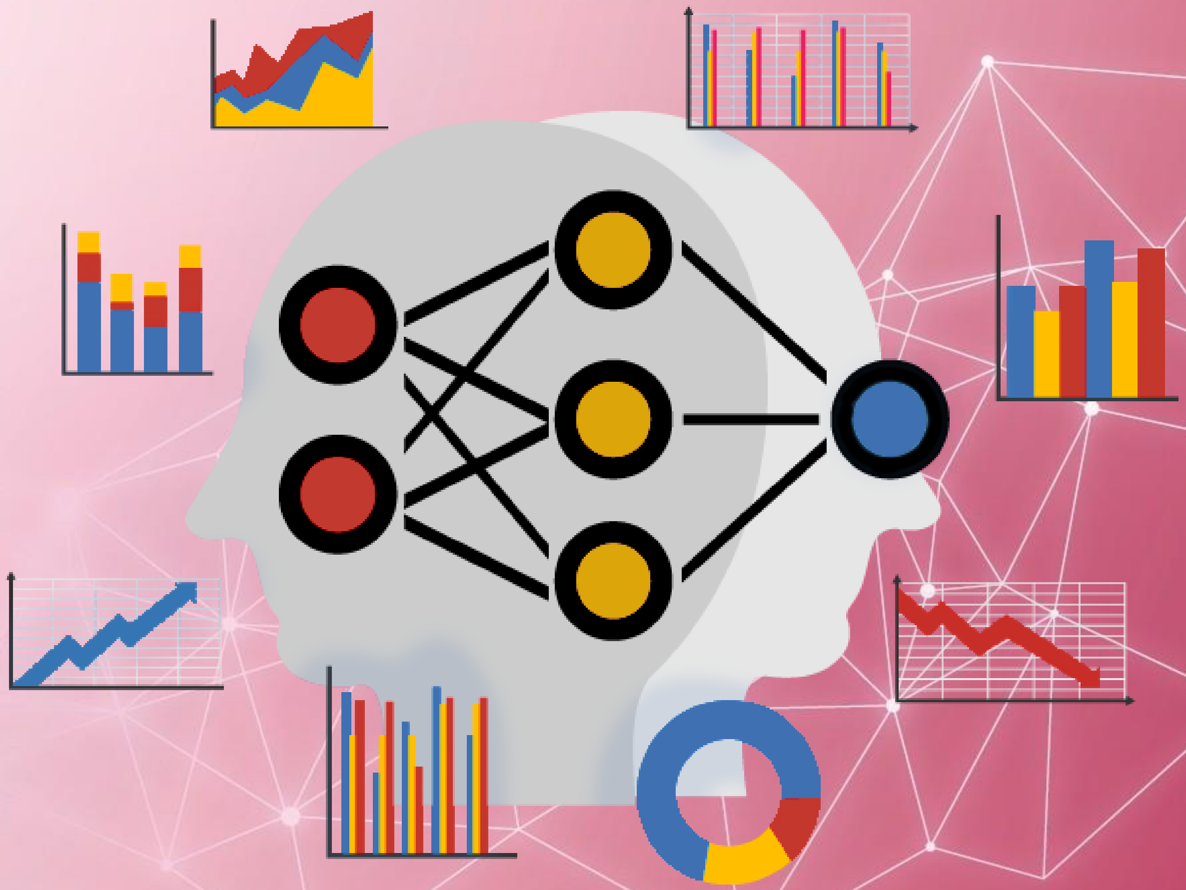
Knowledge Transfer Office
香港城市大學
City University of Hong Kong

5001 Group Project



Jiangping Lu (57719967)

Yuxin Miao (57393649)

