# Heart Risk Prediction using supervised ML

import sklearn  
import numpy as np  
import io  
import pandas as pd  
import matplotlib.pyplot as plt  
import plotly.express as px  
import seaborn as sns  
sns.set(rc={'figure.figsize':(11.7,8.27)})  
import matplotlib.ticker as ticker  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import Normalizer  
from sklearn import metrics  
from sklearn.metrics import accuracy\_score,confusion\_matrix,classification\_report, roc\_auc\_score  
from sklearn.model\_selection import cross\_val\_score  
import warnings  
warnings.simplefilter(action='ignore')  
from sklearn.model\_selection import RandomizedSearchCV, GridSearchCV, StratifiedKFold  
kfold = StratifiedKFold(n\_splits=5)  
  
import util

## *Data Loading from Source*

### Using UCI heart risk data used at kaggle competieion

* Download UCI Cleveland heart risk data available on Kaggle <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>

df\_heart\_disease = pd.read\_csv('heart\_disease\_dataset\_UCI.csv')

## Data Pre-Processing:

* Data cleaning
* Remove NANs
* Look for missing and unimportant data

## Description of feature vectors:

* description

It's a clean, easy to understand set of data. However, the meaning of some of the column headers are not obvious. Here's what they mean,

* **age**: The person's age in years
* **sex**: The person's sex (1 = male, 0 = female)
* **cp**:  
  0 = typical angina 1 = atypical angina 2 = non-anginal pain 3 = asymptomatic
* **trestbps**: The person's resting blood pressure (mm Hg on admission to the hospital)
* **chol**: The person's cholesterol measurement in mg/dl
* **fbs**: The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)
* **restecg**: Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
* **thalach**: The person's maximum heart rate achieved
* **exang**: Exercise induced angina (1 = yes; 0 = no)
* **oldpeak**: ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot. See more here)
* **slope**: 0 = upsloping 1 = flat 2 = downsloping
* **ca**: The number of major vessels (0-3)
* **thal**: A blood disorder called thalassemia 0: NULL (dropped from the dataset previously) 1: fixed defect (no blood flow in some part of the heart) 2: normal blood flow 3: reversible defect (a blood flow is observed but it is not normal)
* **target**: Heart disease (0 = no, 1 = yes)

## Check for Null and NANs

### Need to convert object types to numerical types

df\_heart\_disease.dtypes

age int64  
sex int64  
cp int64  
trestbps int64  
chol int64  
fbs int64  
restecg int64  
thalach int64  
exang int64  
oldpeak float64  
slope int64  
ca int64  
thal int64  
target int64  
dtype: object

#df\_heart\_disease = df\_heart\_disease.astype(float, errors = 'raise')

#df\_heart\_disease.dtypes

df\_heart\_disease.isnull().sum()

age 0  
sex 0  
cp 0  
trestbps 0  
chol 0  
fbs 0  
restecg 0  
thalach 0  
exang 0  
oldpeak 0  
slope 0  
ca 0  
thal 0  
target 0  
dtype: int64

df\_heart\_disease.isna().sum()

age 0  
sex 0  
cp 0  
trestbps 0  
chol 0  
fbs 0  
restecg 0  
thalach 0  
exang 0  
oldpeak 0  
slope 0  
ca 0  
thal 0  
target 0  
dtype: int64

df\_heart\_disease.columns

Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',  
 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],  
 dtype='object')

input\_features = df\_heart\_disease.iloc[:,1:]  
output\_features = df\_heart\_disease.iloc[:,-1]

df\_heart\_disease.describe()

age sex cp trestbps chol fbs \  
count 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000   
mean 54.366337 0.683168 0.966997 131.623762 246.264026 0.148515   
std 9.082101 0.466011 1.032052 17.538143 51.830751 0.356198   
min 29.000000 0.000000 0.000000 94.000000 126.000000 0.000000   
25% 47.500000 0.000000 0.000000 120.000000 211.000000 0.000000   
50% 55.000000 1.000000 1.000000 130.000000 240.000000 0.000000   
75% 61.000000 1.000000 2.000000 140.000000 274.500000 0.000000   
max 77.000000 1.000000 3.000000 200.000000 564.000000 1.000000   
  
 restecg thalach exang oldpeak slope ca \  
count 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000   
mean 0.528053 149.646865 0.326733 1.039604 1.399340 0.729373   
std 0.525860 22.905161 0.469794 1.161075 0.616226 1.022606   
min 0.000000 71.000000 0.000000 0.000000 0.000000 0.000000   
25% 0.000000 133.500000 0.000000 0.000000 1.000000 0.000000   
50% 1.000000 153.000000 0.000000 0.800000 1.000000 0.000000   
75% 1.000000 166.000000 1.000000 1.600000 2.000000 1.000000   
max 2.000000 202.000000 1.000000 6.200000 2.000000 4.000000   
  
 thal target   
count 303.000000 303.000000   
mean 2.313531 0.544554   
std 0.612277 0.498835   
min 0.000000 0.000000   
25% 2.000000 0.000000   
50% 2.000000 1.000000   
75% 3.000000 1.000000   
max 3.000000 1.000000

### Consider rows with target = 1 or target = 0. Ignore other values

df\_heart\_disease = df\_heart\_disease[(df\_heart\_disease['target'] == 1) | (df\_heart\_disease['target'] == 0)]

df\_heart\_disease.describe()

age sex cp trestbps chol fbs \  
count 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000   
mean 54.366337 0.683168 0.966997 131.623762 246.264026 0.148515   
std 9.082101 0.466011 1.032052 17.538143 51.830751 0.356198   
min 29.000000 0.000000 0.000000 94.000000 126.000000 0.000000   
25% 47.500000 0.000000 0.000000 120.000000 211.000000 0.000000   
50% 55.000000 1.000000 1.000000 130.000000 240.000000 0.000000   
75% 61.000000 1.000000 2.000000 140.000000 274.500000 0.000000   
max 77.000000 1.000000 3.000000 200.000000 564.000000 1.000000   
  
 restecg thalach exang oldpeak slope ca \  
count 303.000000 303.000000 303.000000 303.000000 303.000000 303.000000   
mean 0.528053 149.646865 0.326733 1.039604 1.399340 0.729373   
std 0.525860 22.905161 0.469794 1.161075 0.616226 1.022606   
min 0.000000 71.000000 0.000000 0.000000 0.000000 0.000000   
25% 0.000000 133.500000 0.000000 0.000000 1.000000 0.000000   
50% 1.000000 153.000000 0.000000 0.800000 1.000000 0.000000   
75% 1.000000 166.000000 1.000000 1.600000 2.000000 1.000000   
max 2.000000 202.000000 1.000000 6.200000 2.000000 4.000000   
  
 thal target   
count 303.000000 303.000000   
mean 2.313531 0.544554   
std 0.612277 0.498835   
min 0.000000 0.000000   
25% 2.000000 0.000000   
50% 2.000000 1.000000   
75% 3.000000 1.000000   
max 3.000000 1.000000

# ***Exploratory Data Analysis***:

## Check outliers using Inter Quantile Range (IQR):

We are not going to remove outliers since they carry valuable info about certain types of patients. However, it can give us an idea on the necessecity of the exploratory data analysis.

