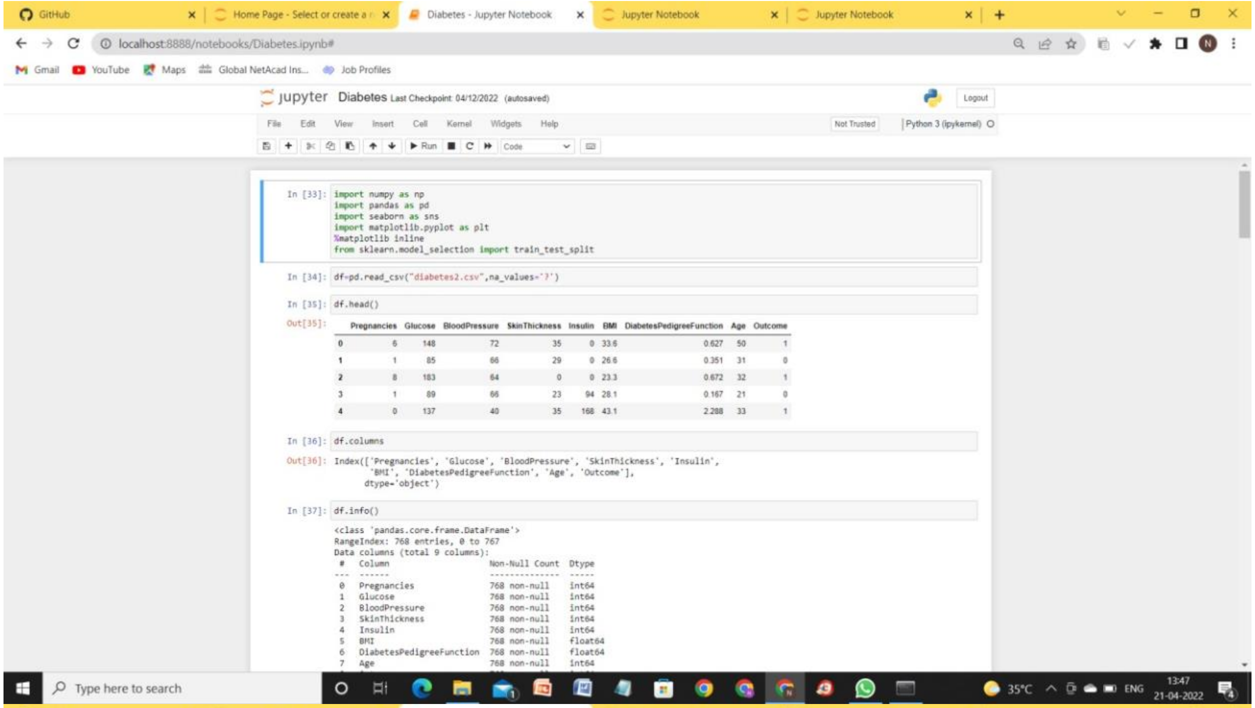
AI EXP NO. – 10 (SVM)

Name -HUSNA QASIM

Reg no. - RA1911031010138

Aim :- Implementing SVM on heart-attack dataset.

