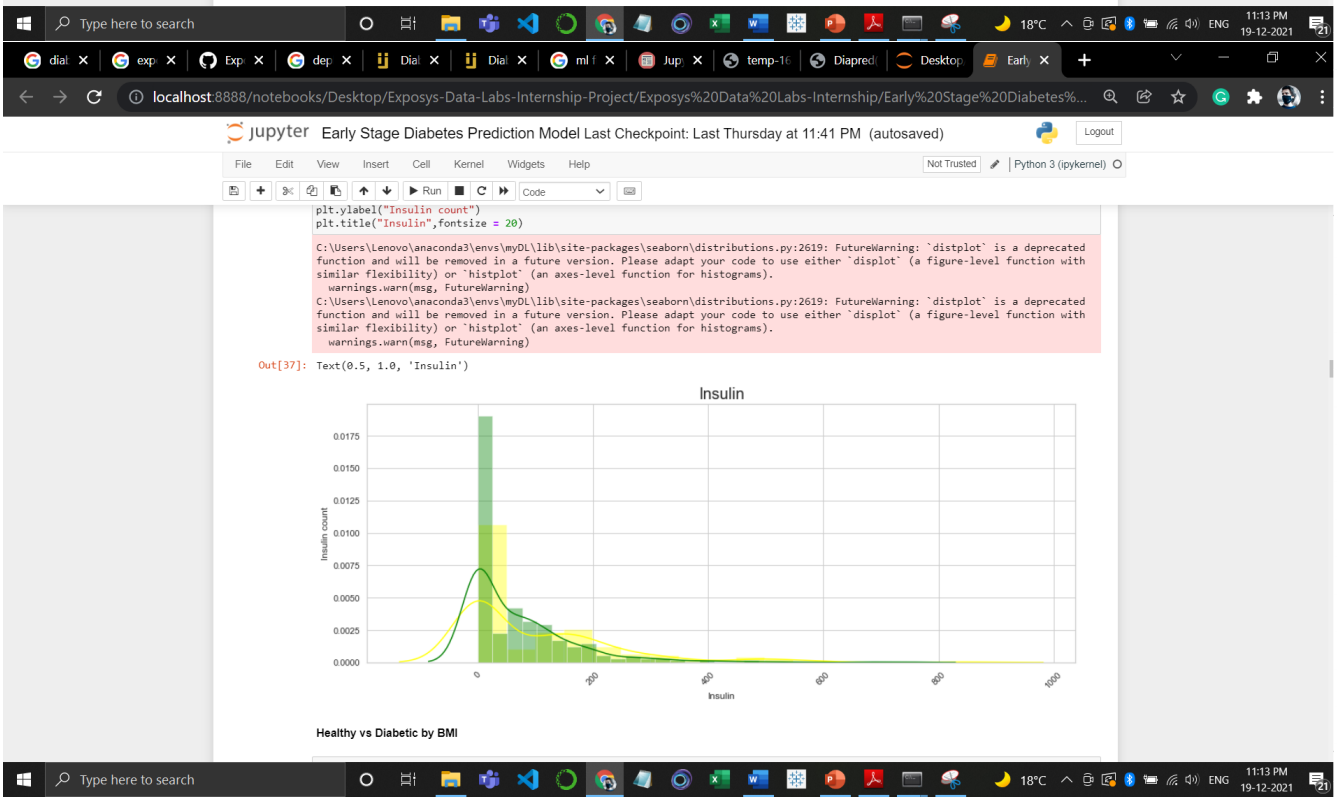

jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
766 1
767 0
Name: Outcome, Length: 768, dtype: int64
In [373]:
Splitting the dataset into training and test set.
#test_size 0.3 means for testing data 30% and training data 70%
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
In [374]:
print("Number of dimensions of dataset: ",df.shape)
print("Training Data ---- ")
print("Number of dimensions x_train dataset: ", x_train.shape)
print("Number of dimensions y_train dataset: ", y_train.shape)
print("Testing Data ---- ")
print("Number of dimensions x_test dataset: ", x_test.shape)
print("Number of dimensions y_test dataset: ", y_test.shape)
Number of dimensions of dataset: (768, 9)
Training Data ----
Number of dimensions x_train dataset: (614, 5)
Number of dimensions y_train dataset: (614,)
Testing Data ----
Number of dimensions x_test dataset: (154, 5)
Number of dimensions y_test dataset: (154,)
In [375]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
corr_matrix = x_train.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.title("Training Data Before Normalization" ,fontsize=20)
plt.show()
Training Data Before Normalization
Glucose 1 0.33 0.23 0.12 0.28
Insulin 0.33 1 0.19 0.19 -0.064
1.0
0.8
0.6
Type here to search
18°C
ENG
11:13 PM
19-12-2021

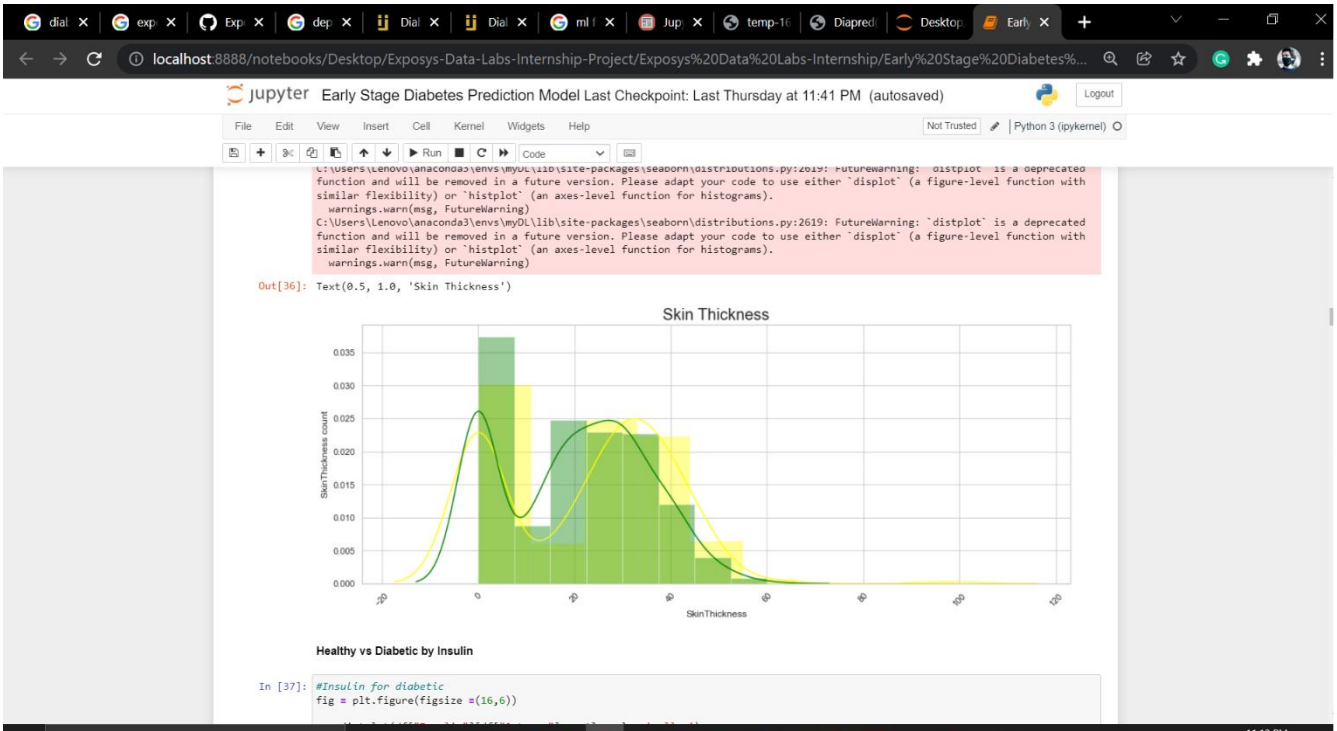

jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Data Preprocessing
In [372]:
#Removing unnessary columns
x = df.drop(["Pregnancies","BloodPressure","SkinThickness","Outcome"],axis = 1)
y = df.iloc[:,-1]
In [377]:
x
Out[377]:
Glucose Insulin BMI DiabetesPedigreeFunction Age
0 148 0 33.6 0.627 50
1 85 0 26.6 0.351 31
2 183 0 23.3 0.672 32
3 89 94 28.1 0.167 21
4 137 168 43.1 2.288 33
763 101 180 32.9 0.171 63
764 122 0 36.8 0.340 27
765 121 112 26.2 0.245 30
766 126 0 30.1 0.349 47
767 93 0 30.4 0.315 23
768 rows × 5 columns
In [378]:
y
Out[378]:
0 1
1 0
2 1
3 0
4 1
763 0
764 0
Type here to search
18°C
ENG
11:13 PM
19-12-2021

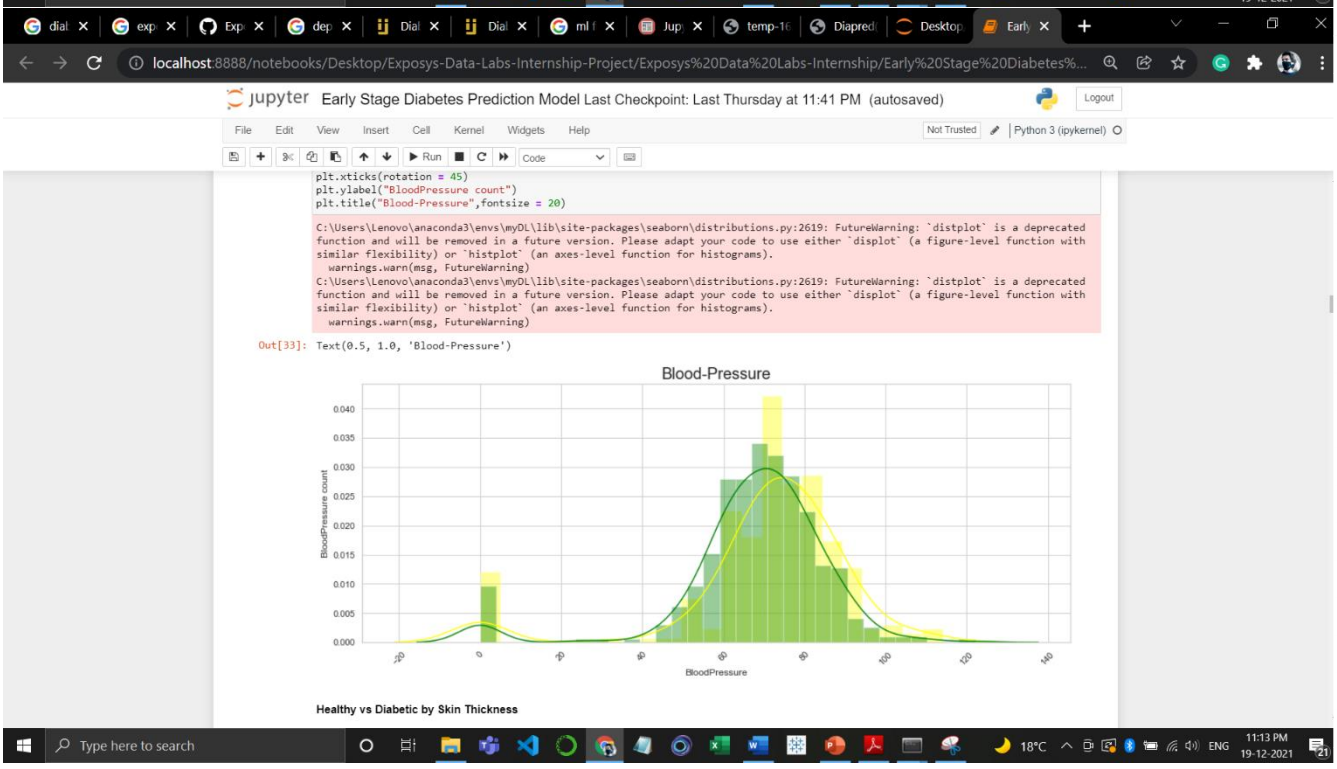

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[40]: Text(0.5, 1.0, 'Age')
Age
Age count
Age
Data Cleaning
In [45]: df.isna().sum()
Out[45]: Pregnancies 0
Glucose 0
Type here to search
18°C
ENG
11:13 PM
19-12-2021