Q1 = df\_heart\_disease.quantile(0.90)  
Q3 = df\_heart\_disease.quantile(0.10)  
IQR = Q3 - Q1

IQR

age -24.0  
sex -1.0  
cp -2.0  
trestbps -42.0  
chol -120.8  
fbs -1.0  
restecg -1.0  
thalach -60.6  
exang -1.0  
oldpeak -2.8  
slope -1.0  
ca -2.0  
thal -1.0  
target -1.0  
dtype: float64

df\_heart\_disease\_IQR = (df\_heart\_disease < (Q1 - 1.5 \* IQR)) |(df\_heart\_disease > (Q3 + 1.5 \* IQR))

df\_heart\_disease\_IQR

age sex cp trestbps chol fbs restecg thalach exang oldpeak \  
0 True True True True True True True True True True   
1 True True True True True True True True True True   
2 True True True True True True True True True True   
3 True True True True True True True True True True   
4 True True True True True True True True True True   
.. ... ... ... ... ... ... ... ... ... ...   
298 True True True True True True True True True True   
299 True True True True True True True True True True   
300 True True True True True True True True True True   
301 True True True True True True True True True True   
302 True True True True True True True True True True   
  
 slope ca thal target   
0 True True True True   
1 True True True True   
2 True True True True   
3 True True True True   
4 True True True True   
.. ... ... ... ...   
298 True True True True   
299 True True True True   
300 True True True True   
301 True True True True   
302 True True True True   
  
[303 rows x 14 columns]

### Drop Column which holds outlier

Row\_with\_outliers = df\_heart\_disease\_IQR.all(axis=0)  
indx = Row\_with\_outliers[Row\_with\_outliers== False].index.to\_list()  
indx

[]

df\_heart\_disease = df\_heart\_disease.drop(indx, axis = 1)

df\_heart\_disease.reset\_index(drop=True, inplace=True)

df\_heart\_disease.head()

age sex cp trestbps chol fbs restecg thalach exang oldpeak slope \  
0 63 1 3 145 233 1 0 150 0 2.3 0   
1 37 1 2 130 250 0 1 187 0 3.5 0   
2 41 0 1 130 204 0 0 172 0 1.4 2   
3 56 1 1 120 236 0 1 178 0 0.8 2   
4 57 0 0 120 354 0 1 163 1 0.6 2   
  
 ca thal target   
0 0 1 1   
1 0 2 1   
2 0 2 1   
3 0 2 1   
4 0 2 1

fig, out\_fig = plt.subplots(figsize = (20,15))  
plt.xticks(rotation=45)  
out\_fig = sns.boxplot(data = df\_heart\_disease, orient="h", palette="crest")

for i,col in enumerate(df\_heart\_disease.columns.values):  
 plt.subplot(5,3,i+1)  
 plt.scatter([i for i in range(303)],df\_heart\_disease[col].values.tolist())  
 plt.title(col)  
 fig,ax=plt.gcf(),plt.gca()  
 fig.set\_size\_inches(10,10)  
 plt.tight\_layout()  
plt.show()

%matplotlib inline  
fig, axis = plt.subplots(7,2,figsize=(10, 20));  
df\_heart\_disease.hist(ax=axis);

categorical\_val = []  
continous\_val = []  
  
for column in df\_heart\_disease.columns:  
 if len(df\_heart\_disease[column].unique()) <= 15:  
 categorical\_val.append(column)  
 else:  
 continous\_val.append(column)  
   
plt.figure(figsize=(15, 20))  
for i, column in enumerate(categorical\_val[:-1], 1):  
 plt.subplot(3, 3, i)  
 df\_heart\_disease[df\_heart\_disease["target"] == 0][column].hist(bins=35, color='blue', label='With Heart Disease', alpha=0.6)  
 df\_heart\_disease[df\_heart\_disease["target"] == 1][column].hist(bins=35, color='red', label='Without Heart Disease', alpha=0.6)  
 plt.legend()  
 plt.xlabel(column)  
 plt.legend(loc='upper right');

def data\_Sex(sex):  
 if sex == 0:  
 return 'female'  
 else:  
 return 'male'  
  
def data\_target(target):  
 if target == 0:  
 return 'No Heart Disease'  
 else:  
 return 'With Heart Disease'  
   
def data\_thal(thal):  
 if thal == 0:  
 return 'Non conclusive'  
 elif thal == 2:  
 return 'Normal'  
 elif thal == 1:  
 return 'Fixed Defect'  
 else:  
 return 'Reversible defect'  
  
def data\_cp(cp):  
 if cp == 0:  
 return 'Typical angina'  
 elif cp == 1:  
 return 'Atypical angina'  
 elif cp== 2:  
 return 'Non-anginal pain'  
 else:  
 return 'Asymptomatic'  
   
def data\_restecg(restecg):  
 if restecg == 1:  
 return 'Normal'  
 elif restecg == 0:  
 return 'left ventricular hypertrophy'  
 else:  
 return 'abnormality in ST-T wave'  
  
def data\_st\_slope(slope):  
 if slope == 0:  
 return 'downsloping'  
 elif slope == 1:  
 return 'flat'  
 else:  
 return 'upsloping'  
  
def data\_age(age):  
 if age < 30:  
 return 'young patients'  
 elif age >= 30 and age < 60:  
 return 'middle aged patients'  
 else:  
 return 'elderly patients'  
   
def data\_chol(chol):  
 if chol < 200:  
 return 'Normal Cholesterol Level'  
 else:  
 return 'High Cholesterol Level'   
   
df\_heart\_disease\_with\_catagoricalData = df\_heart\_disease.copy()  
df\_heart\_disease\_with\_catagoricalData['sex'] = df\_heart\_disease['sex'].apply(data\_Sex)  
df\_heart\_disease\_with\_catagoricalData['target'] = df\_heart\_disease['target'].apply(data\_target)  
df\_heart\_disease\_with\_catagoricalData['thal'] = df\_heart\_disease['thal'].apply(data\_thal)  
df\_heart\_disease\_with\_catagoricalData['cp'] = df\_heart\_disease['cp'].apply(data\_cp)  
df\_heart\_disease\_with\_catagoricalData['restecg'] = df\_heart\_disease['restecg'].apply(data\_restecg)  
df\_heart\_disease\_with\_catagoricalData['slope'] = df\_heart\_disease['slope'].apply(data\_st\_slope)  
df\_heart\_disease\_with\_catagoricalData['age\_class'] = df\_heart\_disease['age'].apply(data\_age)  
df\_heart\_disease\_with\_catagoricalData['chol\_level'] = df\_heart\_disease['chol'].apply(data\_chol)

col\_to\_move = df\_heart\_disease\_with\_catagoricalData.pop('target')  
df\_heart\_disease\_with\_catagoricalData.insert(len(df\_heart\_disease\_with\_catagoricalData.columns), 'target', col\_to\_move)

df\_heart\_disease\_with\_catagoricalData.head()

age sex cp trestbps chol fbs \  
0 63 male Asymptomatic 145 233 1   
1 37 male Non-anginal pain 130 250 0   
2 41 female Atypical angina 130 204 0   
3 56 male Atypical angina 120 236 0   
4 57 female Typical angina 120 354 0   
  
 restecg thalach exang oldpeak slope ca \  
0 left ventricular hypertrophy 150 0 2.3 downsloping 0   
1 Normal 187 0 3.5 downsloping 0   
2 left ventricular hypertrophy 172 0 1.4 upsloping 0   
3 Normal 178 0 0.8 upsloping 0   
4 Normal 163 1 0.6 upsloping 0   
  
 thal age\_class chol\_level \  
0 Fixed Defect elderly patients High Cholesterol Level   
1 Normal middle aged patients High Cholesterol Level   
2 Normal middle aged patients High Cholesterol Level   
3 Normal middle aged patients High Cholesterol Level   
4 Normal middle aged patients High Cholesterol Level   
  
 target   
0 With Heart Disease   
1 With Heart Disease   
2 With Heart Disease   
3 With Heart Disease   
4 With Heart Disease

sns.set(rc={'figure.figsize':(6,5), 'xtick.labelsize':10})  
sns.countplot(data= df\_heart\_disease\_with\_catagoricalData, x='age\_class',hue='target')  
plt.title('Relationship between age and risk of heart disease \n');  
plt.legend(loc='upper right');

sns.set(rc={'figure.figsize':(15, 7)})  
sns.countplot(data= df\_heart\_disease\_with\_catagoricalData[df\_heart\_disease\_with\_catagoricalData['target']=='With Heart Disease'], x='age',hue='sex')  
plt.title('Relationship between gender and risk of heart disease at all age \n');  
plt.legend(loc='upper right');

sns.set(rc={'figure.figsize':(6,5), 'xtick.labelsize':10})  
plot\_ = sns.countplot(data= df\_heart\_disease\_with\_catagoricalData[df\_heart\_disease\_with\_catagoricalData['target']=='With Heart Disease'], x='chol\_level', hue='sex')  
plt.title('Relationship between gender and risk of heart disease at all age with varying cholesterollevel \n');  
plt.tight\_layout();  
plt.legend(loc='upper right');