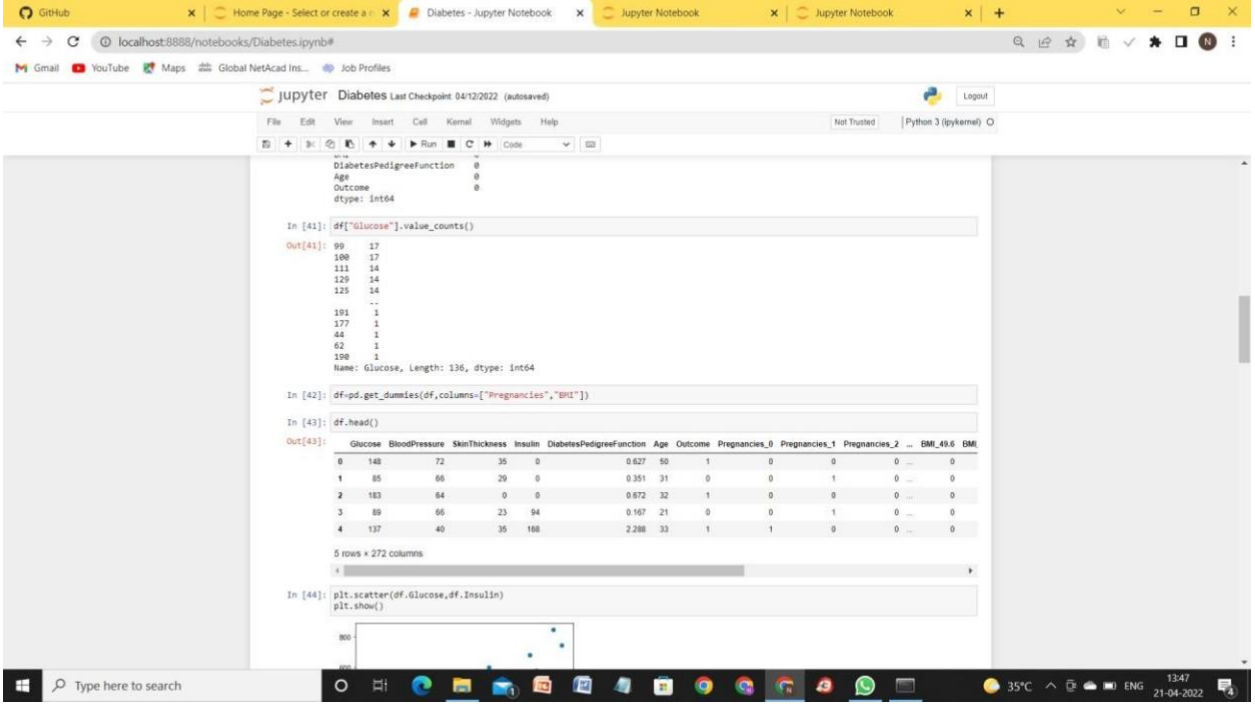


```
In [33]: import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline
         from sklearn.model_selection import train_test_split

In [34]: df=pd.read_csv("diabetes2.csv",na_values='?')

In [35]: df.head()
```

Out[35]:

	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

```
In [36]: df.columns

Out[36]: Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
                'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
               dtype='object')

In [37]: df.info()

         <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
```



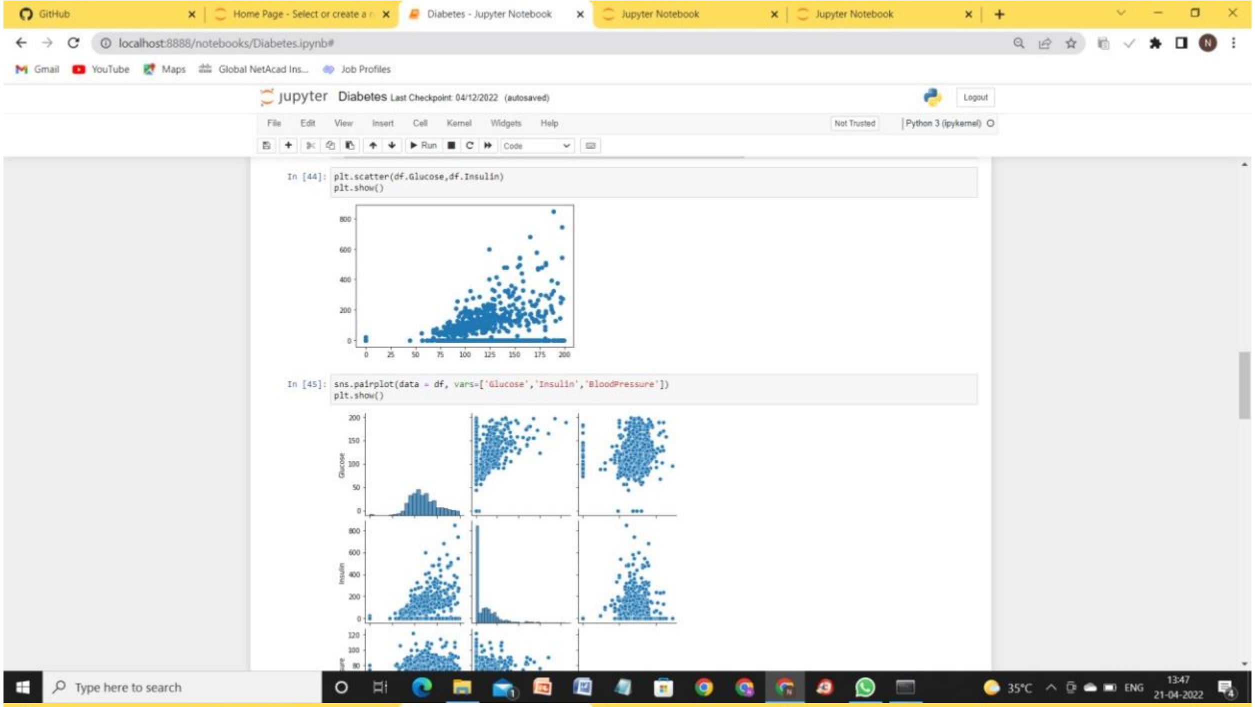
```
         DiabetesPedigreeFunction    0
         Age                         0
         Outcome                     0
         dtype: int64

In [41]: df["Glucose"].value_counts()

Out[41]: 99     17
         100    17
         111    14
         129    14
         125    14
                ..
         191     1
         177     1
         44      1
         62      1
         190     1
         Name: Glucose, Length: 136, dtype: int64

In [42]: df=pd.get_dummies(df,columns=["Pregnancies","BMI"])

In [43]: df.head()
```