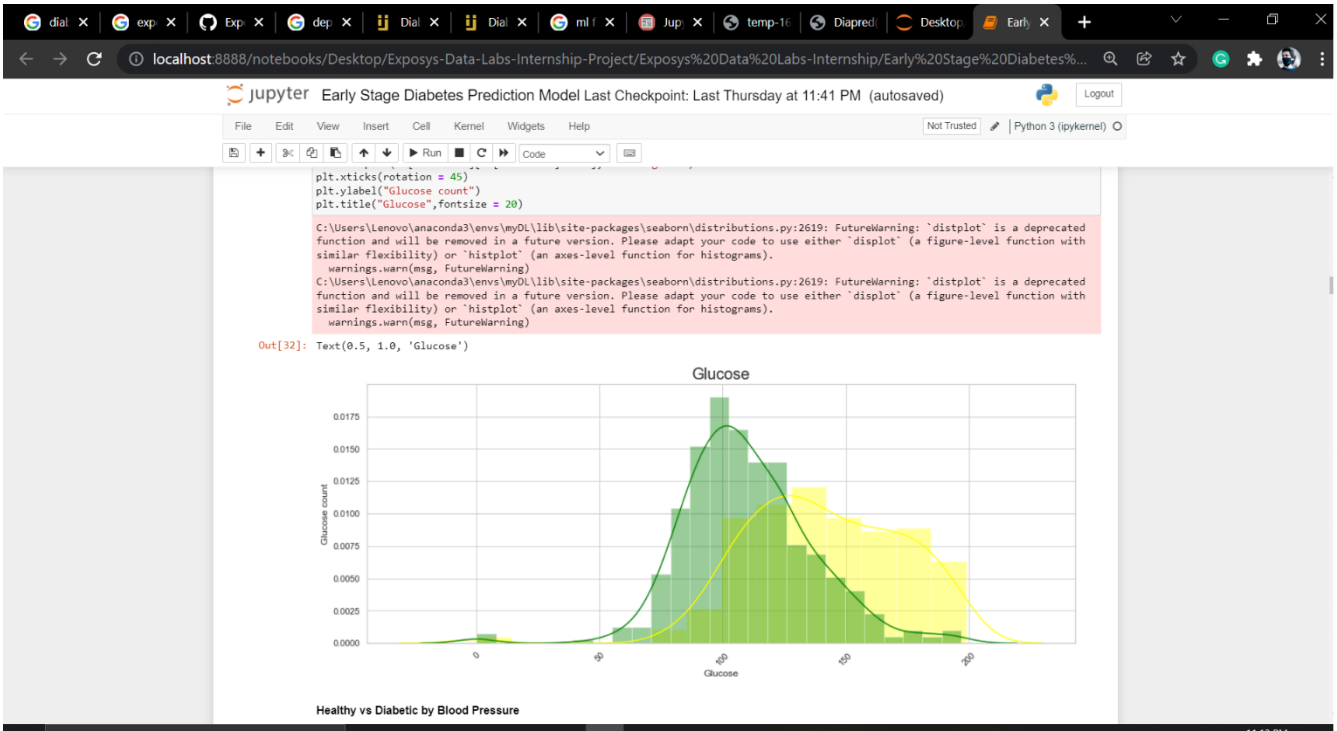

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.xticks(rotation = 45)
plt.ylabel("DiabetesPedigreeFunction count")
plt.title("DiabetesPedigreeFunction",fontsize = 20)
function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[39]: Text(0.5, 1.0, 'DiabetesPedigreeFunction')
DiabetesPedigreeFunction
DiabetesPedigreeFunction count
DiabetesPedigreeFunction
Healthy vs Diabetic by Age
Type here to search
18°C
ENG
11:13 PM
19-12-2021

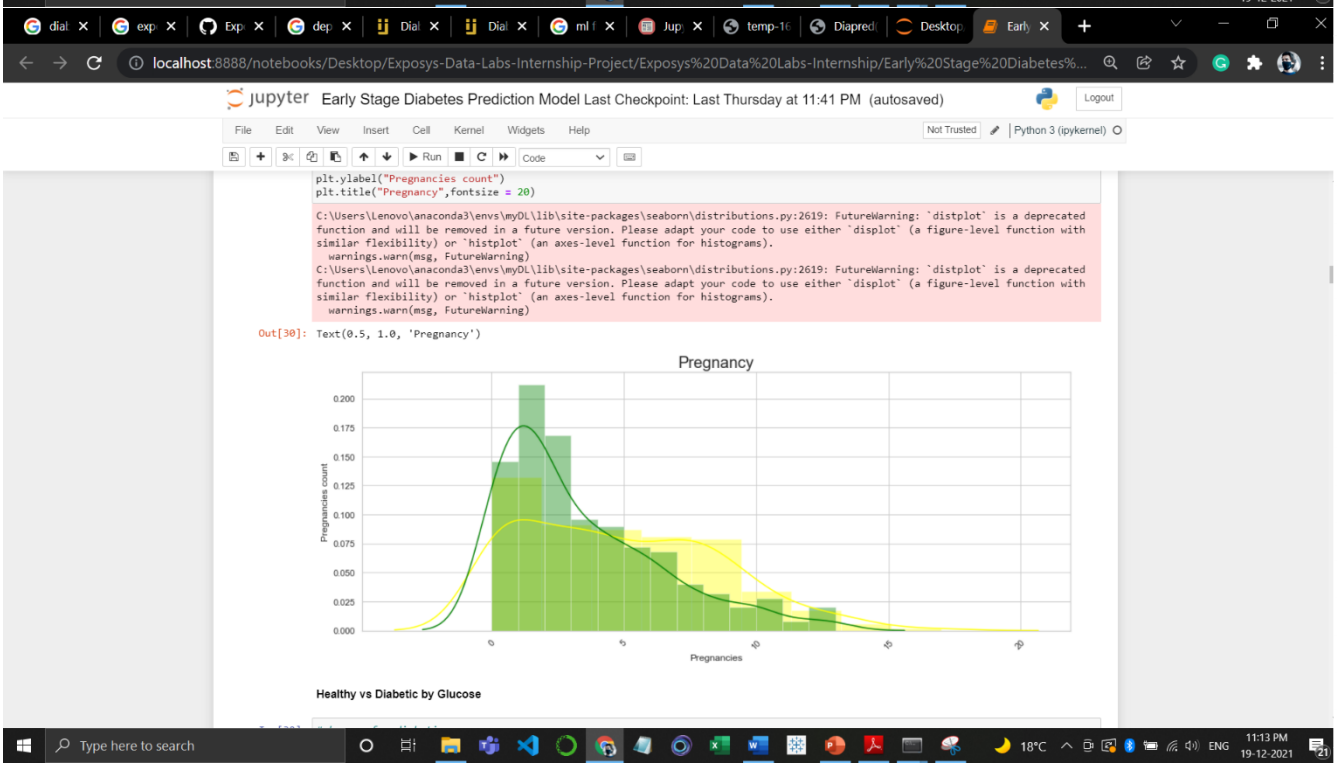

jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.xticks(rotation = 45)
plt.ylabel("BMI count")
plt.title("BMI",fontsize = 20)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[38]: Text(0.5, 1.0, 'BMI')
BMI
BMI count
BMI
Healthy vs Diabetic by DiabetesPedigreeFunction

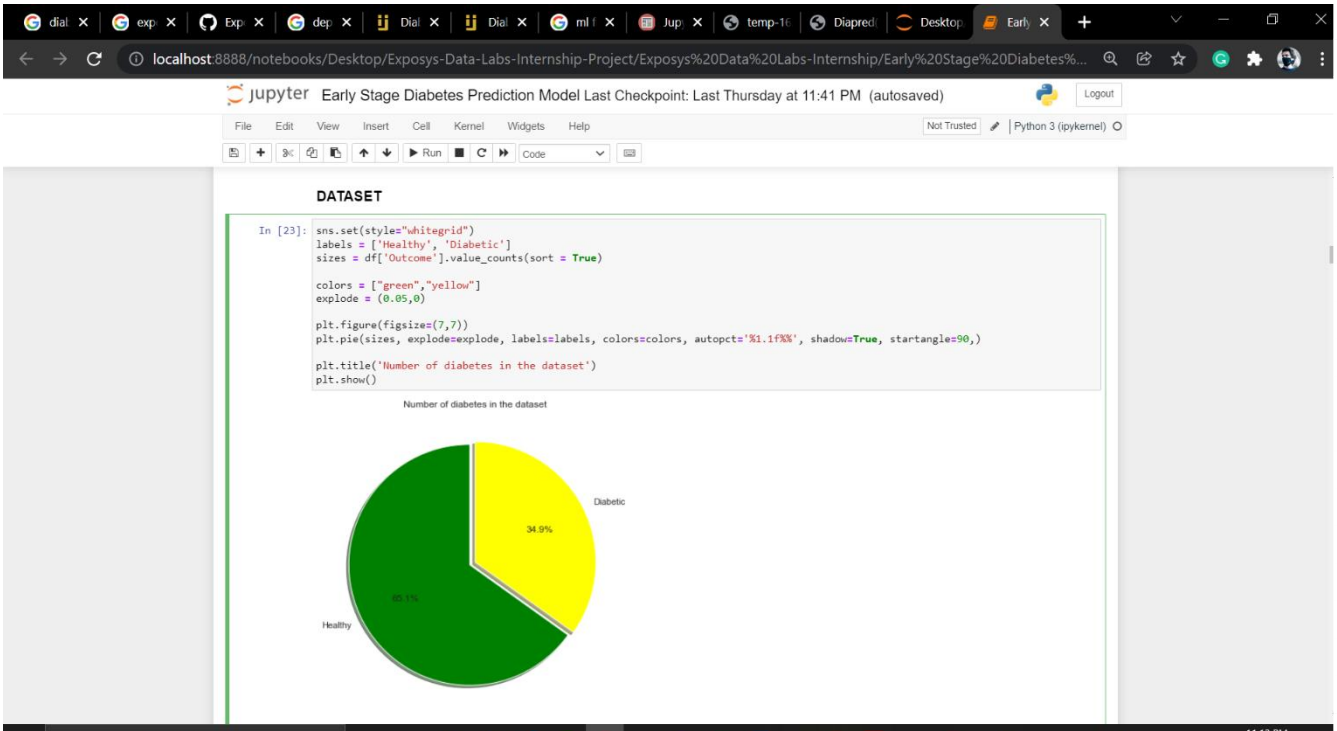

jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.ylabel("Insulin count")
plt.title("Insulin",fontsize = 20)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[37]: Text(0.5, 1.0, 'Insulin')
Insulin
Insulin count
Insulin
Healthy vs Diabetic by BMI