sns.set(rc={'figure.figsize':(6, 5), 'xtick.labelsize':10})  
plot\_ = sns.countplot(data= df\_heart\_disease\_with\_catagoricalData, x='chol\_level', hue='target')  
plt.tight\_layout();  
plt.legend(loc='upper right');

sns.set(rc={'figure.figsize':(6, 5), 'xtick.labelsize':10})  
sns.countplot(data= df\_heart\_disease\_with\_catagoricalData[df\_heart\_disease\_with\_catagoricalData['target'] == 'With Heart Disease'], x='sex',hue='thal')  
plt.title('Relationship between gender and risk of heart disorder "thalassemia" \n');  
plt.legend(loc='upper right');

sns.set(rc={'figure.figsize':(6, 5), 'xtick.labelsize':10})  
sns.countplot(data= df\_heart\_disease\_with\_catagoricalData, x='cp',hue='target')  
plt.title('Chest Pain varying with existence of heart disease \n');  
plt.legend(loc='upper right');

sns.set(rc={'figure.figsize':(6, 5), 'xtick.labelsize':10})  
sns.countplot(data= df\_heart\_disease\_with\_catagoricalData, x='chol\_level',hue='target')  
plt.title('Effect of cholesterol level and heart disease \n');

sns.countplot(data= df\_heart\_disease\_with\_catagoricalData, x='restecg',hue='target')  
plt.title('Resting electrocardiographic measurement varying with existence of heart disease \n');

plt.figure(figsize=(10,5))  
sns.pointplot(x=df\_heart\_disease['age'],y=df\_heart\_disease['thalach'],color='red',alpha=0.8)  
plt.xlabel('Age',fontsize = 15,color='blue')  
plt.xticks(rotation=45)  
plt.ylabel('Thalach',fontsize = 15,color='blue')  
plt.title('Age vs Thalach',fontsize = 15,color='blue')  
plt.grid()  
plt.show()

plt.figure(figsize=(10,5))  
sns.pointplot(x=df\_heart\_disease['age'],y=df\_heart\_disease['chol'],color='red', alpha=0.8)  
plt.xlabel('Age',fontsize = 15,color='blue')  
plt.xticks(rotation=45)  
plt.ylabel('Chol',fontsize = 15,color='blue')  
plt.title('Age vs Chol',fontsize = 15,color='blue')  
plt.grid()  
plt.show()

plt.figure(figsize=(10,5))  
sns.pointplot(x=df\_heart\_disease['age'],y=df\_heart\_disease['oldpeak'],color='red', alpha=0.8)  
plt.xlabel('Age',fontsize = 15,color='blue')  
plt.xticks(rotation=45)  
plt.ylabel('Chol',fontsize = 15,color='blue')  
plt.title('Age vs oldpeak',fontsize = 15,color='blue')  
plt.grid()  
plt.show()

plt.figure(figsize=(10,5))  
sns.pointplot(x=df\_heart\_disease['age'],y=df\_heart\_disease['trestbps'],color='red', alpha=0.8)  
plt.xlabel('Age',fontsize = 15,color='blue')  
plt.xticks(rotation=45)  
plt.ylabel('trestbps',fontsize = 15,color='blue')  
plt.title('Age vs trestbps',fontsize = 15,color='blue')  
plt.grid()  
plt.show()

plt.figure(figsize=(14,7))  
sns.heatmap(df\_heart\_disease.drop('target', axis=1).corr(),annot=True,cmap="magma",fmt='.2f');

import hvplot.pandas  
  
df\_heart\_disease.drop('target', axis=1).corrwith(df\_heart\_disease.target).hvplot.barh(  
 width=600, height=400,   
 title="Correlation between Heart Disease and Feature Vector",   
 ylabel='Correlation', xlabel='Feature Vector',  
)

### Features are not highly correlated. So it is reasonable to use these features to the machine learning model.

# *Important Feature Selection*:

### Univariate Feature Selection

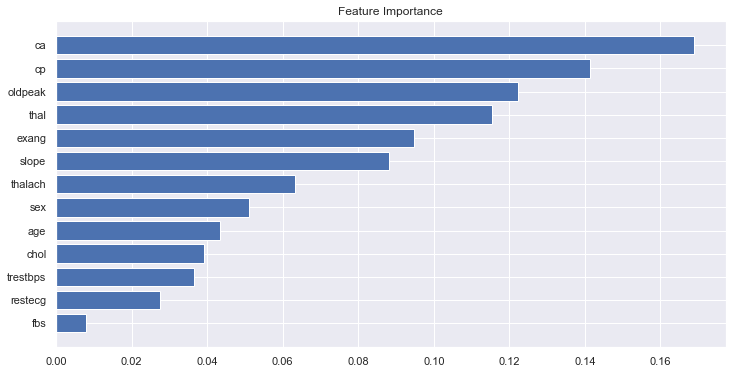
#Top x% features to consider  
fraction\_of\_top\_features = 0.8  
  
X = df\_heart\_disease.drop('target', axis=1)  
y = df\_heart\_disease['target']  
X\_train, X\_test,y\_train,y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)  
  
transformer = Normalizer()  
X\_train\_scaled = transformer.fit\_transform(X\_train)  
X\_test\_scaled = transformer.transform(X\_test)

from sklearn.feature\_selection import SelectKBest, chi2  
UV\_model = SelectKBest(chi2, k='all').fit(X\_train\_scaled, y\_train)  
mask = np.argsort(np.flip(UV\_model.scores\_)) #list of booleans for selected features  
best\_features\_SKBest = []   
best\_features\_SKBest = X\_train.columns[mask]   
  
best\_features\_SKBest

Index(['thalach', 'restecg', 'oldpeak', 'age', 'ca', 'cp', 'exang', 'thal',  
 'chol', 'slope', 'sex', 'trestbps', 'fbs'],  
 dtype='object')

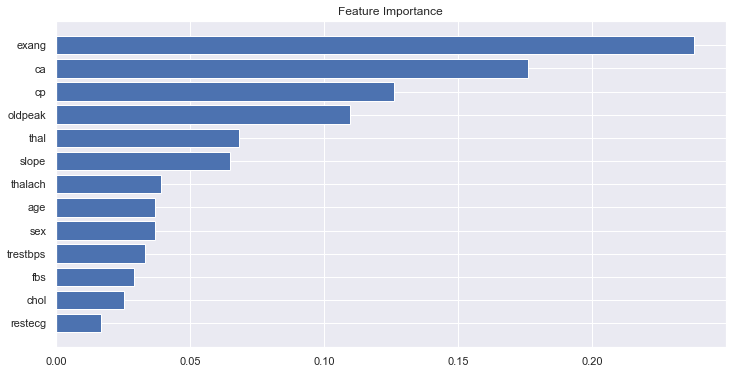
### Recursive feature elimination with Random Forest Classifier

from sklearn.ensemble import RandomForestClassifier  
from sklearn.inspection import permutation\_importance  
  
model\_rf = RandomForestClassifier(n\_estimators=1500, max\_depth=5)  
model\_rf.fit(X\_train\_scaled, y\_train)  
  
feature\_importance = model\_rf.feature\_importances\_  
sorted\_idx = np.argsort(feature\_importance)  
fig = plt.figure(figsize=(12, 6))  
plt.barh(range(len(sorted\_idx)), feature\_importance[sorted\_idx], align='center')  
plt.yticks(range(len(sorted\_idx)), np.array(X\_train.columns)[sorted\_idx])  
plt.title('Feature Importance')  
  
best\_features\_rf= X\_train.columns[np.flip(sorted\_idx)]  
best\_features\_rf = best\_features\_rf[:int(fraction\_of\_top\_features\*len(best\_features\_rf))]



from sklearn.inspection import permutation\_importance  
import xgboost as xgb  
  
model\_xgb = xgb.XGBClassifier(n\_estimators=1500, max\_depth=5, eta=0.05)  
model\_xgb.fit(X\_train\_scaled, y\_train)  
  
feature\_importance = model\_xgb.feature\_importances\_  
sorted\_idx = np.argsort(feature\_importance)  
fig = plt.figure(figsize=(12, 6))  
plt.barh(range(len(sorted\_idx)), feature\_importance[sorted\_idx], align='center')  
plt.yticks(range(len(sorted\_idx)), np.array(X\_train.columns)[sorted\_idx])  
plt.title('Feature Importance')  
  
best\_features\_xgb = X\_train.columns[np.flip(sorted\_idx)]  
best\_features\_xgb = best\_features\_xgb[:int(fraction\_of\_top\_features\*len(best\_features\_xgb))]

[16:52:33] WARNING: /Users/runner/work/xgboost/xgboost/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 set eval\_metric if you'd like to restore the old behavior.