Out[43]:

	Glucose	BloodPressure	SkinThickness	Insulin	DiabetesPedigreeFunction	Age	Outcome	Pregnancies_0	Pregnancies_1	Pregnancies_2	...	BMI_49.6	BMI
0	148	72	35	0	0.627	50	1	0	0	0	...	0	
1	85	66	29	0	0.351	31	0	0	1	0	...	0	
2	183	64	0	0	0.672	32	1	0	0	0	...	0	
3	89	66	23	94	0.167	21	0	0	1	0	...	0	
4	137	40	35	168	2.288	33	1	1	0	0	...	0	

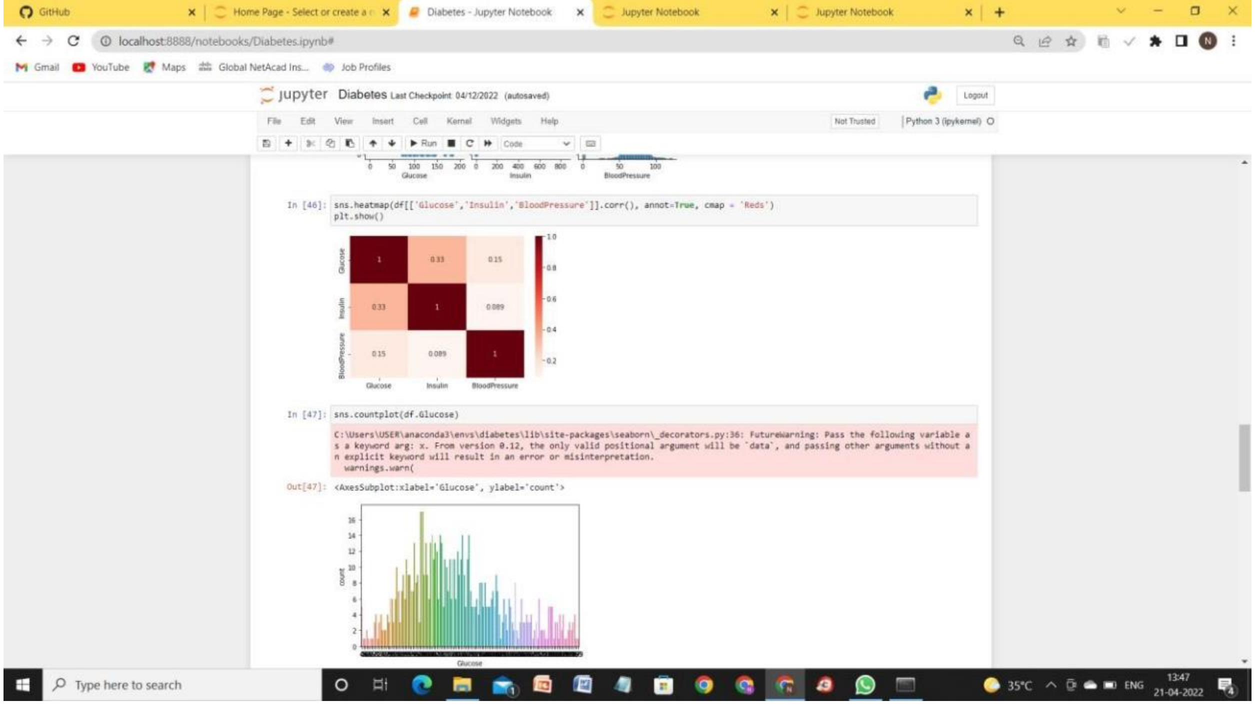
5 rows × 272 columns

```
In [44]: plt.scatter(df.Glucose,df.Insulin)
         plt.show()
```


