```
!pip install catboost
```

```
Requirement already satisfied: catboost in /usr/local/lib/python3.10/dist-packages (1.2.5)
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from catboost) (0.20.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from catboost) (3.7.1)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.25.2)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-packages (from catboost) (2.0.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from catboost) (1.11.4)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from catboost) (5.15.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2.8
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (4.53.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (3.1.2)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly->catboost) (8.3.0)
```

```
from sklearn.model_selection  import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import roc_auc_score, roc_curve
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import f1_score
from sklearn import metrics
from matplotlib import pyplot
import seaborn as sns
sns.set(style= "darkgrid", color_codes = True)
from catboost import CatBoostClassifier
import pandas as pd
from numpy import mean
from numpy import std

import warnings
warnings.filterwarnings('ignore')
```

```
diabetes = pd.read_csv('/content/drive/MyDrive/Bangalore-Internship/diabetes.csv')
diabetes.head()
```

	Age	Gender	BMI	SBP	DBP	FPG	Chol	Tri	HDL	LDL	ALT	BUN	CCR	FFPG	smok
0	26	1	20.1	119	81	5.80	4.36	0.86	0.90	2.43	12.0	5.40	63.8	5.40	
1	40	1	17.7	97	54	4.60	3.70	1.02	1.50	2.04	9.2	3.70	70.3	4.10	
2	40	2	19.7	85	53	5.30	5.87	1.29	1.75	3.37	10.1	4.10	61.1	4.85	
3	43	1	23.1	111	71	4.50	4.05	0.74	1.27	2.60	36.5	4.38	73.4	5.30	
4	36	1	26.5	130	82	5.54	6.69	3.49	0.91	3.64	69.3	3.86	67.5	5.53	

```
diabetes.info()
```

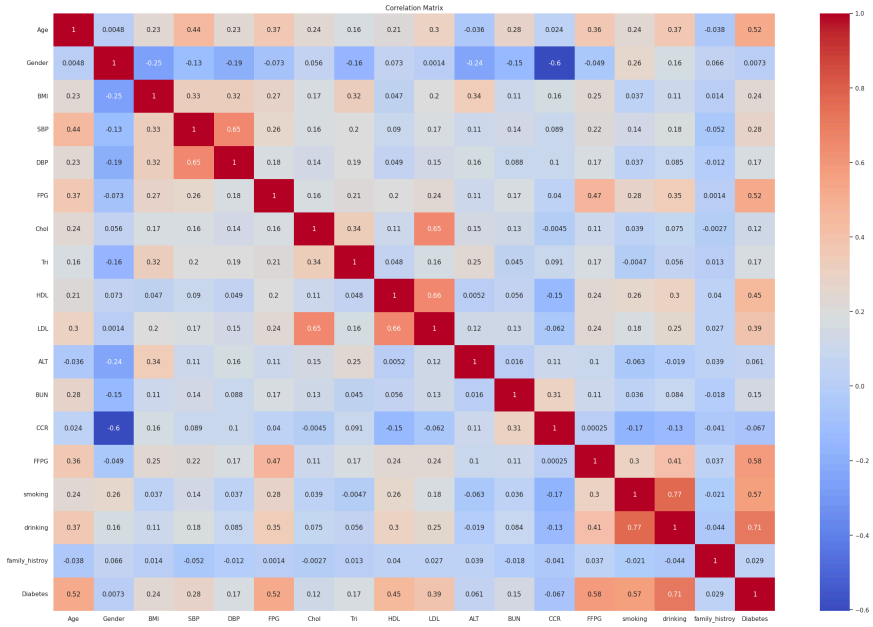
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4303 entries, 0 to 4302
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             4303 non-null   int64
 1   Gender          4303 non-null   int64
 2   BMI             4303 non-null   float64
 3   SBP             4303 non-null   int64
 4   DBP             4303 non-null   int64
 5   FPG             4303 non-null   float64
 6   Chol            4303 non-null   float64
 7   Tri             4303 non-null   float64
 8   HDL             4303 non-null   float64
 9   LDL             4303 non-null   float64
 10  ALT             4303 non-null   float64
 11  BUN             4303 non-null   float64
 12  CCR             4303 non-null   float64
 13  FFPG            4303 non-null   float64
 14  smoking         4303 non-null   float64
```