### Recursive feature elimination with XGBoost Classifier

best\_feature\_list = list(set.intersection(set(best\_features\_SKBest), set(best\_features\_rf), set(best\_features\_xgb)))

### Use PCA to see how many features are important:

from sklearn.decomposition import PCA  
  
pca = PCA(n\_components = len(X\_train.columns)).fit(X\_train\_scaled)  
  
plt.plot(pca.explained\_variance\_ratio\_.cumsum(), lw=3, color='#087E8B')  
plt.title('Significance of principal components', size=20)  
plt.show()

Based on PCA seems like first 4 features are significant

### Best Feature Set:

best\_feature\_list

### Visualizing simple decision tree based classification using ***Best Feature Set***

from sklearn.tree import export\_graphviz  
from sklearn.tree import DecisionTreeClassifier  
from six import StringIO   
from IPython.display import Image   
import pydotplus  
  
clf = DecisionTreeClassifier(criterion="entropy", max\_depth=5)  
clf = clf.fit(X\_train[best\_feature\_list],y\_train)  
  
dot\_data = StringIO()  
export\_graphviz(clf, out\_file=dot\_data,   
 filled=False, rounded=True,  
 special\_characters=True,feature\_names = X\_train[best\_feature\_list].columns ,class\_names=['No Heart Risk','With Heart Risk'])  
graph = pydotplus.graph\_from\_dot\_data(dot\_data.getvalue())   
Image(graph.create\_png())

# ML Model Development for heart risk detection

### Get a tuned model that has comparatively highest accuracy:

Use features with numerical values only ***df\_heart\_disease***

### Use extracted best features from these schemes.

***best\_feature\_list***

X\_train = X\_train[best\_feature\_list]  
X\_test = X\_test[best\_feature\_list]

df\_heart\_disease = df\_heart\_disease[best\_feature\_list + ['target']]

## *XGBoost based classification for heart risk*

## Hyperparameter optimization using RandomizedSearchCV  
import xgboost

#Initialize Model  
clf\_mdl = xgboost.XGBClassifier(use\_label\_encoder = False, verbosity = 1, eval\_metric='logloss');  
  
params = {  
 'objective':['binary:logistic'],  
 'learning\_rate': [0.001, 0.005, 0.01, 0.1,0.3,0.5,0.7,1],   
 'max\_depth': [1, 2, 3, 4, 5, 6, 7],  
 'min\_child\_weight': [1e-5, 1e-3, 1e-2],  
 'subsample': [0.01, 0.1, 0.3,0.5,0.7,1],  
 'colsample\_bytree': [0.7,1],  
 'n\_estimators': [100, 200, 300, 400, 500, 1000]  
}  
  
#Initializing Grid Search with Stratified K Fold  
xgb\_ml = RandomizedSearchCV(clf\_mdl, param\_distributions=params, n\_jobs=4, cv=kfold)  
xgb\_ml.fit(X\_train,y\_train)

RandomizedSearchCV(cv=StratifiedKFold(n\_splits=5, random\_state=None, shuffle=False),  
 estimator=XGBClassifier(base\_score=None, booster=None,  
 colsample\_bylevel=None,  
 colsample\_bynode=None,  
 colsample\_bytree=None,  
 enable\_categorical=False,  
 eval\_metric='logloss', gamma=None,  
 gpu\_id=None, importance\_type=None,  
 interaction\_constraints=None,  
 learning\_rate=None,  
 max\_delta...  
 use\_label\_encoder=False,  
 validate\_parameters=None,  
 verbosity=1),  
 n\_jobs=4,  
 param\_distributions={'colsample\_bytree': [0.7, 1],  
 'learning\_rate': [0.001, 0.005, 0.01,  
 0.1, 0.3, 0.5, 0.7,  
 1],  
 'max\_depth': [1, 2, 3, 4, 5, 6, 7],  
 'min\_child\_weight': [1e-05, 0.001,  
 0.01],  
 'n\_estimators': [100, 200, 300, 400,  
 500, 1000],  
 'objective': ['binary:logistic'],  
 'subsample': [0.01, 0.1, 0.3, 0.5, 0.7,  
 1]})

xgb\_ml.best\_params\_

{'subsample': 0.3,  
 'objective': 'binary:logistic',  
 'n\_estimators': 500,  
 'min\_child\_weight': 0.001,  
 'max\_depth': 1,  
 'learning\_rate': 0.01,  
 'colsample\_bytree': 1}

Tuned\_model\_xgb = xgb\_ml.best\_estimator\_  
Tuned\_model\_xgb

XGBClassifier(base\_score=0.5, booster='gbtree', colsample\_bylevel=1,  
 colsample\_bynode=1, colsample\_bytree=1, enable\_categorical=False,  
 eval\_metric='logloss', gamma=0, gpu\_id=-1, importance\_type=None,  
 interaction\_constraints='', learning\_rate=0.01, max\_delta\_step=0,  
 max\_depth=1, min\_child\_weight=0.001, missing=nan,  
 monotone\_constraints='()', n\_estimators=500, n\_jobs=8,  
 num\_parallel\_tree=1, predictor='auto', random\_state=0,  
 reg\_alpha=0, reg\_lambda=1, scale\_pos\_weight=1, subsample=0.3,  
 tree\_method='exact', use\_label\_encoder=False,  
 validate\_parameters=1, verbosity=1)

from sklearn.model\_selection import cross\_val\_score  
score=cross\_val\_score(Tuned\_model\_xgb,X\_train,y\_train,cv=10)  
score

array([0.82608696, 0.86956522, 0.7826087 , 0.86956522, 0.7826087 ,  
 0.7826087 , 0.91304348, 0.90909091, 0.77272727, 0.77272727])

score.mean()

0.8280632411067194

prediction\_xgb = Tuned\_model\_xgb.predict(X\_test)

print("Model accuracy score:", accuracy\_score(y\_test, prediction\_xgb.astype(int)))

Model accuracy score: 0.868421052631579

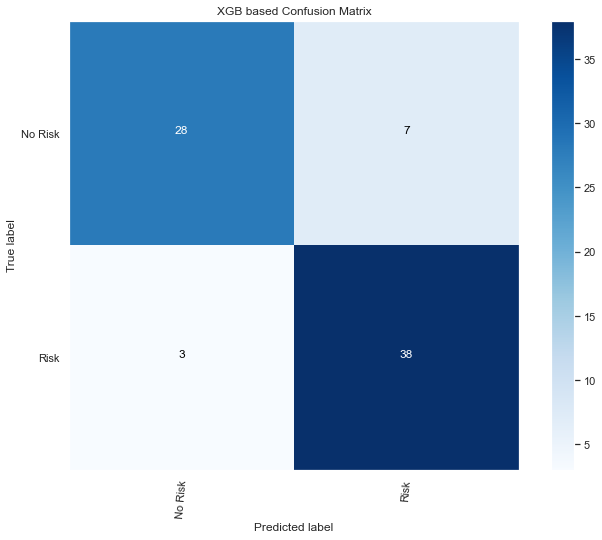
print("Model ROC\_AUC score:", roc\_auc\_score(y\_test, prediction\_xgb.astype(int)))

Model ROC\_AUC score: 0.8634146341463415

cm\_xgb = confusion\_matrix(y\_test, prediction\_xgb, labels=None)