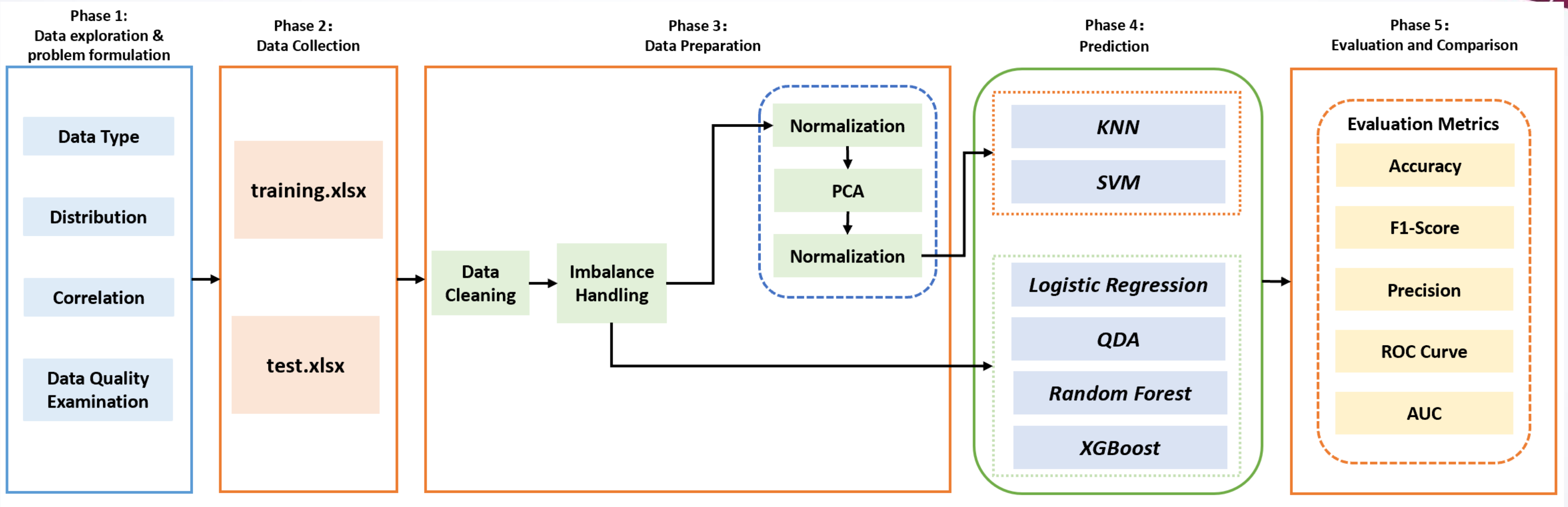
Ho Yin Tang (55695318)

Nov 25 2022

專業 創新 胸懷全球
Professional · Creative
For The World

Methodology & Architecture

- **Architecture of proposed model:**



Rationale:

- To understand the property of the data
- To formulate the problem
- To design data processing method and modeling method

Rationale:

- To allow cross-validation to train and evaluate perdition performance

Rationale:

- To ensure the quality of the data
- To convert the data to match model input format

Rationale:

- To remove the impact of scale in distance-based algorithms

Rationale:

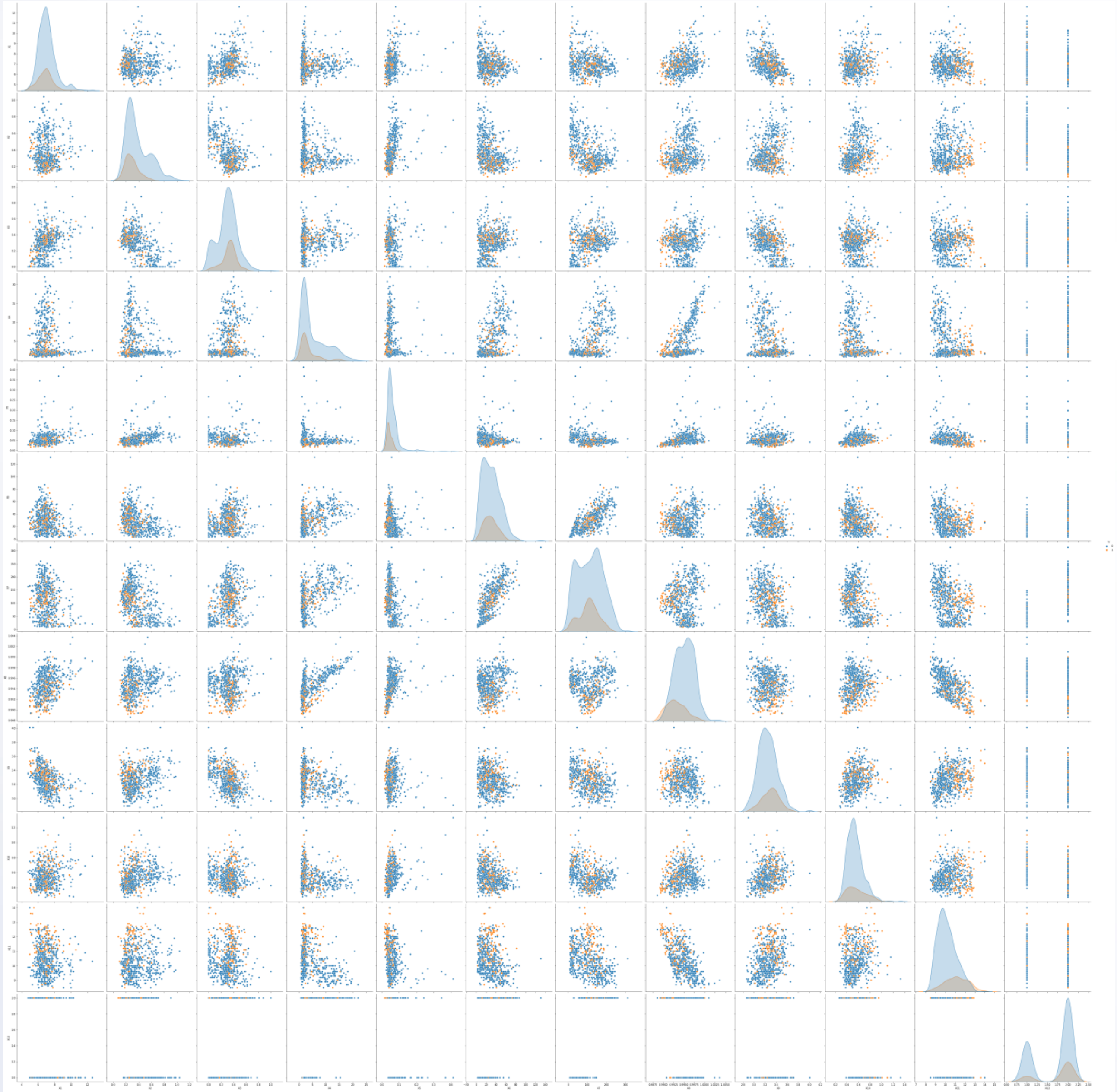
- To pursue optimal model for best predictability among a wide list of classification models