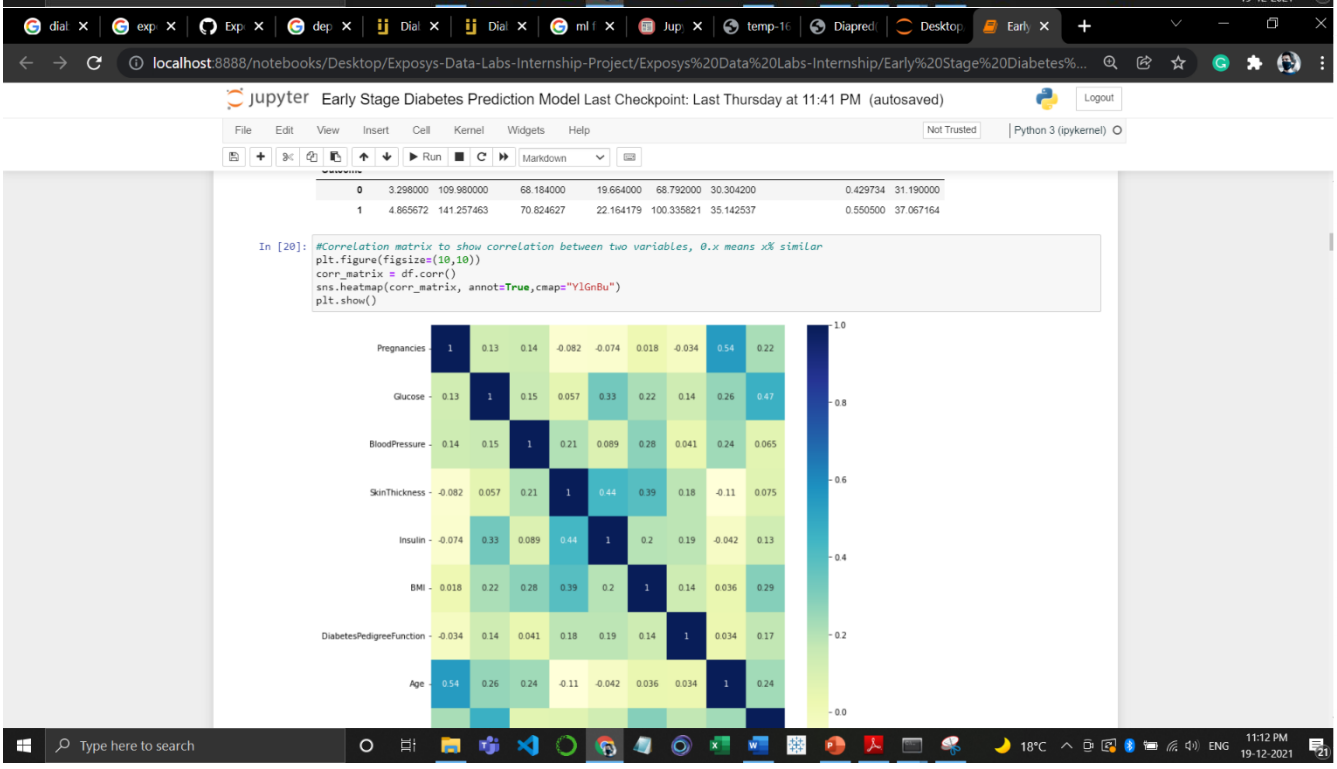

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[36]: Text(0.5, 1.0, 'Skin Thickness')
Skin Thickness
SkinThickness count
SkinThickness
Healthy vs Diabetic by Insulin
In [37]: #Insulin for diabetic
fig = plt.figure(figsize =(16,6))
Type here to search
18°C
ENG
11:13 PM
19-12-2021

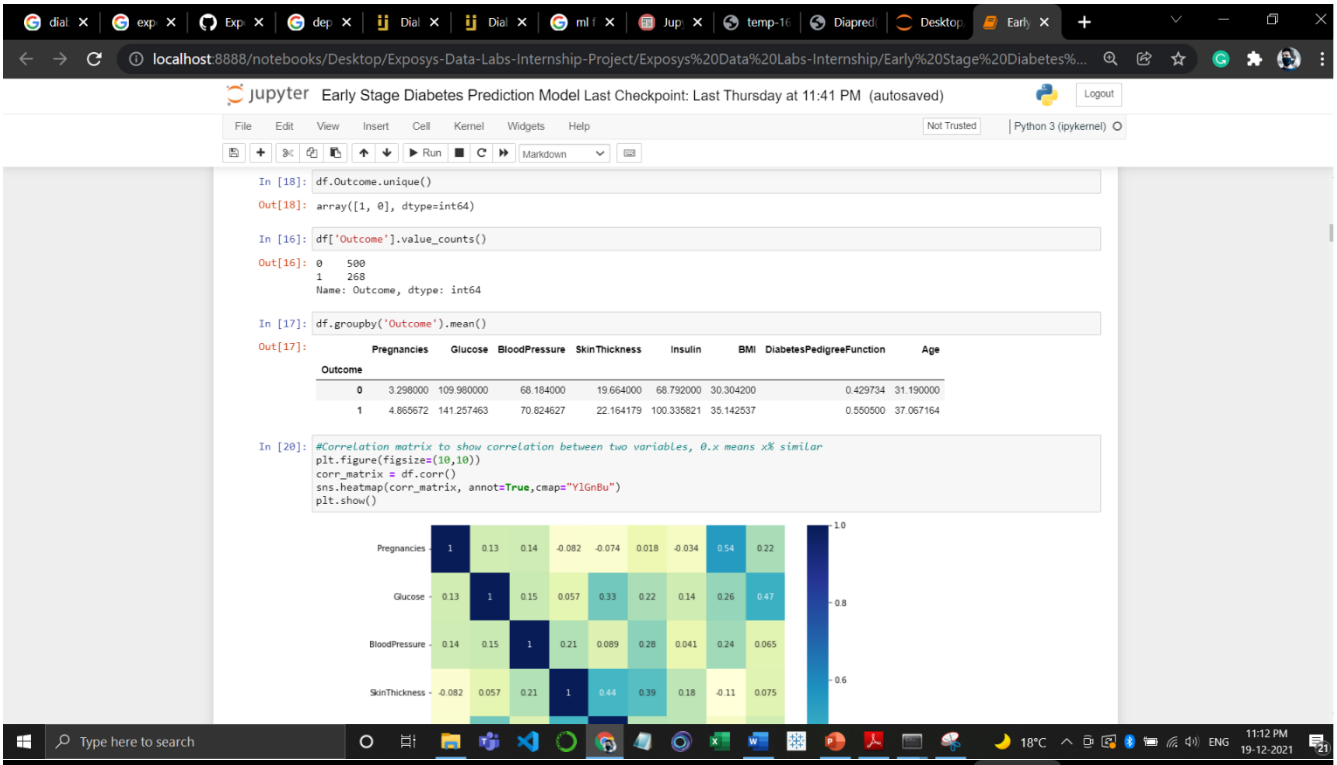

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.xticks(rotation = 45)
plt.ylabel("BloodPressure count")
plt.title("Blood-Pressure",fontsize = 20)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[33]: Text(0.5, 1.0, 'Blood-Pressure')
Blood-Pressure
BloodPressure count
BloodPressure
Healthy vs Diabetic by Skin Thickness
Type here to search
18°C
ENG
11:13 PM
19-12-2021

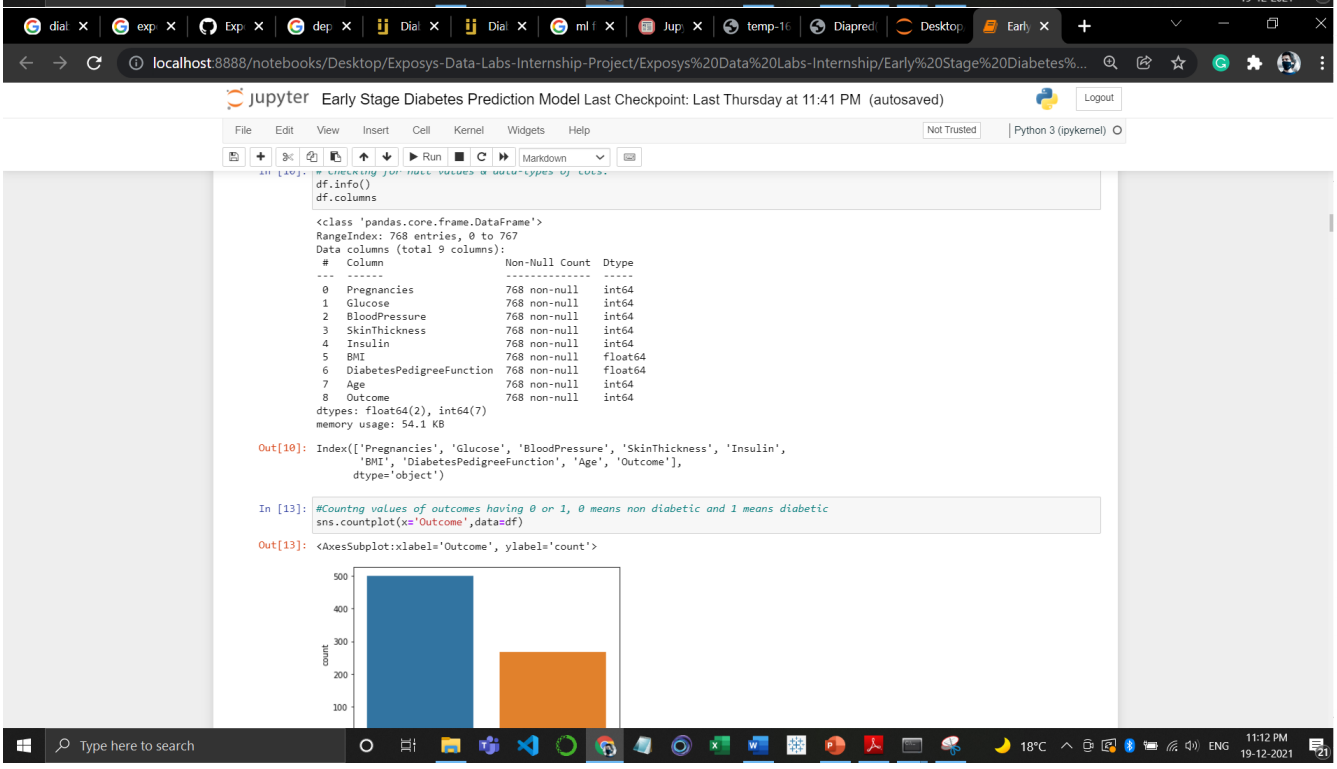

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.xticks(rotation = 45)
plt.ylabel("Glucose count")
plt.title("Glucose",fontsize = 20)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[32]: Text(0.5, 1.0, 'Glucose')
Glucose
Glucose count
Glucose
Healthy vs Diabetic by Blood Pressure
Type here to search
18°C
ENG
11:13 PM
19-12-2021