GitHub
Home Page - Select or create a n
Diabetes - Jupyter Notebook
Jupyter Notebook
Jupyter Notebook
localhost:8888/notebooks/Diabetes.ipynb#
Gmail
YouTube
Maps
Global NetAcad Ins...
Job Profiles
jupyter Diabetes Last Checkpoint: 04/12/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Glucose
Insulin
Type here to search
35°C ENG
13:47
21-04-2022

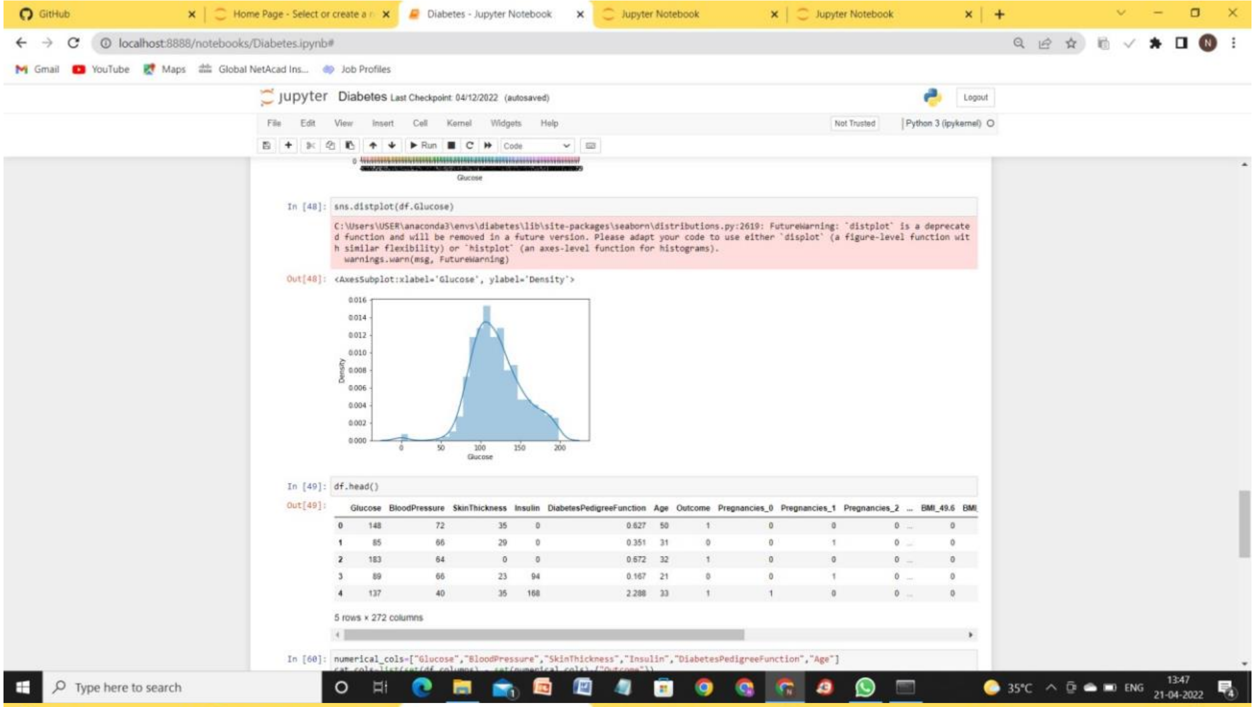
```
In [44]: plt.scatter(df.Glucose,df.Insulin)
         plt.show()
```

```
In [45]: sns.pairplot(data = df, vars=['Glucose','Insulin','BloodPressure'])
         plt.show()
```