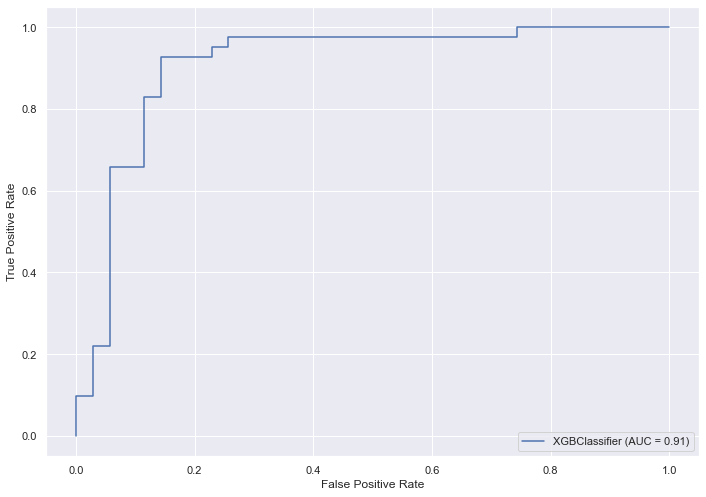
util.plot\_confusion\_matrix(cm\_xgb,classes=['No Risk', 'Risk'], title='XGB based Confusion Matrix')

Confusion Matrix  
confusion matrix:  
[[28 7]  
 [ 3 38]]

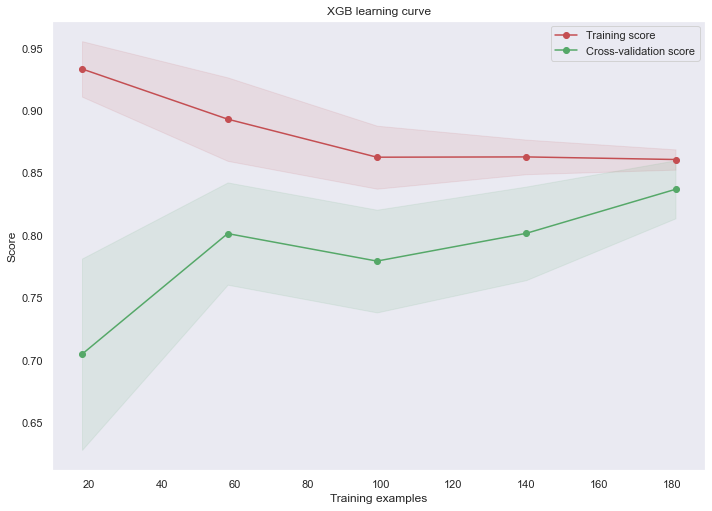


metrics.plot\_roc\_curve(Tuned\_model\_xgb, X\_test, y\_test)

<sklearn.metrics.\_plot.roc\_curve.RocCurveDisplay at 0x14087f7c0>



util.plot\_learning\_curve(estimator = Tuned\_model\_xgb, title = "XGB learning curve", X = X\_train, y = y\_train, cv = kfold);



## *Random Forest based classification for heart risk*

from sklearn.ensemble import RandomForestClassifier

#Initialize Model  
clf\_mdl\_2 = RandomForestClassifier()  
  
params = {'bootstrap': [True, False],  
 'max\_depth': range(1,10, 1),  
 'max\_features': ['auto', 'sqrt'],  
 'min\_samples\_leaf': [1, 2, 4],  
 'min\_samples\_split': [2, 5, 10],  
 'n\_estimators': [100, 200, 300, 400, 500]}  
  
#Initializing Grid Search with Stratified K Fold  
rf\_ml = RandomizedSearchCV(clf\_mdl\_2, param\_distributions=params, n\_jobs=-1, cv=kfold)  
rf\_ml.fit(X\_train,y\_train)

RandomizedSearchCV(cv=StratifiedKFold(n\_splits=5, random\_state=None, shuffle=False),  
 estimator=RandomForestClassifier(), n\_jobs=-1,  
 param\_distributions={'bootstrap': [True, False],  
 'max\_depth': range(1, 10),  
 'max\_features': ['auto', 'sqrt'],  
 'min\_samples\_leaf': [1, 2, 4],  
 'min\_samples\_split': [2, 5, 10],  
 'n\_estimators': [100, 200, 300, 400,  
 500, 1500]})

rf\_ml.best\_params\_

{'n\_estimators': 400,  
 'min\_samples\_split': 5,  
 'min\_samples\_leaf': 4,  
 'max\_features': 'auto',  
 'max\_depth': 2,  
 'bootstrap': True}

Tuned\_model\_rf = rf\_ml.best\_estimator\_  
Tuned\_model\_rf

RandomForestClassifier(max\_depth=2, min\_samples\_leaf=4, min\_samples\_split=5,  
 n\_estimators=400)

score=cross\_val\_score(Tuned\_model\_rf,X\_train,y\_train,cv=10)  
score

array([0.82608696, 0.86956522, 0.7826087 , 0.86956522, 0.7826087 ,  
 0.73913043, 0.86956522, 0.90909091, 0.77272727, 0.81818182])

score.mean()

0.8239130434782608

prediction\_rf = Tuned\_model\_rf.predict(X\_test)

print("Model accuracy score:", accuracy\_score(y\_test, prediction\_rf.astype(int)))

Model accuracy score: 0.8421052631578947

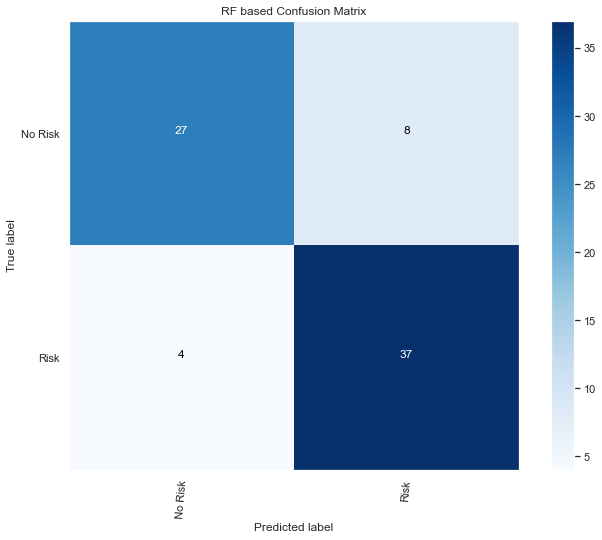
print("Model ROC\_AUC score:", roc\_auc\_score(y\_test, prediction\_rf.astype(int)))

Model ROC\_AUC score: 0.8369337979094077

cm\_rf = confusion\_matrix(y\_test, prediction\_rf, labels=None)

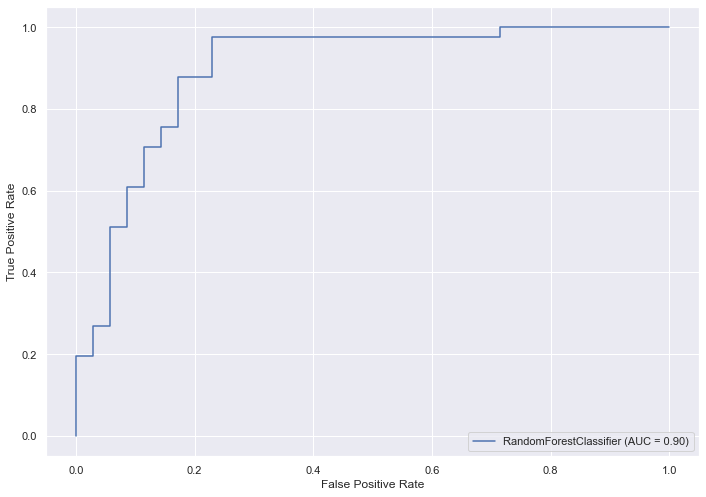
util.plot\_confusion\_matrix(cm\_rf,classes=['No Risk', 'Risk'], title='RF based Confusion Matrix')

Confusion Matrix  
confusion matrix:  
[[27 8]  
 [ 4 37]]