```
 15  drinking        4303 non-null   float64
 16  family_histroy  4303 non-null   int64
 17  Diabetes        4303 non-null   int64
dtypes: float64(12), int64(6)
memory usage: 605.2 KB
```

```
diabetes.describe()
```

	Age	Gender	BMI	SBP	DBP	FPG	43(
count	4303.000000	4303.000000	4303.000000	4303.000000	4303.000000	4303.000000	
mean	48.085057	1.351615	24.123923	123.219382	76.360446	5.226368	
std	14.686155	0.477530	3.397294	17.513858	11.004056	0.781089	
min	22.000000	1.000000	15.600000	72.000000	45.000000	1.780000	
25%	35.000000	1.000000	21.700000	111.000000	69.000000	4.700000	
50%	46.000000	1.000000	24.000000	122.000000	76.000000	5.140000	
75%	59.000000	2.000000	26.300000	134.000000	83.000000	5.700000	
max	93.000000	2.000000	45.800000	200.000000	134.000000	6.990000	

```
corr_matrix = diabetes.corr()
fig, ax = pyplot.subplots(figsize=(30, 20))
sns.heatmap(corr_matrix, cmap='coolwarm', annot=True, ax=ax)
ax.set_title('Correlation Matrix')
pyplot.show()
```



```
x= diabetes.drop(columns='Diabetes')
y= diabetes['Diabetes']
X_train, X_val, y_train, y_val=train_test_split(x,y, shuffle=True, random_state=12, test_size=0.1)
```

Double-click (or enter) to edit

```
scaler = StandardScaler()

X_V = X_val.values
scaled_x_train = scaler.fit_transform(X_train)
scaled_x_val = scaler.transform(X_V)


param_grid = {
    'iterations': [50, 100, 150],
    'learning_rate': [0.05, 0.01, 0.1],
    'max_depth': [2, 4, 6, 8],
    'l2_leaf_reg' : [2,4,6,8],
    'rsm' : [0.3,0.5,0.6],
}

model = CatBoostClassifier()

kfold = StratifiedKFold(n_splits=3, shuffle=True, random_state=7)
grid_search = GridSearchCV(model, param_grid=param_grid, cv=kfold, n_jobs=-1)
grid_search.fit(scaled_x_train,y_train)
print("Best score: {:.4f}".format(grid_search.best_score_))
print("Best parameters: {}".format(grid_search.best_params_))
```

```
Best parameters: {'iterations': 150, 'l2_leaf_reg': 6, 'learning_rate': 0.1, 'max_depth': 4, 'rsm': 0.6}
```

```
# setup hyperparameters for catboost
model = CatBoostClassifier(verbose=0, eval_metric='Accuracy',iterations=150, learning_rate=0.1 ,max_depth=4 ,l2_leaf_reg=6 ,rsm=0.6)
model.fit(scaled_x_train, y_train)
y_pred = model.predict(scaled_x_val)
```

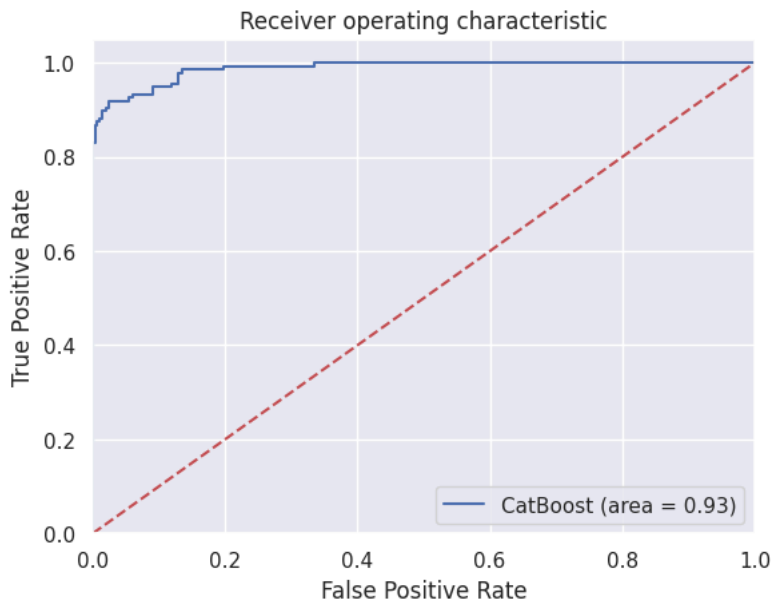
```
# define the evaluation method
cv = StratifiedKFold(n_splits=10)