GitHub
Home Page - Select or create a n
Diabetes - Jupyter Notebook
Jupyter Notebook
Jupyter Notebook
localhost:8888/notebooks/Diabetes.ipynb#
jupyter Diabetes Last Checkpoint: 04/12/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Glucose
Insulin
BloodPressure
1
0.33
0.15
0.089
1.0
0.8
0.6
0.4
0.2
count
Glucose
Type here to search
35°C ENG
13:47
21-04-2022

```
In [46]: sns.heatmap(df[['Glucose','Insulin','BloodPressure']].corr(), annot=True, cmap = 'Reds')
         plt.show()
```

```
In [47]: sns.countplot(df.Glucose)

C:\Users\USER\anaconda3\envs\diabetes\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable a
s a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without a
n explicit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[47]: <AxesSubplot:xlabel='Glucose', ylabel='count'>
```



Diabetes Last Checkpoint: 04/12/2022 (autosaved)

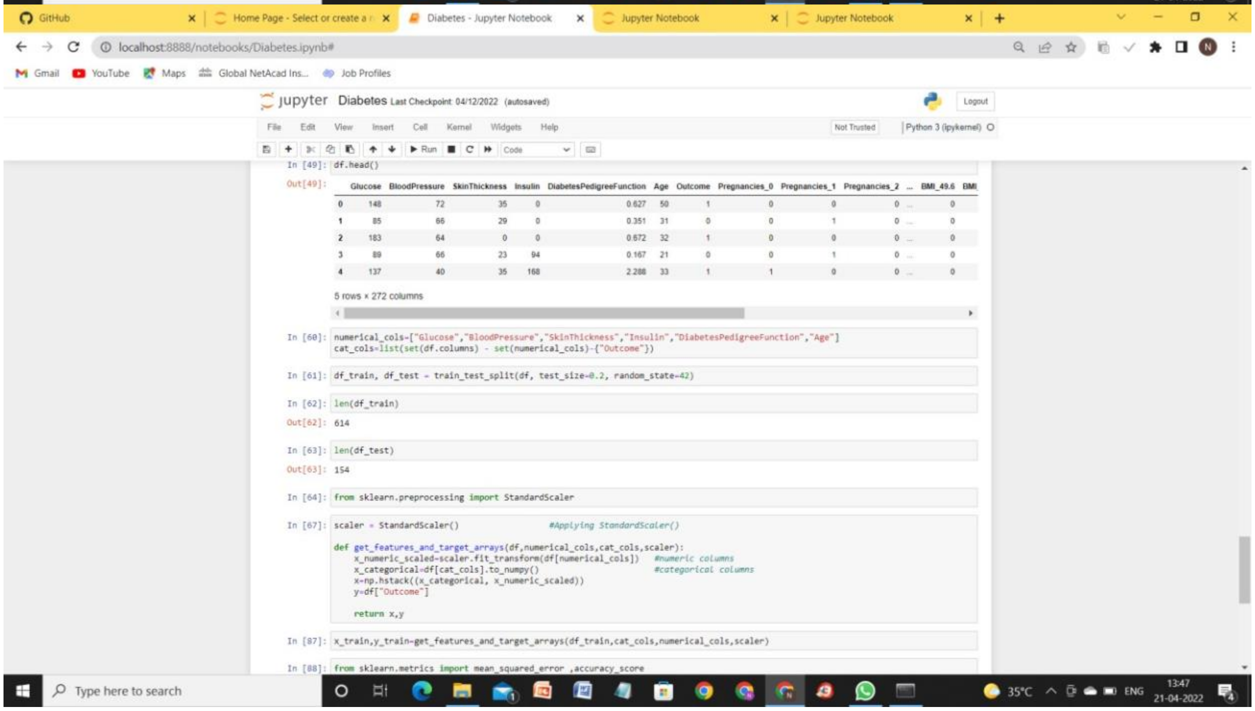
```
In [48]: sns.distplot(df.Glucose)

C:\Users\USER\anaconda3\envs\diabetes\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function wit
h similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

Out[48]: <AxesSubplot:xlabel='Glucose', ylabel='Density'>
```

```
In [49]: df.head()
Out[49]:
```