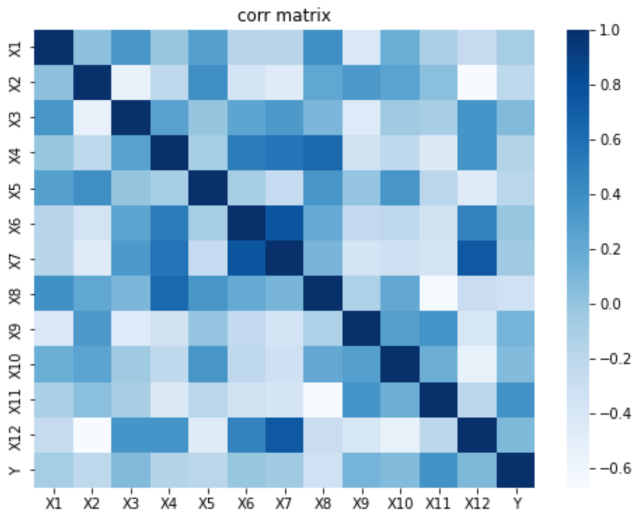
Rationale:

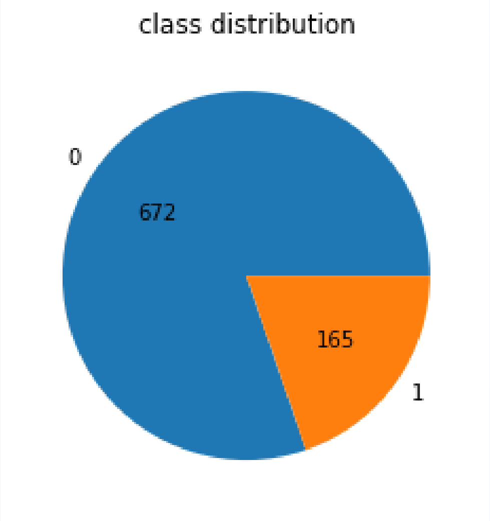
- To have an overall evaluation of prediction performance related to classification problem

Data exploration & Problem Formulation

Scatter plot between pairs of variables







Important Information Discovered

- The target classes distribution is not even
- Collinearity among some of the predictors
- No clear distinction between target classes in some predictors