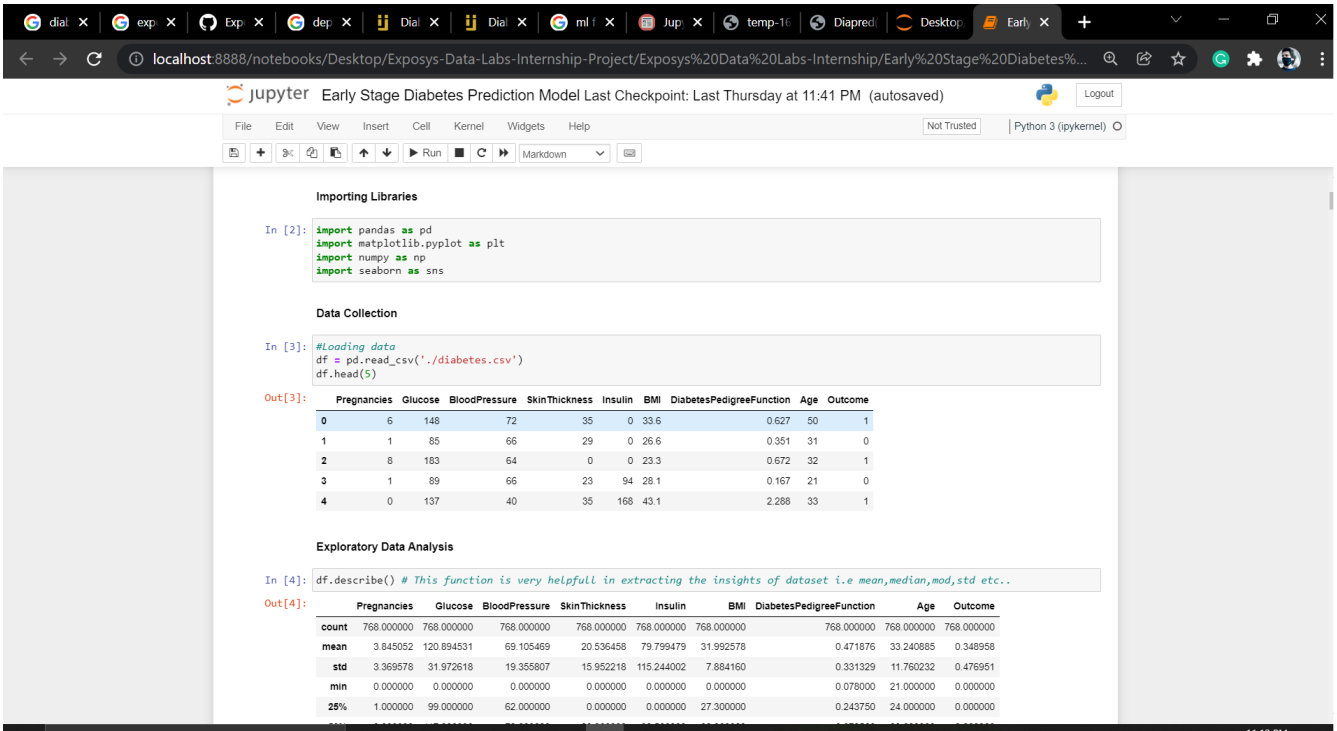

localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.ylabel("Pregnancies count")
plt.title("Pregnancy",fontsize = 20)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
C:\Users\Lenovo\anaconda3\envs\myDL\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
Out[30]: Text(0.5, 1.0, 'Pregnancy')
Pregnancy
Pregnancies count
Pregnancies
Healthy vs Diabetic by Glucose
Type here to search
18°C
ENG
11:13 PM
19-12-2021

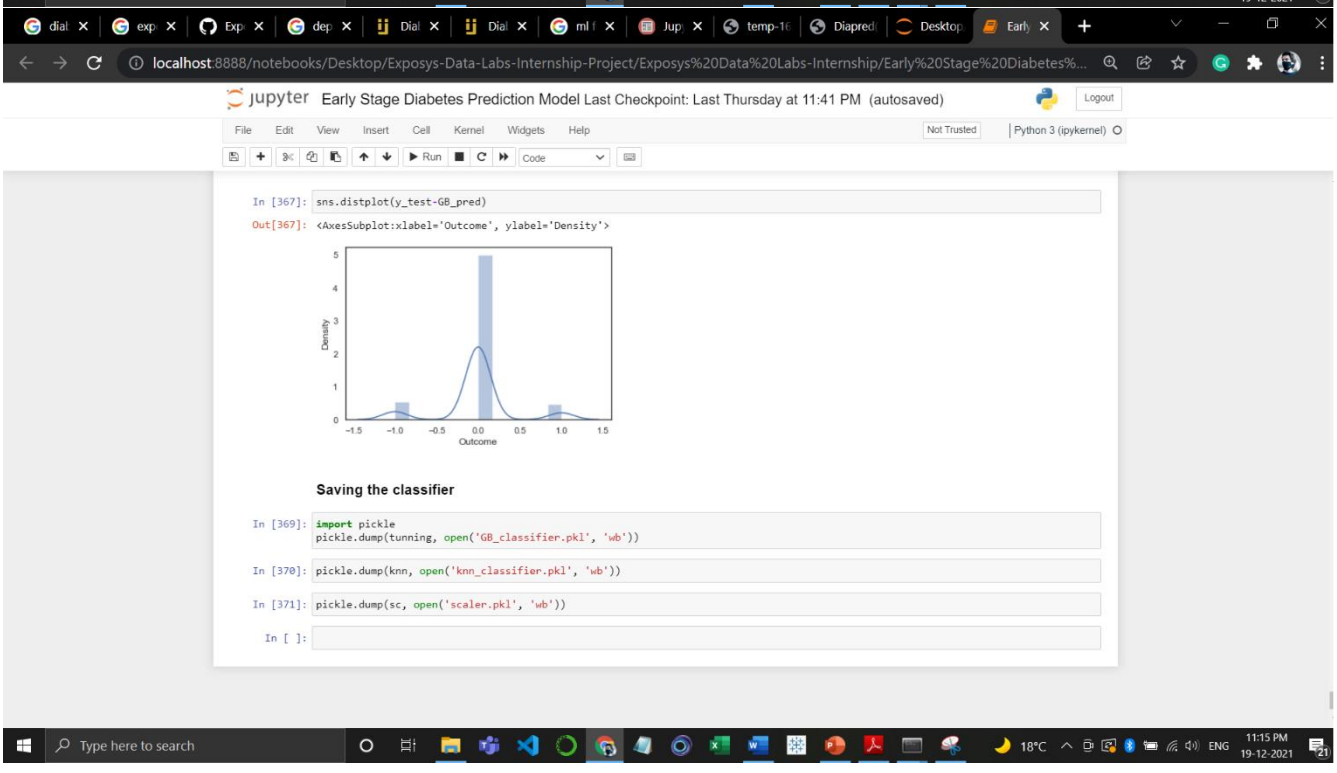

jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
DATASET
In [23]:
sns.set(style="whitegrid")
labels = ['Healthy', 'Diabetic']
sizes = df['Outcome'].value_counts(sort = True)
colors = ["green","yellow"]
explode = (0.05,0)
plt.figure(figsize=(7,7))
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=90,)
plt.title('Number of diabetes in the dataset')
plt.show()
Number of diabetes in the dataset
Diabetic
34.9%
65.1%
Healthy


jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
0 3.298000 109.980000 68.184000 19.664000 68.792000 30.304200 0.429734 31.190000
1 4.865672 141.257463 70.824627 22.164179 100.335821 35.142537 0.550500 37.067164
In [20]:
#Correlation matrix to show correlation between two variables, 0.x means x% similar
plt.figure(figsize=(10,10))
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.show()
Pregnancies 1 0.13 0.14 -0.082 -0.074 0.018 -0.034 0.54 0.22
Glucose 0.13 1 0.15 0.057 0.33 0.22 0.14 0.26 0.47
BloodPressure 0.14 0.15 1 0.21 0.089 0.28 0.041 0.24 0.065
SkinThickness -0.082 0.057 0.21 1 0.44 0.39 0.18 -0.11 0.075
Insulin -0.074 0.33 0.089 0.44 1 0.2 0.19 -0.042 0.13
BMI 0.018 0.22 0.28 0.39 0.2 1 0.14 0.036 0.29
DiabetesPedigreeFunction -0.034 0.14 0.041 0.18 0.19 0.14 1 0.034 0.17
Age 0.54 0.26 0.24 -0.11 -0.042 0.036 0.034 1 0.24
1.0
0.8
0.6
0.4
0.2
0.0