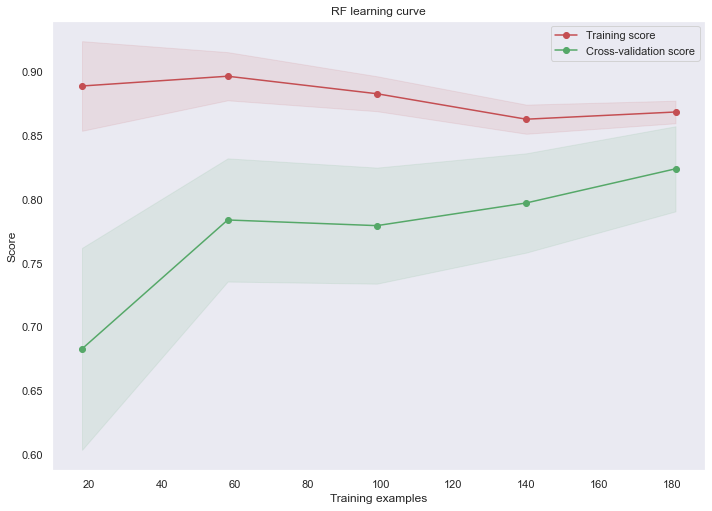
metrics.plot\_roc\_curve(Tuned\_model\_rf, X\_test, y\_test)

<sklearn.metrics.\_plot.roc\_curve.RocCurveDisplay at 0x1401239a0>



util.plot\_learning\_curve(estimator = Tuned\_model\_rf, title = "RF learning curve", X = X\_train, y = y\_train, cv = kfold)

<module 'matplotlib.pyplot' from '/Users/rrahman/Library/Python/3.8/lib/python/site-packages/matplotlib/pyplot.py'>



## *Logistic Regression based classification for heart risk*

from sklearn.linear\_model import LogisticRegression

#Initialize Model  
clf\_mdl\_3 = LogisticRegression()  
  
params = {  
 "max\_iter": range(100,500,2),  
 "solver" : ['newton-cg', 'lbfgs', 'liblinear'],  
 "C": [0.5, 0.1, 1.0]  
}  
  
lgr\_ml = RandomizedSearchCV(clf\_mdl\_3, param\_distributions=params, n\_jobs=-1, cv=kfold)  
lgr\_ml.fit(X\_train,y\_train)

RandomizedSearchCV(cv=StratifiedKFold(n\_splits=5, random\_state=None, shuffle=False),  
 estimator=LogisticRegression(), n\_jobs=-1,  
 param\_distributions={'C': [0.5, 0.1, 1.0],  
 'max\_iter': range(100, 500, 2),  
 'solver': ['newton-cg', 'lbfgs',  
 'liblinear']})

lgr\_ml.best\_params\_

{'solver': 'liblinear', 'max\_iter': 142, 'C': 0.5}

Tuned\_model\_lgr = lgr\_ml.best\_estimator\_  
Tuned\_model\_lgr

LogisticRegression(C=0.5, max\_iter=142, solver='liblinear')

score=cross\_val\_score(Tuned\_model\_lgr,X\_train,y\_train,cv=10)  
score

array([0.69565217, 0.7826087 , 0.86956522, 0.86956522, 0.82608696,  
 0.73913043, 0.95652174, 0.86363636, 0.81818182, 0.81818182])

score.mean()

0.8239130434782609

prediction\_lgr = Tuned\_model\_lgr.predict(X\_test)

print("Model accuracy score:", accuracy\_score(y\_test, prediction\_lgr.astype(int)))

Model accuracy score: 0.881578947368421

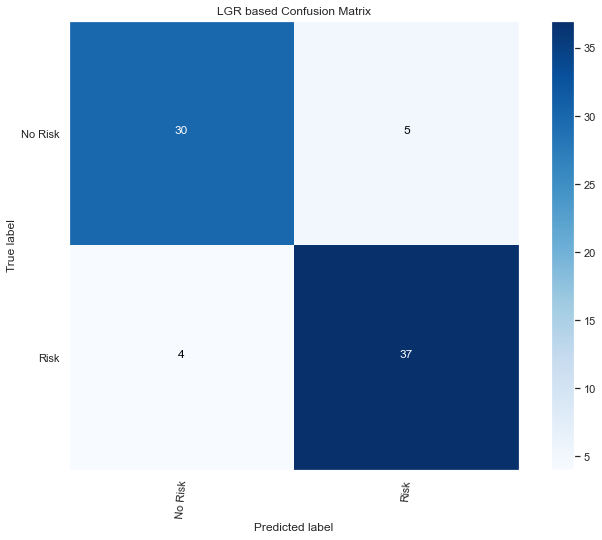
print("Model ROC\_AUC score:", roc\_auc\_score(y\_test, prediction\_lgr.astype(int)))

Model ROC\_AUC score: 0.8797909407665506

cm\_lgr = confusion\_matrix(y\_test, prediction\_lgr, labels=None)

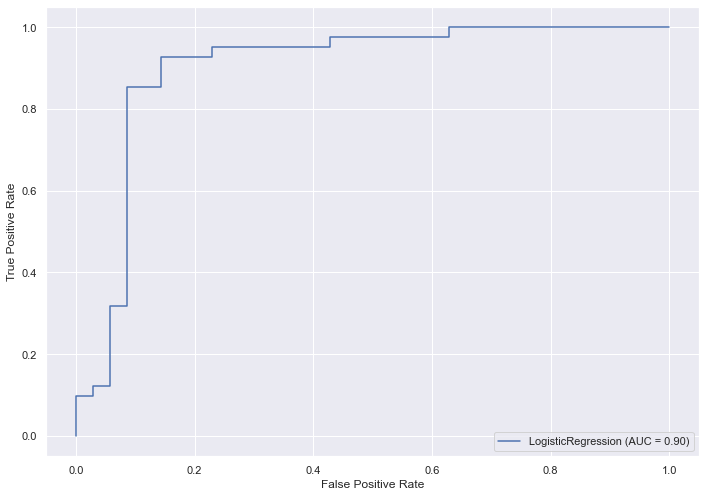
util.plot\_confusion\_matrix(cm\_lgr,classes=['No Risk', 'Risk'], title='LGR based Confusion Matrix')

Confusion Matrix  
confusion matrix:  
[[30 5]  
 [ 4 37]]



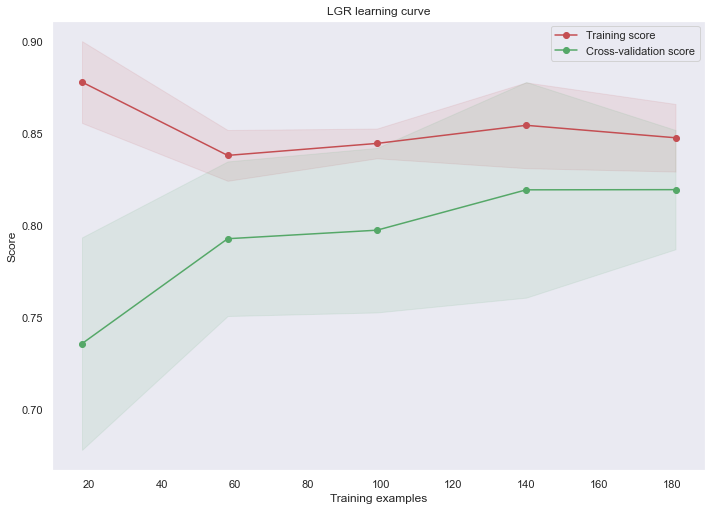
metrics.plot\_roc\_curve(Tuned\_model\_lgr, X\_test, y\_test)

<sklearn.metrics.\_plot.roc\_curve.RocCurveDisplay at 0x140053a60>



util.plot\_learning\_curve(estimator = Tuned\_model\_lgr, title = "LGR learning curve", X = X\_train, y = y\_train, cv = kfold)

<module 'matplotlib.pyplot' from '/Users/rrahman/Library/Python/3.8/lib/python/site-packages/matplotlib/pyplot.py'>



## *LightGBM based classification for heart risk*

## Hyperparameter optimization using RandomizedSearchCV  
import lightgbm as lgb

#Initialize Model  
clf\_mdl\_4 = lgb.LGBMClassifier();  
  
params = {'num\_leaves':range(10,100, 10), 'min\_child\_samples':range(5,25,5),'max\_depth': range(5, 15, 1),  
 'learning\_rate':[0.05,0.1,0.2],'reg\_alpha': [0,0.01,0.03]}  
  
#Initializing Grid Search with Stratified K Fold  
lgb\_ml = RandomizedSearchCV(clf\_mdl\_4, param\_distributions=params, n\_jobs=-1, cv=kfold)  
lgb\_ml.fit(X\_train,y\_train)