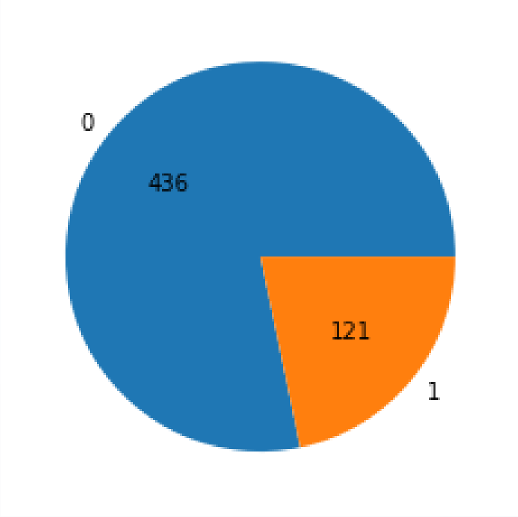
Problem Formulation

- Imbalance classification problem
- Distinction between target classes in some predictors need to be amplified
- It is expected that model that won't be significantly affected by collinearity will perform better

Data preparation

Data Cleaning

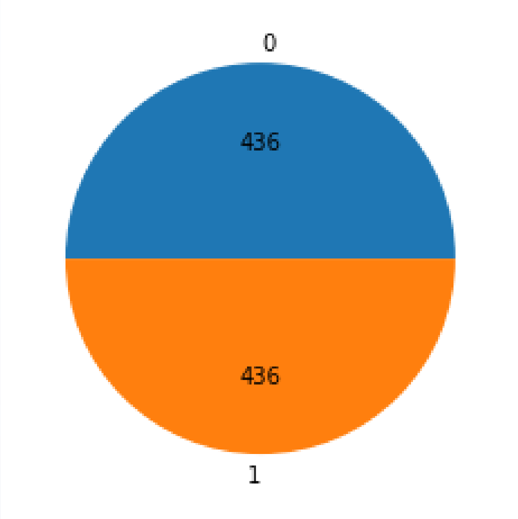
- Purpose:
  - Remove noise from the data to enhance model performance
- Methodology:
  - Outlier removal with condensed nearest neighbors undersampling technique
- Description:
  - This method can not only make the class distribution in the training set more balanced, but also remove outlier from both classes in the training set, we implemented a modification of the aforementioned technique called ***Tomek Links***, developed by Tomek (1976)



Class Distribution after Tomek Link

Imbalance Handling

- Purpose:
  - Balance the distribution of classes in training set such that the model fitted will demonstrate a higher ability to identity minority class accurately.
- Methodology:
  - Oversampling with synthetic minority over-sampling technique (SMOTE)
- Description:
  - We implemented a modification of the aforementioned technique called ***Borderline SMOTE***, developed by Han, H., Wang, WY., Mao, BH. (2005). It can identify noises and signals in minority class, and only oversample the signals.



Class Distribution after Borderline SMOTE

Data Preparation

- **Normalization:**
  - ***StandardScaler()***

$$Z = \frac{x - \mu}{\sigma}$$

where, μ is the mean of all sample data, and σ denotes the standard deviation.

```
from sklearn.preprocessing import StandardScaler
stdScaler = StandardScaler().fit(X_train) #return mean and sd
X_train = stdScaler.transform(X_train) #return sdscaler number
X_test = stdScaler.transform(X_test)
```

  - Normalize **before** PCA to ensure each feature is **in the same scale** to prevent over-capturing of certain features with large values, and accelerate convergence of gradient descent method.

- **PCA:**
  - Reduce Dimensionality.
  - Eliminate redundancy and data noise.
  - Identify the **most variable direction** of the observations.
  - Allow us to determine the property of density $f(X)$ when Y is unknown.

```
from sklearn.decomposition import PCA

pca=PCA(n_components=2,svd_solver='auto').fit(X_train)
# Dimensionality reduction
X_train_pca=pca.transform(X_train)
X_test_pca=pca.transform(X_test)
```