# evaluate the model on the dataset
n_scores = cross_val_score(model, scaled_x_train, y_train, scoring='accuracy', cv=cv, n_jobs=-1)

# report performance
print('Mean Accuracy: %.3f (%.3f)' % (mean(n_scores), std(n_scores)))
print("Accuracy score (training): {0:.3f}".format(model.score(scaled_x_train, y_train)))
print("Accuracy score (validation): {0:.3f}".format(model.score(scaled_x_val, y_val)))
```

```
Mean Accuracy: 0.957 (0.012)
Accuracy score (training): 0.964
Accuracy score (validation): 0.954
```

```
CatBoost_roc_auc = roc_auc_score(y_val, model.predict(scaled_x_val))
fpr, tpr, thresholds = roc_curve(y_val, model.predict_proba(scaled_x_val)[:,1])
pyplot.figure()
pyplot.plot(fpr, tpr, label='CatBoost (area = %0.2f)' % CatBoost_roc_auc)
pyplot.plot([0, 1], [0, 1],'r--')
pyplot.xlim([0.0, 1.0])
pyplot.ylim([0.0, 1.05])
pyplot.xlabel('False Positive Rate')
pyplot.ylabel('True Positive Rate')
pyplot.title('Receiver operating characteristic')
pyplot.legend(loc="lower right")
pyplot.savefig('CatBoost_ROC')
pyplot.show()
```




```
#Confusion matrix, Accuracy, sensitivity and specificity
print(classification_report(y_val,y_pred))
cm = confusion_matrix(y_val, y_pred)
print('Confusion Matrix : \n', cm)

total=sum(sum(cm))
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print('f1 score:', f1_score(y_val, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97       295
           1       0.97      0.88      0.92       136

    accuracy                           0.95       431
   macro avg       0.96      0.93      0.94       431
weighted avg       0.95      0.95      0.95       431
```

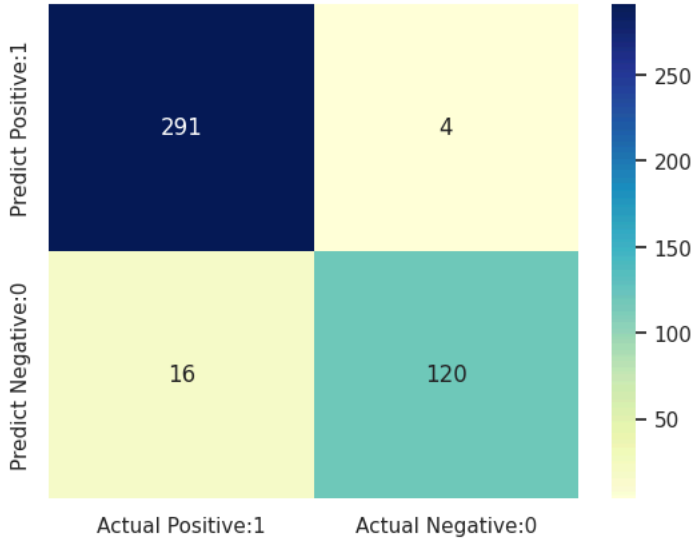
```
Confusion Matrix :
 [[291   4]
 [ 16 120]]
Sensitivity :  0.9864406779661017
Specificity :  0.8823529411764706
f1 score: 0.923076923076923
```

```
# visualize confusion matrix with seaborn heatmap

cm_matrix = pd.DataFrame(data=cm, columns=['Actual Positive:1', 'Actual Negative:0'],
                                 index=['Predict Positive:1', 'Predict Negative:0'])

sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='YlGnBu')
```

```
<Axes: >
```




```
# Save the model in CatBoost's native format
model.save_model('CatBoost_model.json')
```