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
In [18]: df.Outcome.unique()
Out[18]: array([1, 0], dtype=int64)
In [16]: df['Outcome'].value_counts()
Out[16]: 0 500
1 268
Name: Outcome, dtype: int64
In [17]: df.groupby('Outcome').mean()
Out[17]:
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age
Outcome
0 3.298000 109.980000 68.184000 19.664000 68.792000 30.304200 0.429734 31.190000
1 4.865672 141.257463 70.824627 22.164179 100.335821 35.142537 0.550500 37.067164
In [20]: #Correlation matrix to show correlation between two variables, 0.x means x% similar
plt.figure(figsize=(10,10))
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True,cmap="YlGnBu")
plt.show()
Pregnancies 1 0.13 0.14 -0.082 -0.074 0.018 -0.034 0.54 0.22
Glucose 0.13 1 0.15 0.057 0.33 0.22 0.14 0.26 0.47
BloodPressure 0.14 0.15 1 0.21 0.089 0.28 0.041 0.24 0.065
SkinThickness -0.082 0.057 0.21 1 0.44 0.39 0.18 -0.11 0.075
1.0
0.8
0.6
Type here to search
18°C
ENG
11:12 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
df.info()
df.columns
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Column Non-Null Count Dtype
0 Pregnancies 768 non-null int64
1 Glucose 768 non-null int64
2 BloodPressure 768 non-null int64
3 SkinThickness 768 non-null int64
4 Insulin 768 non-null int64
5 BMI 768 non-null float64
6 DiabetesPedigreeFunction 768 non-null float64
7 Age 768 non-null int64
8 Outcome 768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
Out[10]: Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
dtype='object')
In [13]: #Countng values of outcomes having 0 or 1, 0 means non diabetic and 1 means diabetic
sns.countplot(x='Outcome',data=df)
Out[13]: <AxesSubplot:xlabel='Outcome', ylabel='count'>
count
500
400
300
200
100
Type here to search
18°C
ENG
11:12 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Markdown
Importing Libraries
In [2]:
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
Data Collection
In [3]:
#Loading data
df = pd.read_csv('./diabetes.csv')
df.head(5)
Out[3]:
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome
0 6 148 72 35 0 33.6 0.627 50 1
1 1 85 66 29 0 26.6 0.351 31 0
2 8 183 64 0 0 23.3 0.672 32 1
3 1 89 66 23 94 28.1 0.167 21 0
4 0 137 40 35 168 43.1 2.288 33 1
Exploratory Data Analysis
In [4]:
df.describe() # This function is very helpfull in extracting the insights of dataset i.e mean,median,mod,std etc..
Out[4]:
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome
count 768.000000 768.000000 768.000000 768.000000 768.000000 768.000000 768.000000 768.000000 768.000000
mean 3.845052 120.894531 69.105469 20.536458 79.799479 31.992578 0.471876 33.240885 0.348958
std 3.369578 31.972618 19.355807 15.952218 115.244002 7.884160 0.331329 11.760232 0.476951
min 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.078000 21.000000 0.000000
25% 1.000000 99.000000 62.000000 0.000000 0.000000 27.300000 0.243750 24.000000 0.000000
Type here to search
18°C
ENG
11:12 PM
19-12-2021


localhost:8888/notebooks/Desktop/Exposys-Data-Labs-Internship-Project/Exposys%20Data%20Labs-Internship/Early%20Stage%20Diabetes%...
jupyter Early Stage Diabetes Prediction Model Last Checkpoint: Last Thursday at 11:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [367]:
sns.distplot(y_test-GB_pred)
Out[367]:
<AxesSubplot:xlabel='Outcome', ylabel='Density'>
Density
Outcome
Saving the classifier
In [369]:
import pickle
pickle.dump(tunning, open('GB_classifier.pkl', 'wb'))
In [370]:
pickle.dump(knn, open('knn_classifier.pkl', 'wb'))
In [371]:
pickle.dump(sc, open('scaler.pkl', 'wb'))
In [ ]:
Type here to search
18°C
ENG
11:15 PM
19-12-2021