Model 1. Logistic Regression and QDA

Step 1:
Input features and the response to train

X1,X2,X3,X4,X5,X6,X7,X8, X9,X10,X11,X12 & Y

Step 2:
Model specification

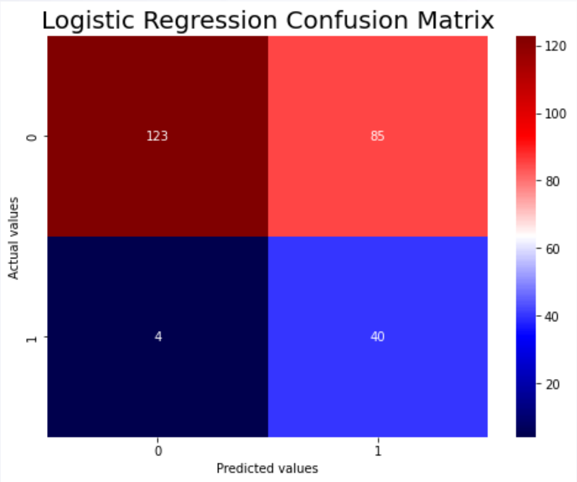
(1)Logistic Regression

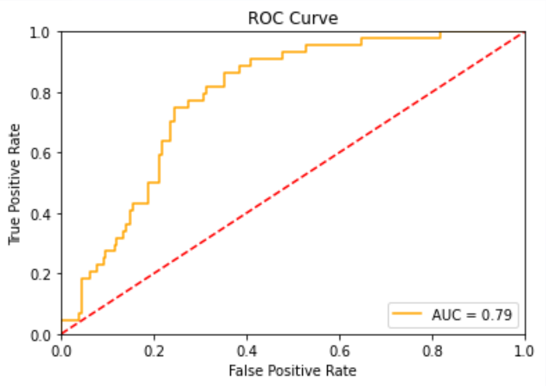
(2) QDA

Step 3:
Model evaluation

AUC=0.79

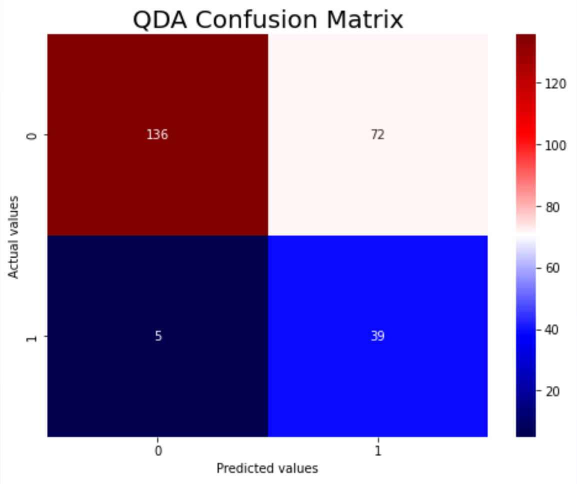
	precision	recall	f1-score	support
0	0.97	0.59	0.73	208
1	0.32	0.91	0.47	44
accuracy			0.65	252
macro avg	0.64	0.75	0.60	252
weighted avg	0.86	0.65	0.69	252

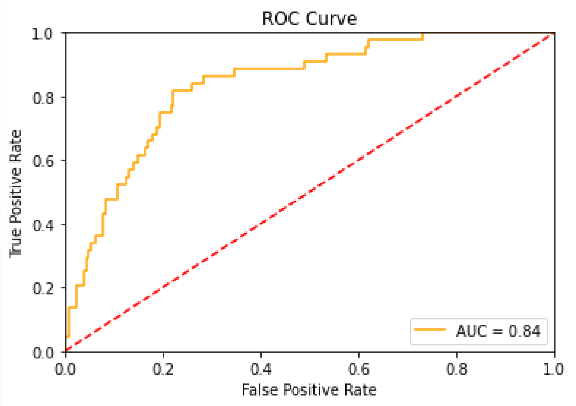




AUC=0.84

	precision	recall	f1-score	support
0	0.96	0.65	0.78	208
1	0.35	0.89	0.50	44
accuracy			0.69	252
macro avg	0.66	0.77	0.64	252
weighted avg	0.86	0.69	0.73	252

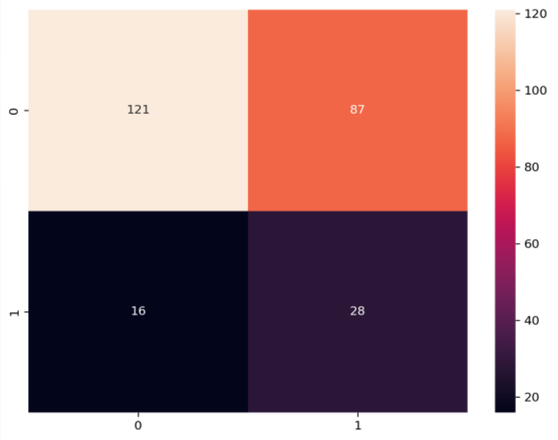




Model 3. SVM

- **A popular method used to solve the binary classification problem and implement the prediction of binary data, offering high accuracy.**
- **It can easily handle multiple continuous and categorical variables.**

Step 1:
Fit the model and obtain the confusion matrix and accuracy score.