RandomizedSearchCV(cv=StratifiedKFold(n\_splits=5, random\_state=None, shuffle=False),  
 estimator=LGBMClassifier(), n\_jobs=-1,  
 param\_distributions={'learning\_rate': [0.05, 0.1, 0.2],  
 'max\_depth': range(5, 15),  
 'min\_child\_samples': range(5, 25, 5),  
 'num\_leaves': range(10, 100, 10),  
 'reg\_alpha': [0, 0.01, 0.03]})

lgb\_ml.best\_params\_

{'reg\_alpha': 0.01,  
 'num\_leaves': 10,  
 'min\_child\_samples': 15,  
 'max\_depth': 11,  
 'learning\_rate': 0.2}

Tuned\_model\_lgb = lgb\_ml.best\_estimator\_  
Tuned\_model\_lgb

LGBMClassifier(learning\_rate=0.2, max\_depth=11, min\_child\_samples=15,  
 num\_leaves=10, reg\_alpha=0.01)

from sklearn.model\_selection import cross\_val\_score  
score=cross\_val\_score(Tuned\_model\_lgb,X\_train,y\_train,cv=10)  
score

array([0.7826087 , 0.82608696, 0.69565217, 0.82608696, 0.7826087 ,  
 0.82608696, 0.7826087 , 0.95454545, 0.77272727, 0.68181818])

score.mean()

0.7930830039525691

prediction\_lgb = Tuned\_model\_lgb.predict(X\_test)

print("Model accuracy score:", accuracy\_score(y\_test, prediction\_lgb.astype(int)))

Model accuracy score: 0.8026315789473685

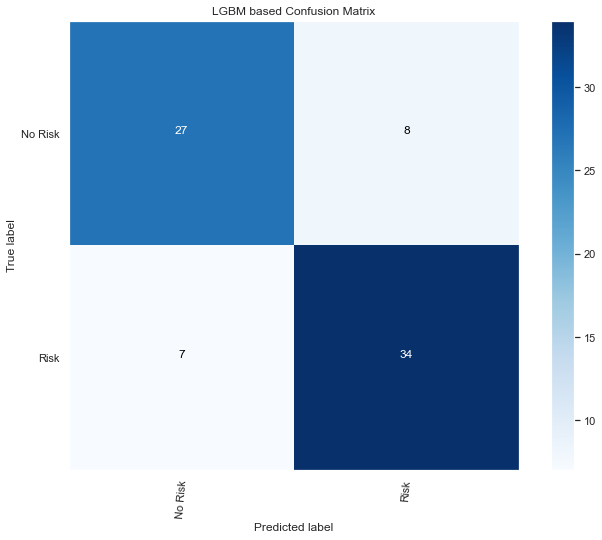
print("Model ROC\_AUC score:", roc\_auc\_score(y\_test, prediction\_lgb.astype(int)))

Model ROC\_AUC score: 0.8003484320557491

cm\_lgb = confusion\_matrix(y\_test, prediction\_lgb, labels=None)

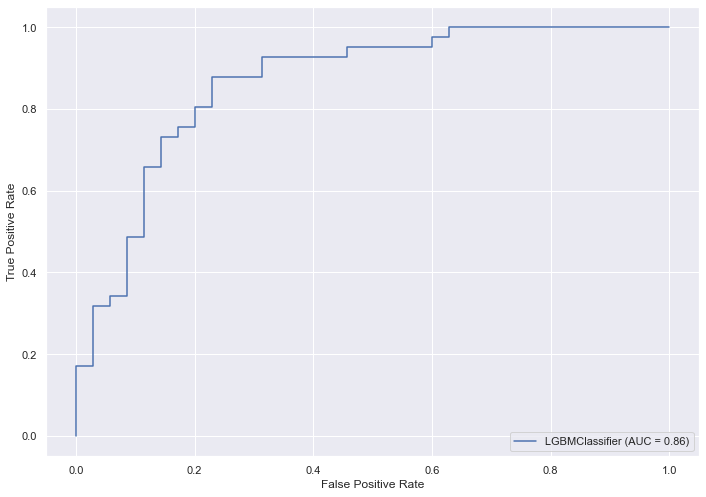
util.plot\_confusion\_matrix(cm\_lgb,classes=['No Risk', 'Risk'], title='LGBM based Confusion Matrix')

Confusion Matrix  
confusion matrix:  
[[27 8]  
 [ 7 34]]



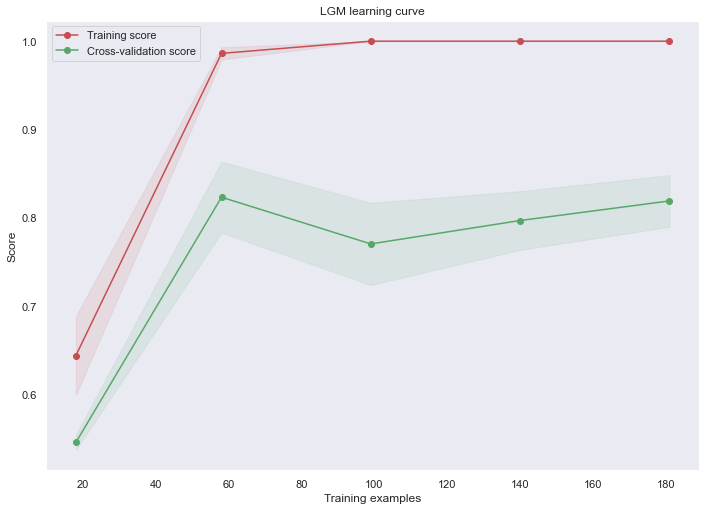
metrics.plot\_roc\_curve(Tuned\_model\_lgb, X\_test, y\_test)

<sklearn.metrics.\_plot.roc\_curve.RocCurveDisplay at 0x13e796c40>



util.plot\_learning\_curve(estimator = Tuned\_model\_lgb, title = "LGM learning curve", X = X\_train, y = y\_train, cv = kfold)

<module 'matplotlib.pyplot' from '/Users/rrahman/Library/Python/3.8/lib/python/site-packages/matplotlib/pyplot.py'>



## *Linear Discriminant Analysis based classification for heart risk*

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

#Initialize Model  
clf\_mdl\_5 = LinearDiscriminantAnalysis()  
  
params = {  
 "solver" : ["svd"],  
 "tol" : [0.0001,0.0002,0.0003]  
}  
  
lda\_ml = RandomizedSearchCV(clf\_mdl\_5, param\_distributions=params, n\_jobs=-1, cv=kfold)  
lda\_ml.fit(X\_train,y\_train)

RandomizedSearchCV(cv=StratifiedKFold(n\_splits=5, random\_state=None, shuffle=False),  
 estimator=LinearDiscriminantAnalysis(), n\_jobs=-1,  
 param\_distributions={'solver': ['svd'],  
 'tol': [0.0001, 0.0002, 0.0003]})

lda\_ml.best\_params\_

{'tol': 0.0001, 'solver': 'svd'}

Tuned\_model\_lda = lda\_ml.best\_estimator\_  
Tuned\_model\_lda

LinearDiscriminantAnalysis()

score=cross\_val\_score(Tuned\_model\_lda,X\_train,y\_train,cv=10)  
score

array([0.73913043, 0.7826087 , 0.86956522, 0.86956522, 0.7826087 ,  
 0.7826087 , 0.82608696, 0.90909091, 0.81818182, 0.77272727])

score.mean()

0.8152173913043479

prediction\_lda = Tuned\_model\_lda.predict(X\_test)

print("Model accuracy score:", accuracy\_score(y\_test, prediction\_lda.astype(int)))