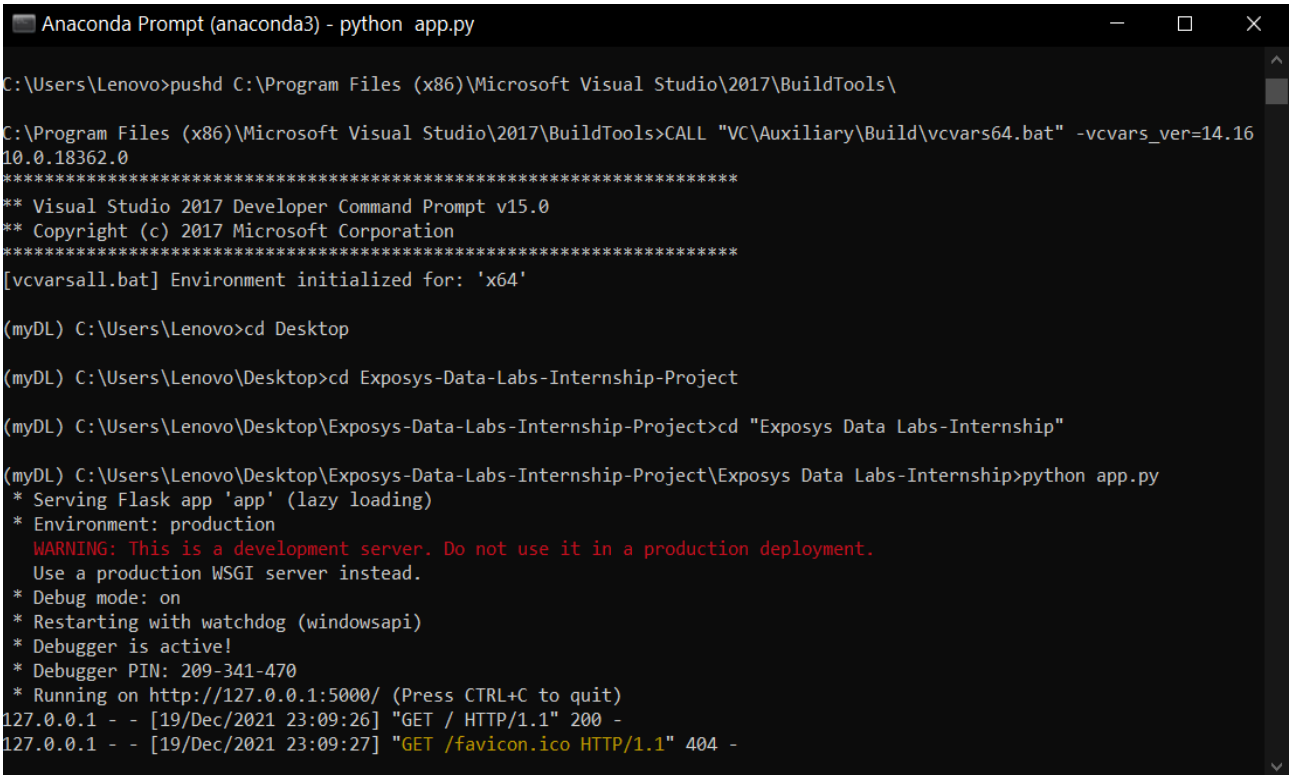
Chapter 5

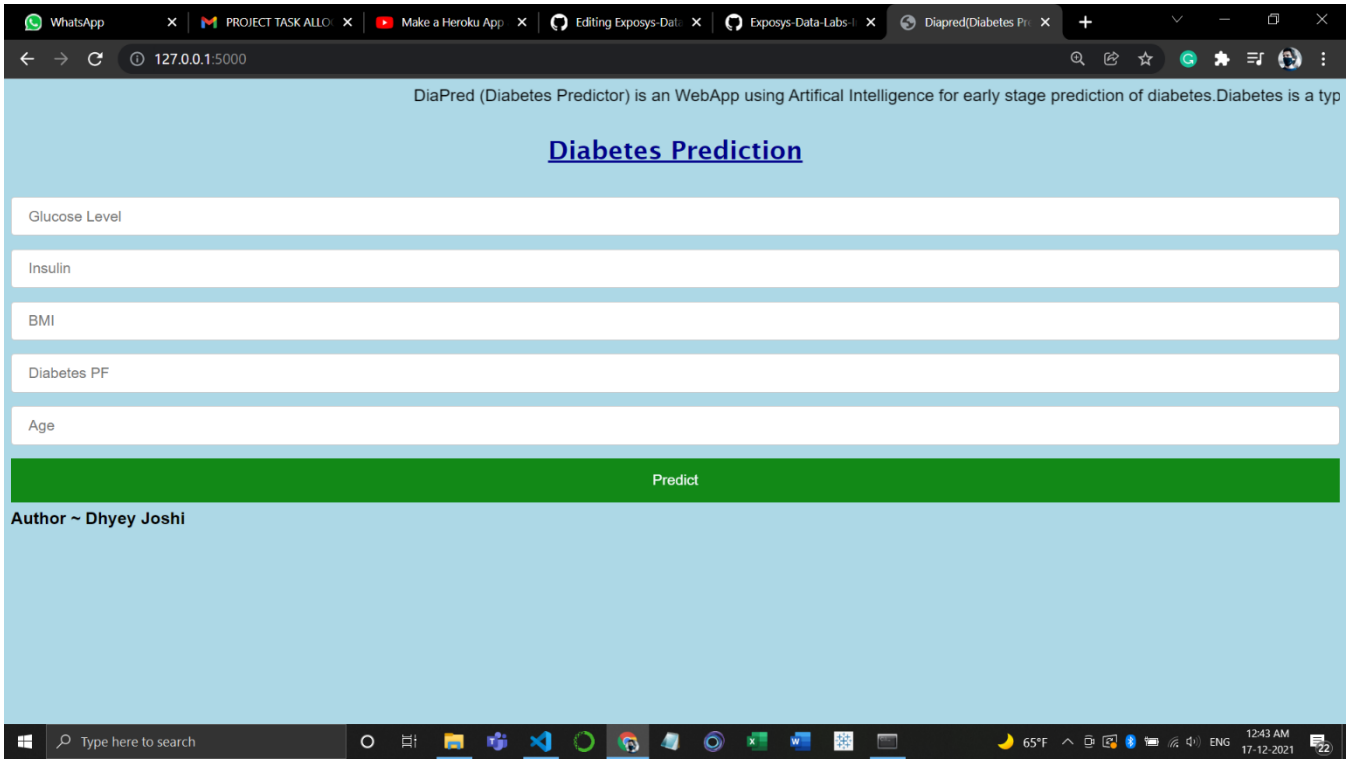
Limitations of the project

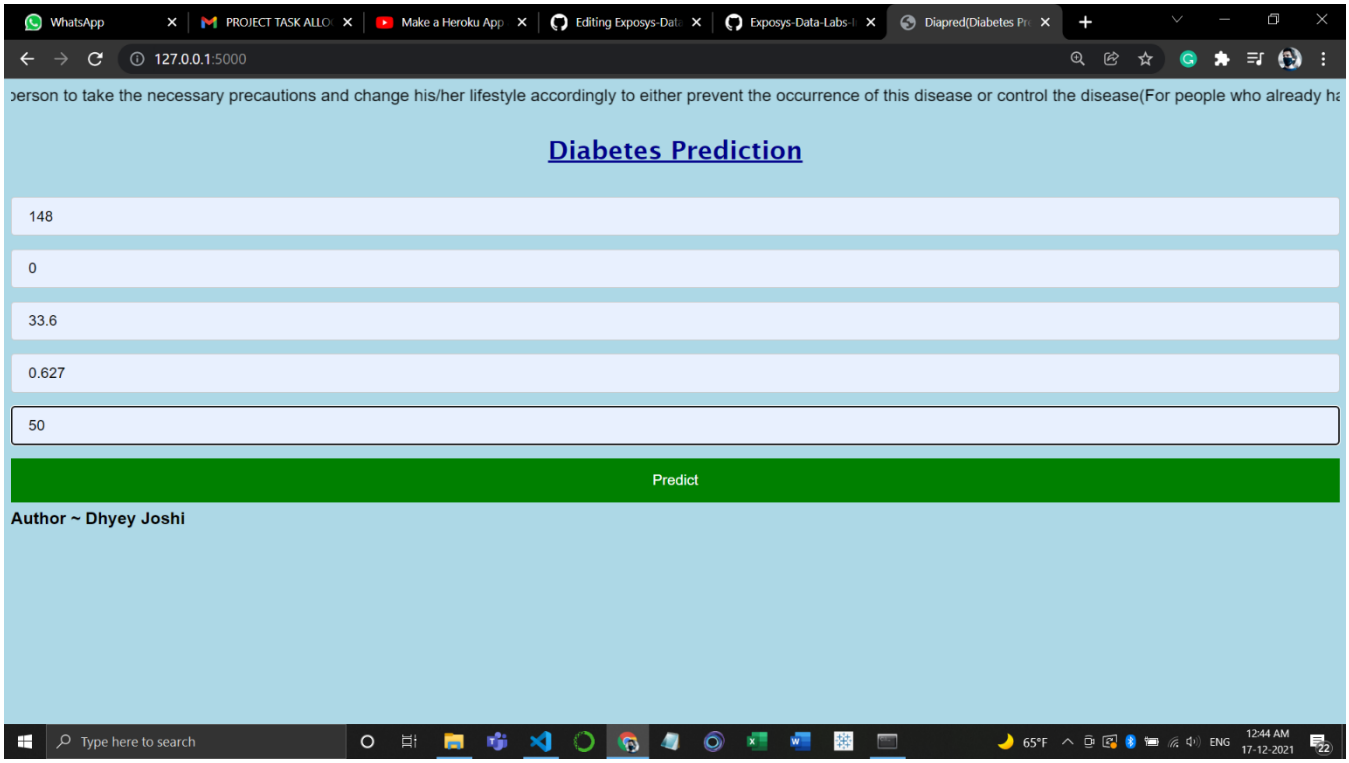
- Each narrow application needs to be specially trained
- Require large amounts of hand-crafted, structured training data
- Learning must generally be supervised: Training data must be tagged
- Require lengthy offline/ batch training
- Do not learn incrementally or interactively, in real time
- Poor transfer learning ability, re-usability of modules, and integration
- Systems are opaque, making them very hard to debug
- Performance cannot be audited or guaranteed at the 'long tail'
- They encode correlation, not causation or ontological relationships
- Do not encode entities, or spatial relationships between entities
- Only handle very narrow aspects of natural language
- Not well suited for high-level, symbolic reasoning or planning