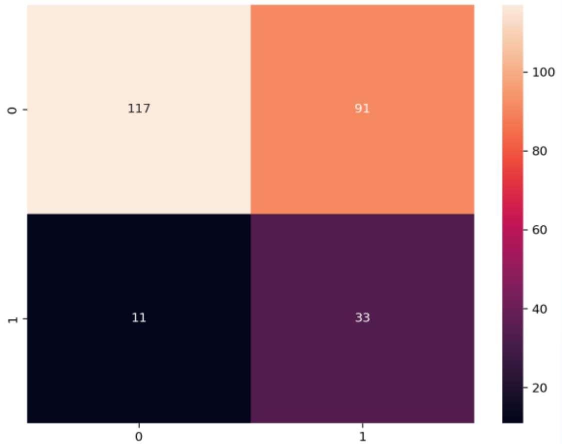

```
              precision    recall  f1-score   support

           0       0.88      0.58      0.70       208
           1       0.24      0.64      0.35        44

    accuracy                           0.59       252
   macro avg       0.56      0.61      0.53       252
weighted avg       0.77      0.59      0.64       252
```

***GridSearchCV* method**

Parameter Tuning

```
Best parameters: {'C': 1, 'gamma': 1, 'kernel': 'rbf'}
Best score: 0.8290598290598291
All results: {'mean_fit_time': array([4.20899549e+00, 1.00027084e-02, 2.54654074e+00, 1.11227036e-02,
        2.16706877e+00, 1.23162746e-02, 2.14732943e+00, 1.74256325e-02,
        2.15022101e+00, 2.09102154e-02, 2.15392747e+00, 1.86525822e-02,
        2.14112740e+00, 1.90710545e-02, 6.22949537e+01, 1.20897770e-02,
        6.18324012e+01, 1.57364368e-02, 6.11773126e+01, 1.67602062e-02,
        6.11166472e+01, 1.96372986e-02, 6.16399938e+01, 2.38285542e-02,
        6.16736504e+01, 2.04199314e-02, 6.11618265e+01, 2.00411320e-02,
        8.72206475e+01, 1.06702232e-01, 8.72351007e+01, 2.63195419e-01,
        8.75078353e+01, 5.13438225e-02, 8.75929256e+01, 2.25289822e-02,
        8.70332735e+01, 2.09994793e-02, 8.85261191e+01, 2.08218575e-02,
        8.80452137e+01, 2.31245995e-02, 1.88624667e+02, 1.49991293e+00,
        1.91024500e+02, 1.67630882e+00, 1.90941468e+02, 2.27840614e-01,
        1.89958094e+02, 2.29214191e-02, 1.90625849e+02, 2.04280853e-02,
        1.89651895e+02, 2.78691292e-02, 1.76210438e+02, 2.19381809e-02]), 'std_fit_time': array([1.18783863e+00, 4.79313799e-04,
        7.29313028e-01, 5.11265846e-04, 7.55608001e-01, 1.35818735e-03,
```

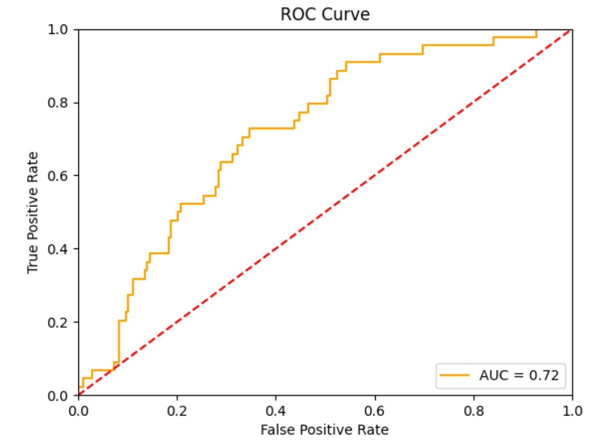
Step 2:
Fit the optimal model with the best parameter combination.




```
              precision    recall  f1-score   support

           0       0.91      0.56      0.70       208
           1       0.27      0.75      0.39        44

    accuracy                           0.60       252
   macro avg       0.59      0.66      0.54       252
weighted avg       0.80      0.60      0.64       252
```

Evaluation

Step 3:
Obtain **ROC/AUC**.