	Glucose	BloodPressure	SkinThickness	Insulin	DiabetesPedigreeFunction	Age	Outcome	Pregnancies_0	Pregnancies_1	Pregnancies_2	...	BMI_49.6	BMI
0	148	72	35	0	0.627	50	1	0	0	0	...	0	
1	85	66	29	0	0.351	31	0	0	1	0	...	0	
2	183	64	0	0	0.672	32	1	0	0	0	...	0	
3	89	66	23	94	0.167	21	0	0	1	0	...	0	
4	137	40	35	168	2.288	33	1	1	0	0	...	0	

5 rows × 272 columns

```
In [60]: numerical_cols=["Glucose","BloodPressure","SkinThickness","Insulin","DiabetesPedigreeFunction","Age"]
```



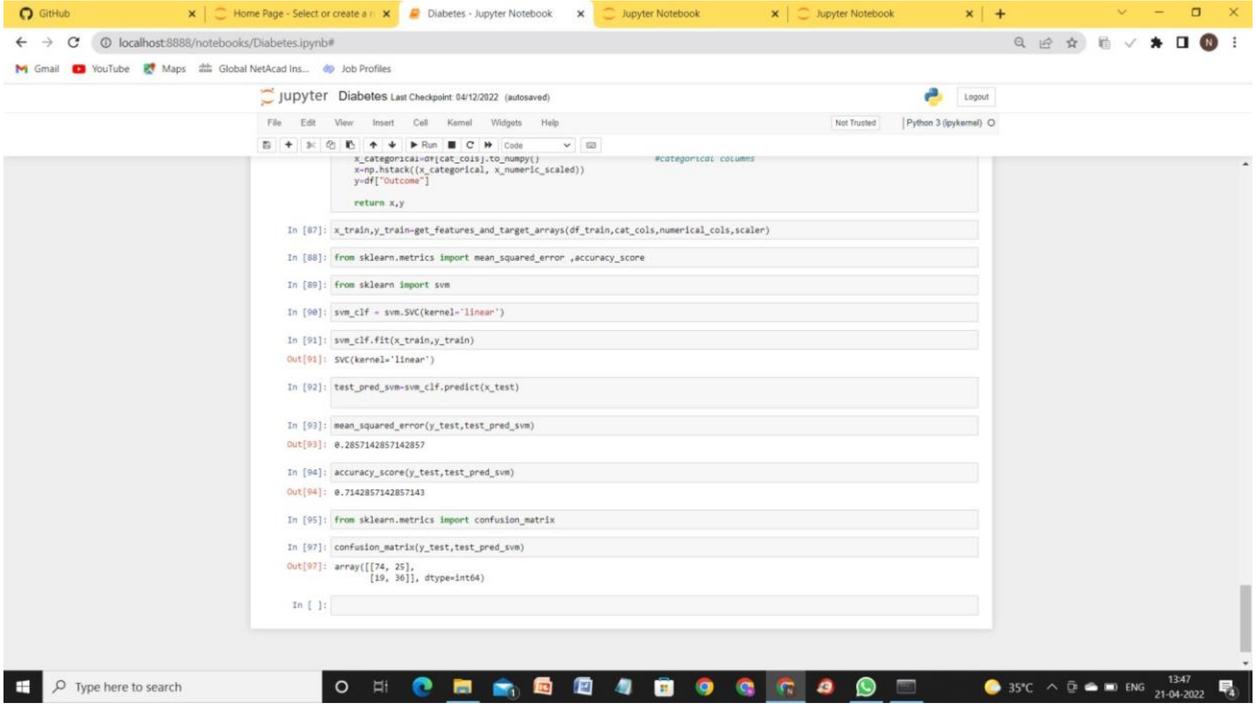
Diabetes Last Checkpoint: 04/12/2022 (autosaved)