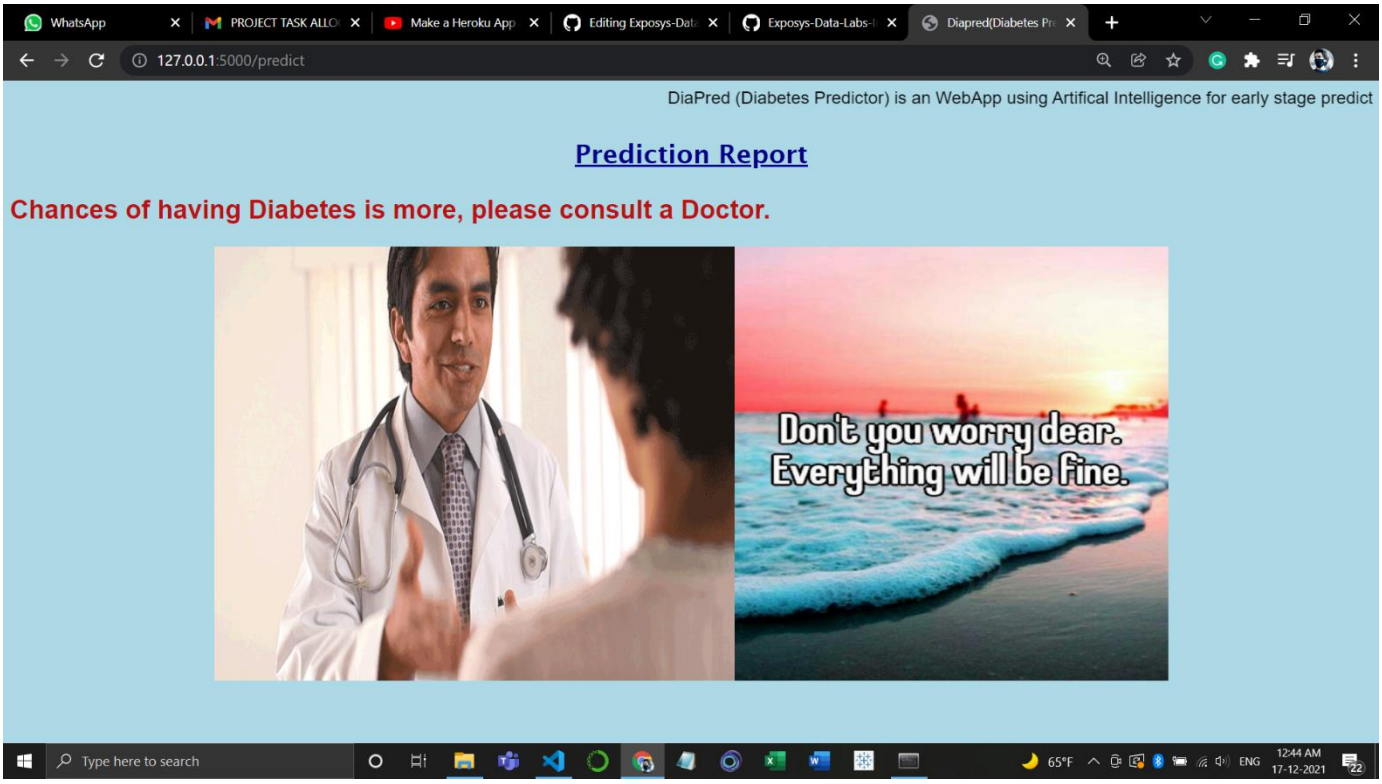
Chapter 6

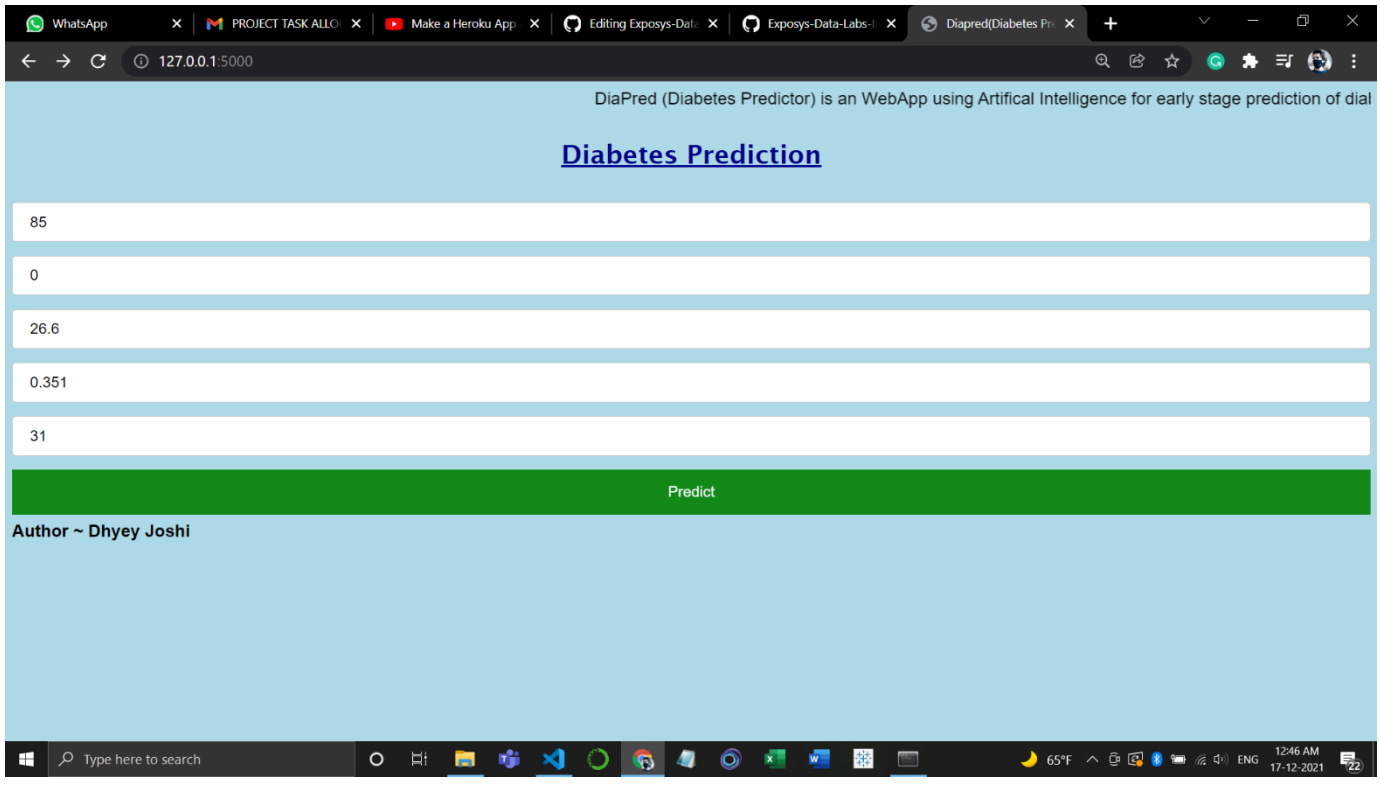
Project Outcomes

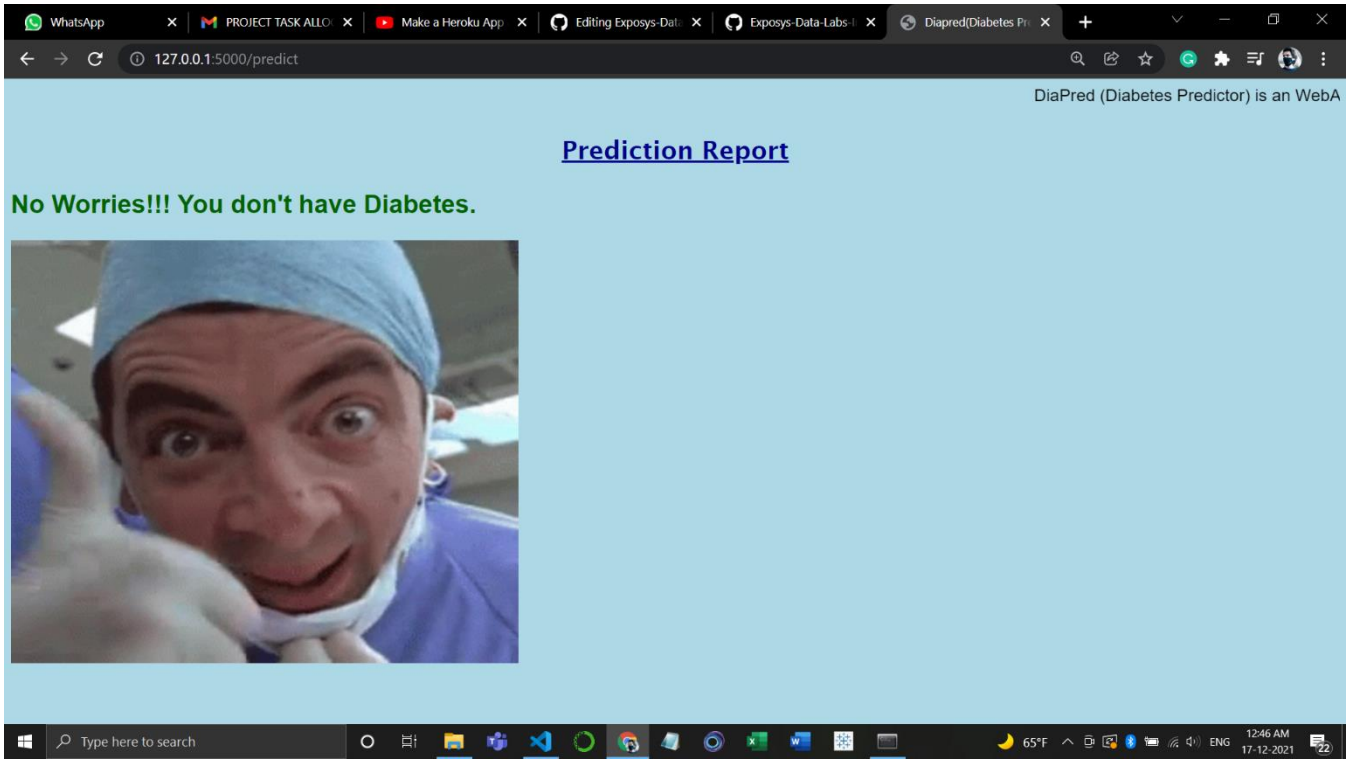


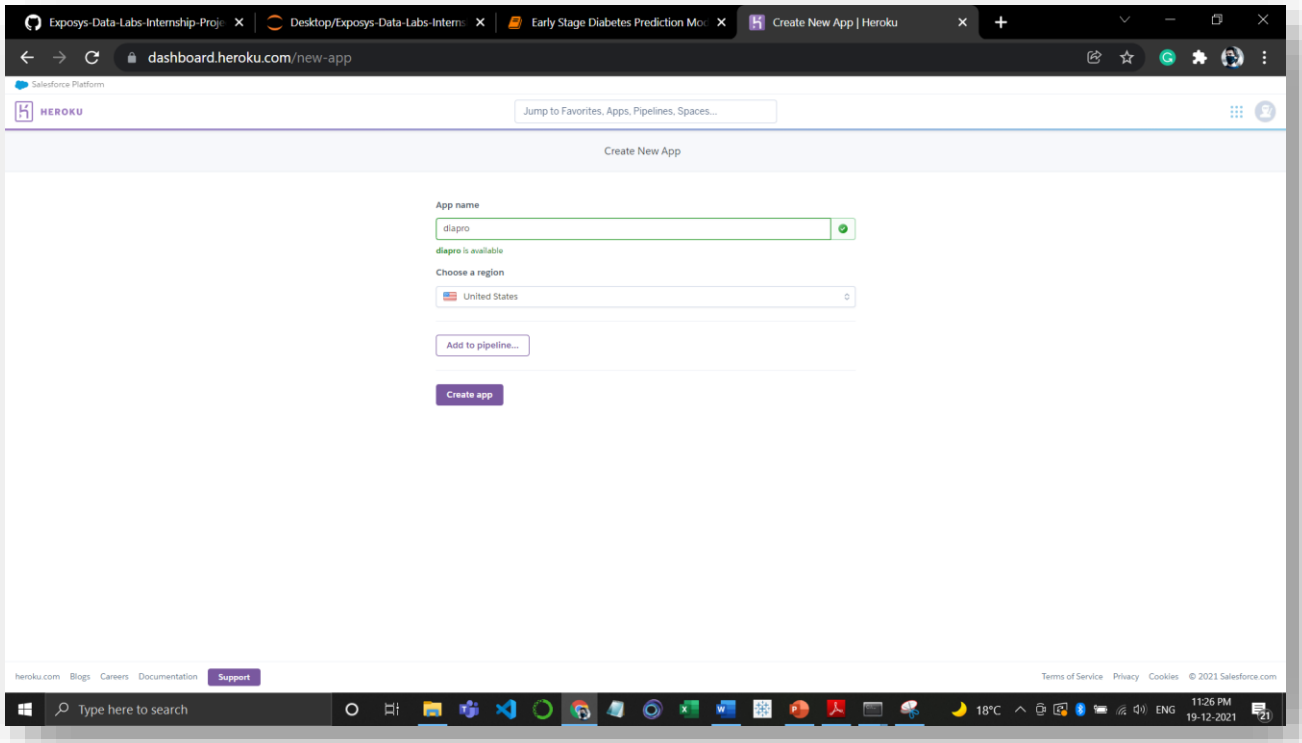

WhatsApp
PROJECT TASK ALLO
Make a Heroku App
Editing Exposys-Data
Exposys-Data-Labs-I
Diapred(Diabetes Pr
127.0.0.1:5000
DiaPred (Diabetes Predictor) is an WebApp using Artifical Intelligence for early stage prediction of diabetes.Diabetes is a typ
Diabetes Prediction
Glucose Level
Insulin
BMI
Diabetes PF
Age
Predict
Author ~ Dhyey Joshi
Type here to search
65°F
ENG
12:43 AM
17-12-2021

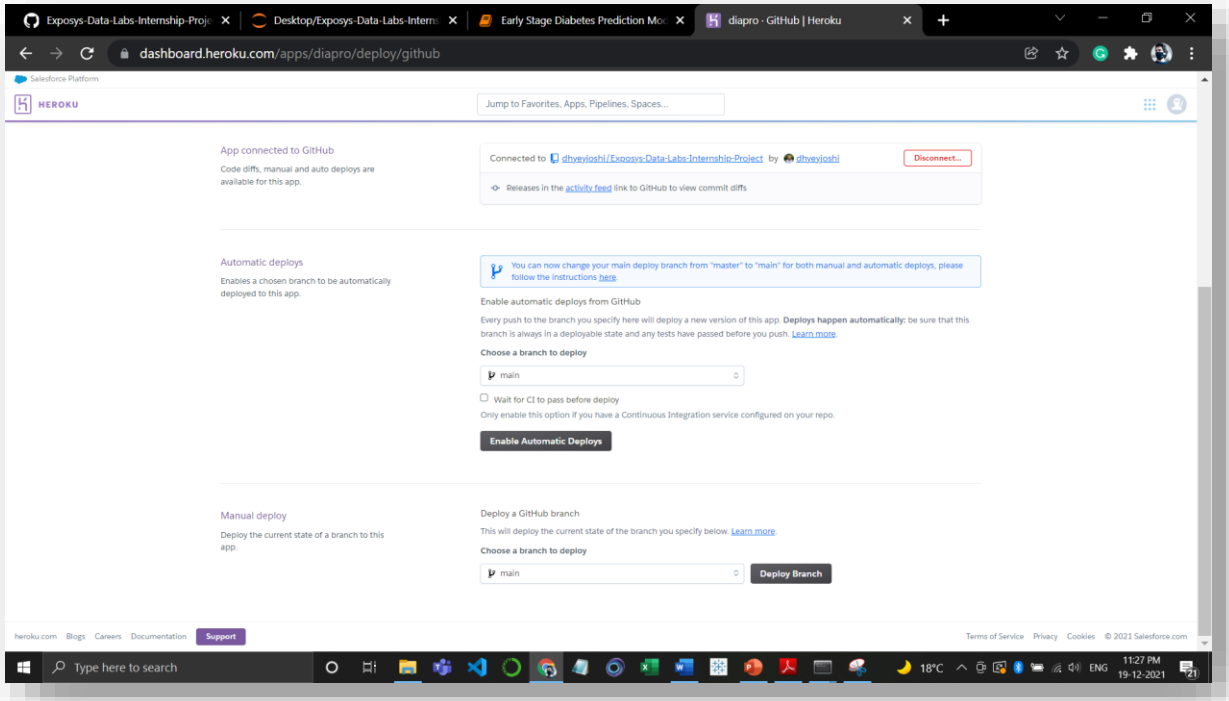

WhatsApp
PROJECT TASK ALLO
Make a Heroku App
Editing Exposys-Data
Exposys-Data-Labs-I
Diapred(Diabetes Pr
127.0.0.1:5000
person to take the necessary precautions and change his/her lifestyle accordingly to either prevent the occurrence of this disease or control the disease(For people who already ha
Diabetes Prediction
148
0
33.6
0.627
50
Predict
Author ~ Dhyey Joshi
Type here to search
65°F
ENG
12:44 AM
17-12-2021