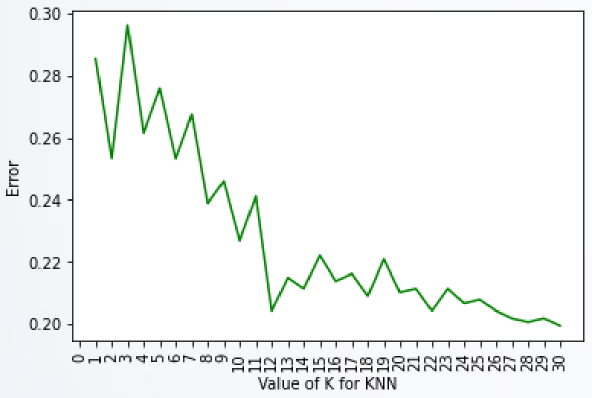
Results

Evaluation Metrics	Value
Accuracy (Training Set)	0.72
Accuracy (Test Set)	0.59
AUC	0.72

Model 4. KNN

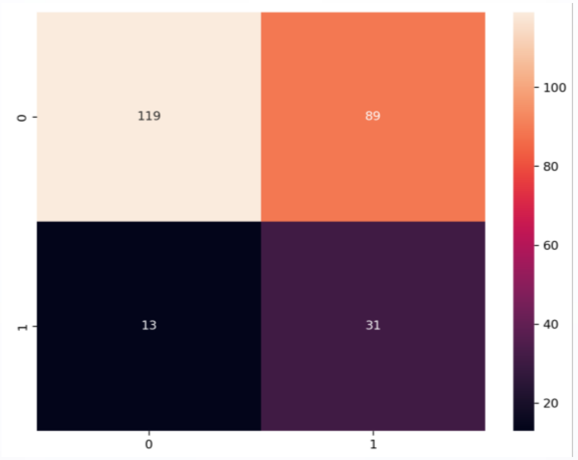
- **One of the simplest and most common algorithms.**
- **New data can be easily classified into a well-suited category.**

Step 1:
Select the optimal **K** by applying **Cross-validation** method.



Select **K** with the **minimum error, but K is not very large to shorten training time.**

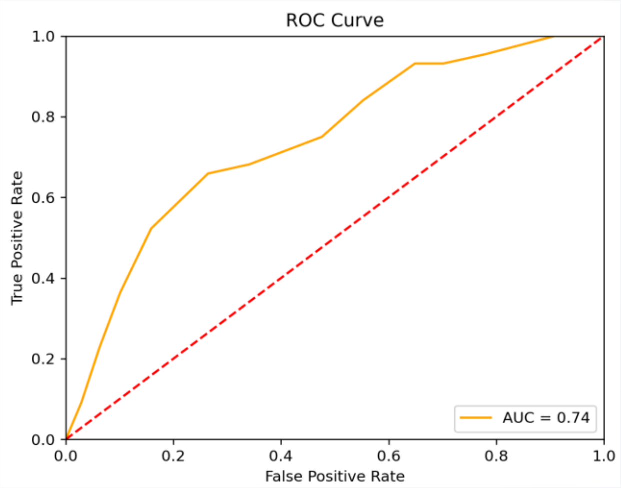
Step 2:
Let **K = 12** and fit the optimal model.
Obtain the accuracy and confusion matrix.



	precision	recall	f1-score	support
0	0.91	0.66	0.76	208
1	0.30	0.68	0.41	44
accuracy			0.66	252
macro avg	0.60	0.67	0.59	252
weighted avg	0.80	0.66	0.70	252

Evaluation

Step 3:
Obtain **ROC/AUC**.



Results

Evaluation Metrics	Value
Accuracy (Training Set)	0.75
Accuracy (Test Set)	0.67
AUC	0.74

Model 5. Random forest and XGBoost

Step 1) Input Processed train data (x,y), fit model, and obtain train accuracy

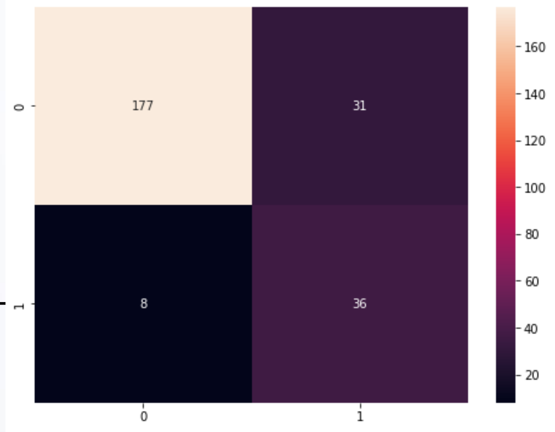
Step 2) Get prediction and prediction probability with fitted model