In [49]: df.head()
Out[49]:

	Glucose	BloodPressure	SkinThickness	Insulin	DiabetesPedigreeFunction	Age	Outcome	Pregnancies_0	Pregnancies_1	Pregnancies_2	...	BMI_49.6	BMI
0	148	72	35	0	0.627	50	1	0	0	0	...	0	
1	85	66	29	0	0.351	31	0	0	1	0	...	0	
2	183	64	0	0	0.672	32	1	0	0	0	...	0	
3	89	66	23	94	0.167	21	0	0	1	0	...	0	
4	137	40	35	168	2.288	33	1	1	0	0	...	0	

5 rows × 272 columns

```
In [60]: numerical_cols=["Glucose","BloodPressure","SkinThickness","Insulin","DiabetesPedigreeFunction","Age"]
         cat_cols=list(set(df.columns) - set(numerical_cols)-{"Outcome"})

In [61]: df_train, df_test = train_test_split(df, test_size=0.2, random_state=42)

In [62]: len(df_train)

Out[62]: 614

In [63]: len(df_test)

Out[63]: 154

In [64]: from sklearn.preprocessing import StandardScaler

In [67]: scaler = StandardScaler()              #Applying StandardScaler()

         def get_features_and_target_arrays(df,numerical_cols,cat_cols,scaler):
             x_numeric_scaled=scaler.fit_transform(df[numerical_cols])   #numeric columns
             x_categorical=df[cat_cols].to_numpy()                       #categorical columns
             x=np.hstack((x_categorical, x_numeric_scaled))
             y=df["Outcome"]

             return x,y

In [87]: x_train,y_train=get_features_and_target_arrays(df_train,cat_cols,numerical_cols,scaler)

In [88]: from sklearn.metrics import mean_squared_error ,accuracy_score
```



GitHub
Home Page - Select or create a n
Diabetes - Jupyter Notebook
Jupyter Notebook
Jupyter Notebook
localhost:8888/notebooks/Diabetes.ipynb#
Gmail
YouTube
Maps
Global NetAcad Ins...
Job Profiles
jupyter Diabetes Last Checkpoint: 04/12/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
35°C
ENG
13:47
21-04-2022

```
    x_categorical=df[cat_cols].to_numpy()                    #categorical columns
    x=np.hstack((x_categorical, x_numeric_scaled))
    y=df["Outcome"]

    return x,y

In [87]: x_train,y_train=get_features_and_target_arrays(df_train,cat_cols,numerical_cols,scaler)

In [88]: from sklearn.metrics import mean_squared_error ,accuracy_score

In [89]: from sklearn import svm

In [90]: svm_clf = svm.SVC(kernel='linear')

In [91]: svm_clf.fit(x_train,y_train)
Out[91]: SVC(kernel='linear')

In [92]: test_pred_svm=svm_clf.predict(x_test)

In [93]: mean_squared_error(y_test,test_pred_svm)
Out[93]: 0.2857142857142857

In [94]: accuracy_score(y_test,test_pred_svm)
Out[94]: 0.7142857142857143

In [95]: from sklearn.metrics import confusion_matrix

In [97]: confusion_matrix(y_test,test_pred_svm)
Out[97]: array([[74, 25],
               [19, 36]], dtype=int64)

 In [ ]:
```

Result :- SVM is implemented successfully on jupyter notebook(anaconda).

https://github.com/Namrata2615/Heart_Attack_Prediction