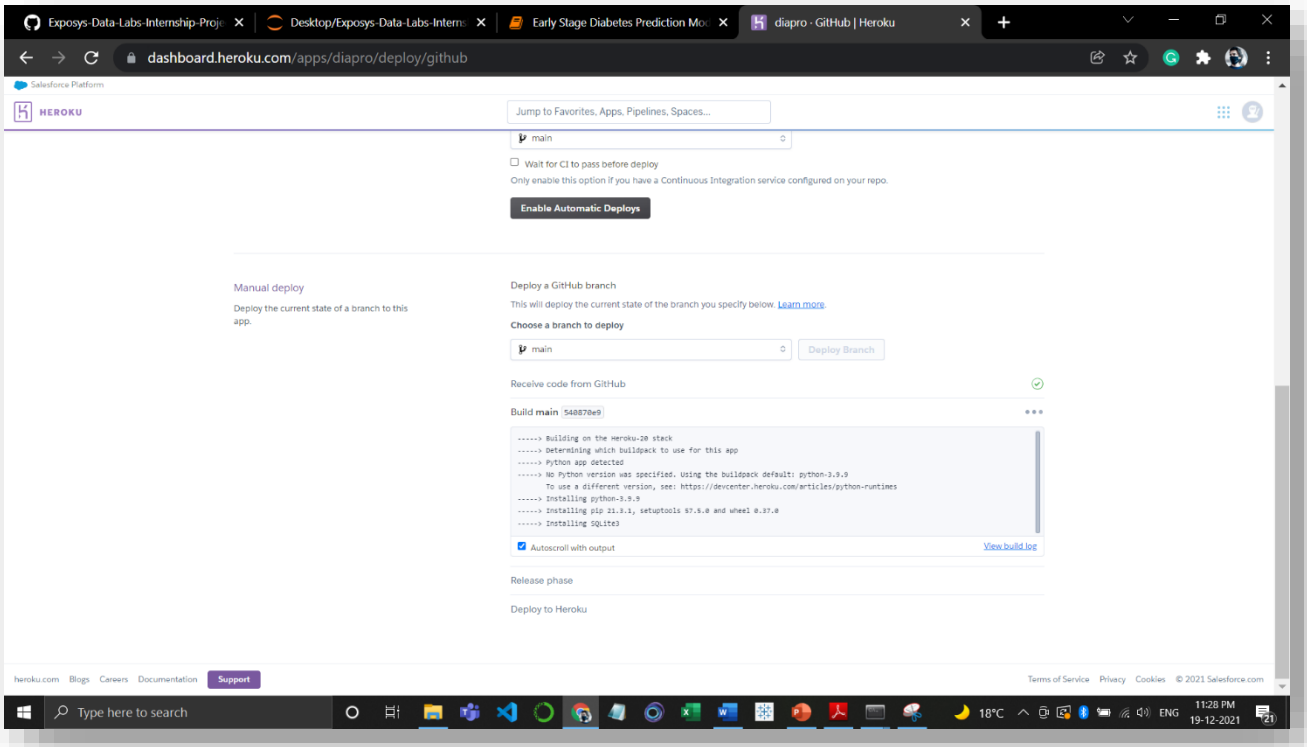

127.0.0.1:5000/predict
DiaPred (Diabetes Predictor) is an WebApp using Artifical Intelligence for early stage predict
Prediction Report
Chances of having Diabetes is more, please consult a Doctor.
Don't you worry dear.
Everything will be fine.
Type here to search
65°F
ENG
12:44 AM
17-12-2021

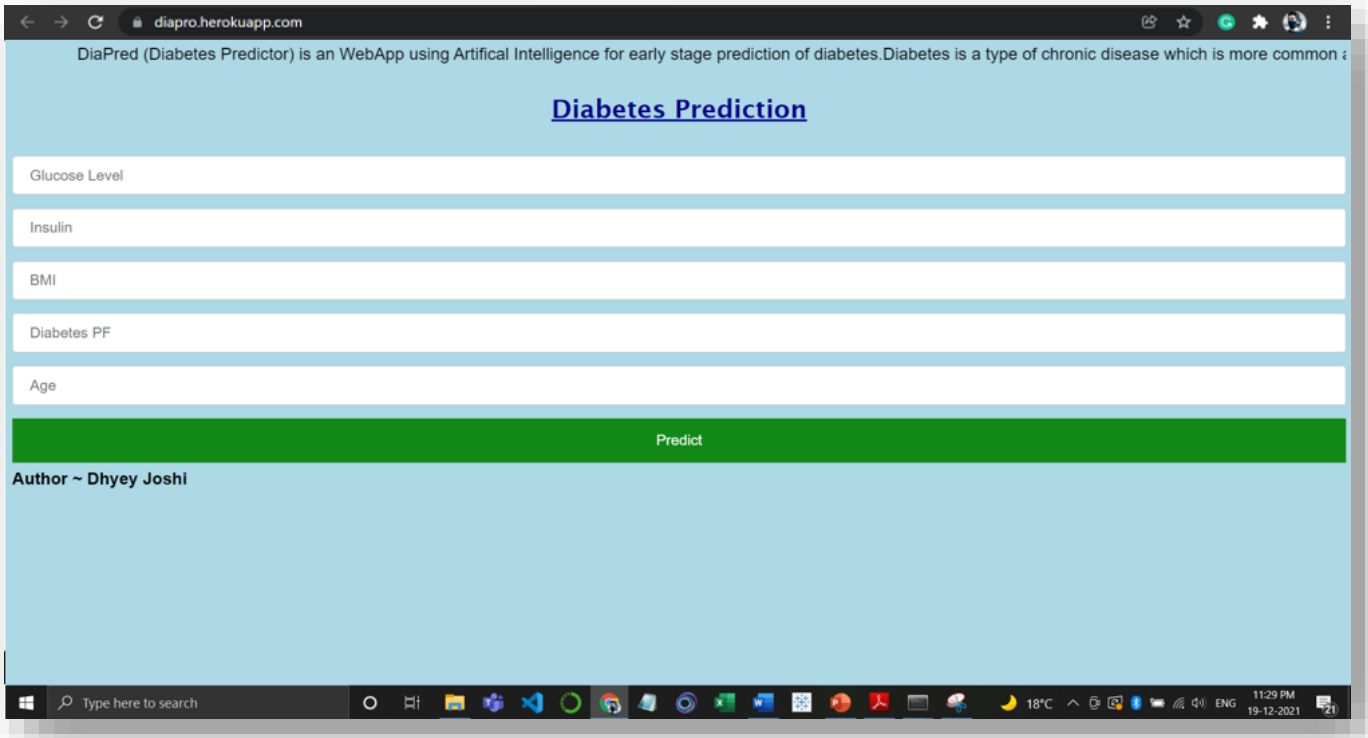

127.0.0.1:5000
DiaPred (Diabetes Predictor) is an WebApp using Artifical Intelligence for early stage prediction of dial
Diabetes Prediction
85
0
26.6
0.351
31
Predict
Author ~ Dhyey Joshi
Type here to search
65°F
ENG
12:46 AM
17-12-2021


WhatsApp
PROJECT TASK ALLO
Make a Heroku App
Editing Exposys-Data
Exposys-Data-Labs-I
Diapred(Diabetes Pr
127.0.0.1:5000/predict
DiaPred (Diabetes Predictor) is an WebA
Prediction Report
No Worries!!! You don't have Diabetes.
Type here to search
65°F
ENG
12:46 AM
17-12-2021


Exposys-Data-Labs-Internship-Proje
Desktop/Exposys-Data-Labs-Interns
Early Stage Diabetes Prediction Mod
Create New App | Heroku
dashboard.heroku.com/new-app
Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Create New App
App name
diapro
diapro is available
Choose a region
United States
Add to pipeline...
Create app
heroku.com Blogs Careers Documentation Support
Terms of Service Privacy Cookies © 2021 Salesforce.com
Type here to search
18°C
ENG
11:26 PM
19-12-2021


Exposys-Data-Labs-Internship-Proje
Desktop/Exposys-Data-Labs-Interns
Early Stage Diabetes Prediction Mod
diapro · GitHub | Heroku
dashboard.heroku.com/apps/diapro/deploy/github
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
App connected to GitHub
Code diffs, manual and auto deploys are available for this app.
Connected to dhyeyjoshi/Exposys-Data-Labs-Internship-Project by dhyeyjoshi
Disconnect...
Releases in the activity feed link to GitHub to view commit diffs
Automatic deploys
Enables a chosen branch to be automatically deployed to this app.
You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.
Enable automatic deploys from GitHub
Every push to the branch you specify here will deploy a new version of this app. Deploys happen automatically: be sure that this branch is always in a deployable state and any tests have passed before you push. Learn more.
Choose a branch to deploy
main
Wait for CI to pass before deploy
Only enable this option if you have a Continuous Integration service configured on your repo.
Enable Automatic Deploys
Manual deploy
Deploy the current state of a branch to this app.
Deploy a GitHub branch
This will deploy the current state of the branch you specify below. Learn more.
Choose a branch to deploy
main
Deploy Branch
heroku.com Blogs Careers Documentation Support
Terms of Service Privacy Cookies © 2021 Salesforce.com
Type here to search
18°C
ENG
11:27 PM
19-12-2021


Exposys-Data-Labs-Internship-Proje
Desktop/Exposys-Data-Labs-Interns
Early Stage Diabetes Prediction Mod
diapro · GitHub | Heroku
dashboard.heroku.com/apps/diapro/deploy/github
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
main
Wait for CI to pass before deploy
Only enable this option if you have a Continuous Integration service configured on your repo.
Enable Automatic Deploys
Manual deploy
Deploy the current state of a branch to this app.
Deploy a GitHub branch
This will deploy the current state of the branch you specify below. Learn more.
Choose a branch to deploy
main
Deploy Branch
Receive code from GitHub
Build main 540870e9
-----> Building on the Heroku-20 stack
-----> Determining which buildpack to use for this app
-----> Python app detected
-----> No Python version was specified. Using the buildpack default: python-3.9.9
To use a different version, see: https://devcenter.heroku.com/articles/python-runtimes
-----> Installing python-3.9.9
-----> Installing pip 21.3.1, setuptools 57.5.0 and wheel 0.37.0
-----> Installing SQLite3
Autoscroll with output
View build log
Release phase
Deploy to Heroku
heroku.com Blogs Careers Documentation Support
Terms of Service Privacy Cookies © 2021 Salesforce.com
Type here to search
18°C
ENG
11:28 PM
19-12-2021


diapro.herokuapp.com
DiaPred (Diabetes Predictor) is an WebApp using Artifical Intelligence for early stage prediction of diabetes.Diabetes is a type of chronic disease which is more common a
Diabetes Prediction
Glucose Level
Insulin
BMI
Diabetes PF
Age
Predict
Author ~ Dhyey Joshi
Type here to search
18°C
ENG
11:29 PM
19-12-2021

Chapter 7

Conclusion

7.1 Conclusion

Machine learning has the great ability to revolutionize the diabetes risk prediction with the help of advanced computational methods and availability of large amount of epidemiological and genetic diabetes risk dataset. Detection of diabetes in its early stages is the key for treatment. This work has described a machine learning approach to predicting diabetes levels. The technique may also help researchers to develop an accurate and effective tool that will reach at the table of clinicians to help them make better decision about the disease status.

Chapter 8

References

[1] Debadri Dutta, Debpriyo Paul, Parthajeet Ghosh, "Analyzing Feature Importances for Diabetes Prediction using Machine Learning". IEEE, pp 942-928, 2018.

[2] K.VijiyaKumar, B.Lavanya, I.Nirmala, S.Sofia Caroline, "Random Forest Algorithm for the Prediction of Diabetes ".Proceeding of International Conference on Systems Compu- tation Automation and Networking, 2019.

[3] Md. Faisal Faruque, Asaduzzaman, Iqbal H. Sarker, "Perfor- mance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus". International Conference on Electrical, Computer and Communication Engineering (ECCE), 7-9 Feb- ruary, 2019.

[4] Tejas N. Joshi, Prof. Pramila M. Chawan, "Diabetes Prediction Using Machine Learning Techniques".Int. Journal of Engineer- ing Research and Application, Vol. 8, Issue 1, (Part -II) Janu- ary 2018, pp.-09-13

[5] Nonso Nnamoko, Abir Hussain, David England, "Predicting Diabetes Onset: an Ensemble Supervised Learning Approach ". IEEE Congress on Evolutionary Computation (CEC), 2018.