Model accuracy score: 0.868421052631579

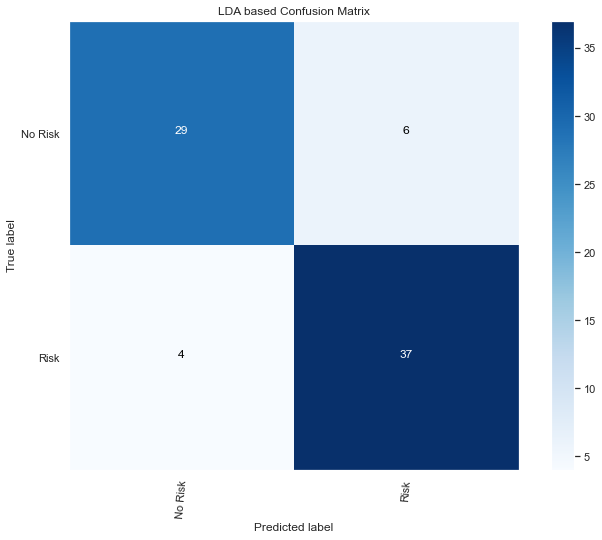
print("Model ROC\_AUC score:", roc\_auc\_score(y\_test, prediction\_lda.astype(int)))

Model ROC\_AUC score: 0.865505226480836

cm\_lda = confusion\_matrix(y\_test, prediction\_lda, labels=None)

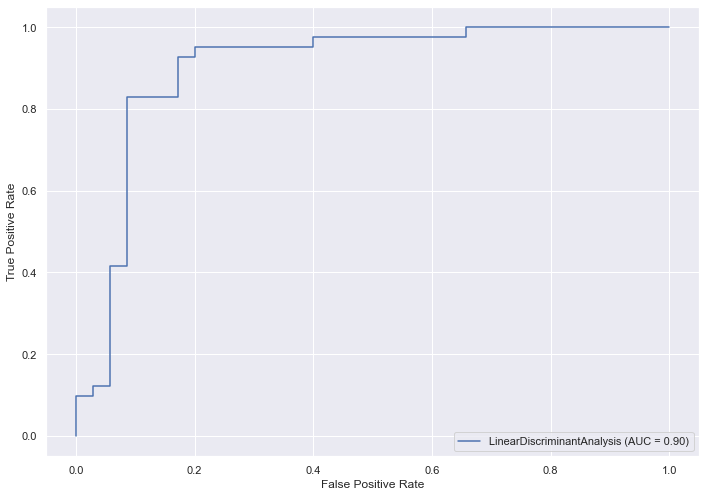
util.plot\_confusion\_matrix(cm\_lda,classes=['No Risk', 'Risk'], title='LDA based Confusion Matrix')

Confusion Matrix  
confusion matrix:  
[[29 6]  
 [ 4 37]]



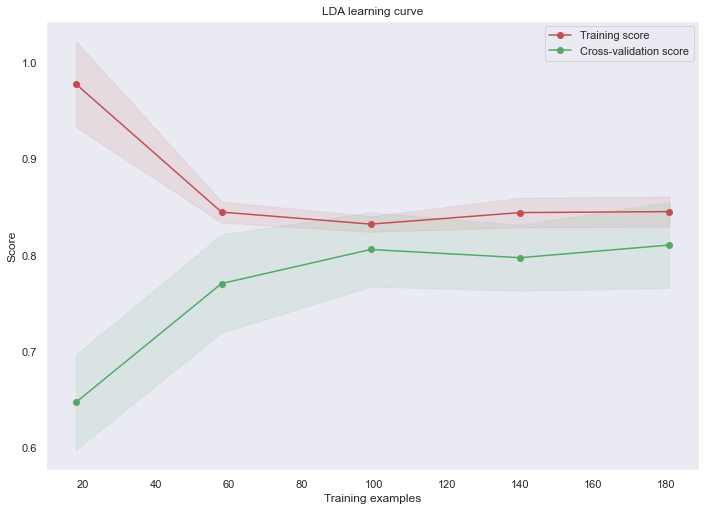
metrics.plot\_roc\_curve(Tuned\_model\_lda, X\_test, y\_test)

<sklearn.metrics.\_plot.roc\_curve.RocCurveDisplay at 0x13e097fa0>



util.plot\_learning\_curve(estimator = Tuned\_model\_lda, title = "LDA learning curve", X = X\_train, y = y\_train, cv = kfold)

<module 'matplotlib.pyplot' from '/Users/rrahman/Library/Python/3.8/lib/python/site-packages/matplotlib/pyplot.py'>



## *Ensemble learning based classification for heart risk*

from sklearn.ensemble import VotingClassifier

VotingPredictor = VotingClassifier(estimators =  
 [('rfc', Tuned\_model\_rf),   
 ('gbc', Tuned\_model\_xgb)],  
 voting='soft', n\_jobs = 4)  
  
  
VotingPredictor = VotingPredictor.fit(X\_train, y\_train)  
  
scores = cross\_val\_score(VotingPredictor, X\_train, y\_train, cv = kfold,  
 n\_jobs = -1, scoring = 'accuracy');  
  
print(scores)

/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):

[0.84782609 0.82608696 0.77777778 0.86666667 0.8 ]

prediction\_vp = VotingPredictor.predict(X\_test)

print("Model accuracy score:", accuracy\_score(y\_test, prediction\_vp.astype(int)))

Model accuracy score: 0.8552631578947368

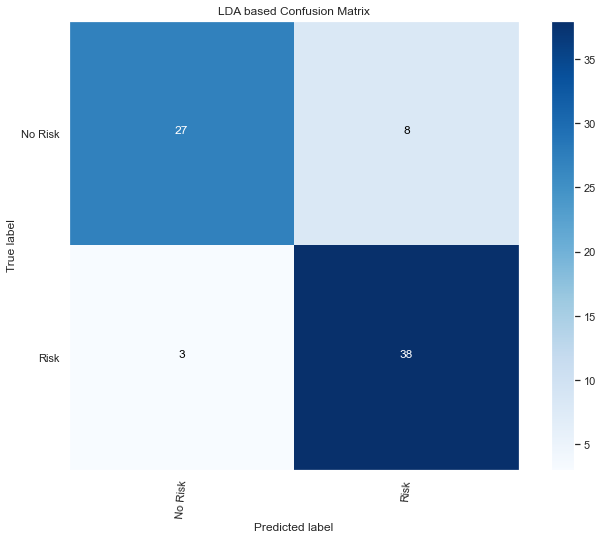
print("Model ROC\_AUC score:", roc\_auc\_score(y\_test, prediction\_vp.astype(int)))

Model ROC\_AUC score: 0.8491289198606271

cm\_vp = confusion\_matrix(y\_test, prediction\_vp, labels=None)

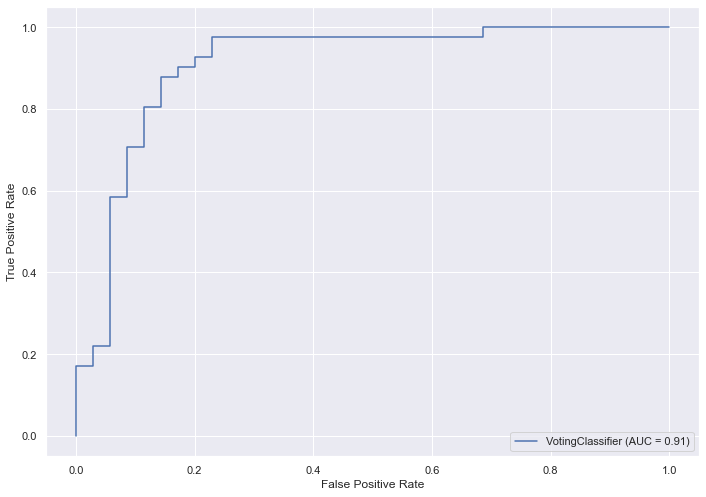
util.plot\_confusion\_matrix(cm\_vp,classes=['No Risk', 'Risk'], title='LDA based Confusion Matrix')

Confusion Matrix  
confusion matrix:  
[[27 8]  
 [ 3 38]]



metrics.plot\_roc\_curve(VotingPredictor, X\_test, y\_test)

<sklearn.metrics.\_plot.roc\_curve.RocCurveDisplay at 0x14070a070>



/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):

/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 from pandas import MultiIndex, Int64Index  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):  
/Users/rrahman/Library/Python/3.8/lib/python/site-packages/xgboost/data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.  
 elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):

util.plot\_learning\_curve(estimator = VotingPredictor, title = "VP learning curve",  
 X = X\_train, y = y\_train, cv = kfold);