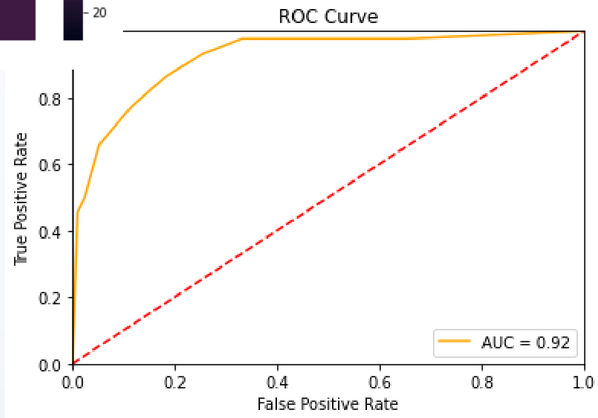
Step 3) Get model evaluation with confusion matrix, classification report, and ROC curve

Random Forest

Training Accuracy = 100%

	precision	recall	f1-score	support
0	0.96	0.85	0.90	208
1	0.54	0.82	0.65	44
accuracy			0.85	252
macro avg	0.75	0.83	0.77	252
weighted avg	0.88	0.85	0.86	252

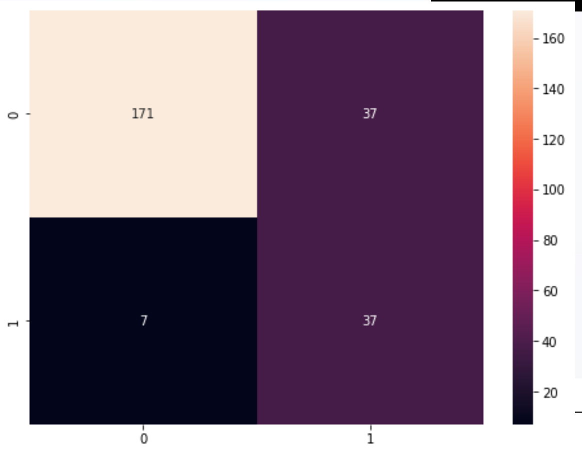


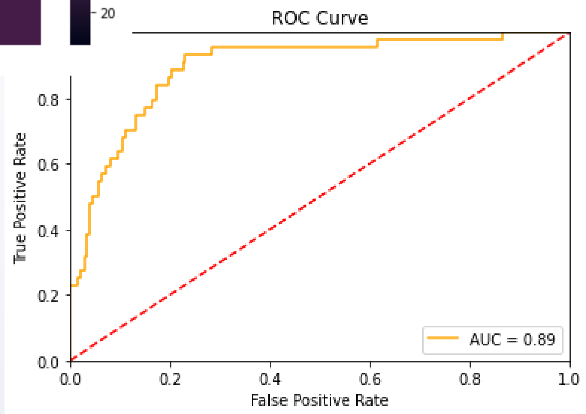


XGBoost

Training Accuracy = 100%

	precision	recall	f1-score	support
0	0.96	0.82	0.89	208
1	0.50	0.84	0.63	44
accuracy			0.83	252
macro avg	0.73	0.83	0.76	252
weighted avg	0.88	0.83	0.84	252





Model comparison by predictability performance

	Logistic Regression	QDA	Random Forest	XGBoost	SVM	KNN
Accuracy (train)	0.72	0.81	1	1	0.72	0.75
Accuracy (test)	0.65	0.69	0.85	0.83	0.59	0.67
F1 score (0)	0.73	0.78	0.90	0.89	0.69	0.77
F1 score (1)	0.47	0.50	0.65	0.63	0.39	0.42
AUC	0.79	0.84	0.92	0.89	0.72	0.74

ROC - AUC

- ➢ The area under the ROC curve.
- ➢ The **larger** the **better, 1** is the **ideal state.**
- ➢ We use class 1 as the true class when calculating ROC – AUC in our implementation

Metrics	Formula
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$
F1-Score	$\frac{2\times Precision\times Recall}{Precision+Recall}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$

Machine learing \ Manual counting	True	False
True	True Positive (TP)	False Positive (FP)
False	False Negative (FN)	True Negative (TN)

Rationale to use as core metric:

- Focus on evaluating the ability of identifying minority class from other class, an important attribute under the context of imbalance classification