00-credit-risk-model-without-feast

September 22, 2025

1 Bank Customer Credit Risk Model Development (Without FEAST)

1.1 Model

We develop a model that predicts if a credit loan customer can be a bad credit loan using the credit risk dataset.

1.2 Dataset

Use the trimmed Bank Customer Credit Risk dataset to have only:

Age (numeric) Sex (text: male, female) Job (numeric: 0 - unskilled and non-resident, 1 - unskilled and resident, 2 - skilled, 3 - highly skilled) Housing (text: own, rent, or free) Saving accounts (text - little, moderate, quite rich, rich) Checking account (numeric, in DM - Deutsch Mark) Credit amount (numeric, in DM) Duration (numeric, in month) Purpose(text: car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/others Risk (Value target - Good or Bad Risk)

See german-credit-data-with-risk to see the data in Kaggle.

2 Setup

---

3 Raw Data (Bank Customer Credit Risk)

```
<class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Age               1000 non-null   int64
 1   Sex               1000 non-null   object
 2   Job               1000 non-null   int64
 3   Housing           1000 non-null   object
 4   Saving accounts   817 non-null    object
 5   Checking account  606 non-null    object
 6   Credit amount     1000 non-null   int64
```

```
 7   Duration          1000 non-null   int64
 8   Purpose           1000 non-null   object
 9   Risk              1000 non-null   float64
dtypes: float64(1), int64(4), object(5)
memory usage: 85.9+ KB
None
```

```
   Age     Sex  Job Housing Saving accounts Checking account  Credit amount
Duration              Purpose  Risk
0   67    male    2     own             NaN           little           1169
6            radio/TV   0.0
1   22  female    2     own          little         moderate           5951
48             radio/TV   1.0
2   49    male    1     own          little              NaN           2096
12           education   0.0
3   45    male    2    free          little           little           7882
42  furniture/equipment   0.0
4   53    male    2    free          little           little           4870
24                  car   1.0
```

3.0.1 Column Types

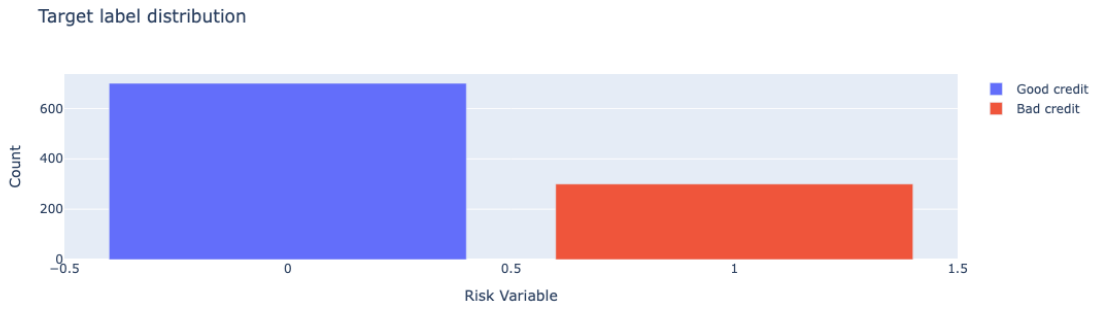
3.0.2 Unique Categorical Values

```
Sex              : [male, female]
Job              : [2, 1, 3, 0]
Housing          : [own, free, rent]
Saving accounts  : [little, quite rich, rich, moderate]
Checking account : [little, moderate, rich]
Purpose          : [radio/TV, education, furniture/equipment, car, business,
domestic appliances, repairs, vacation/others]
```

4 EDA

4.1 Target label (Risk) and imbalance

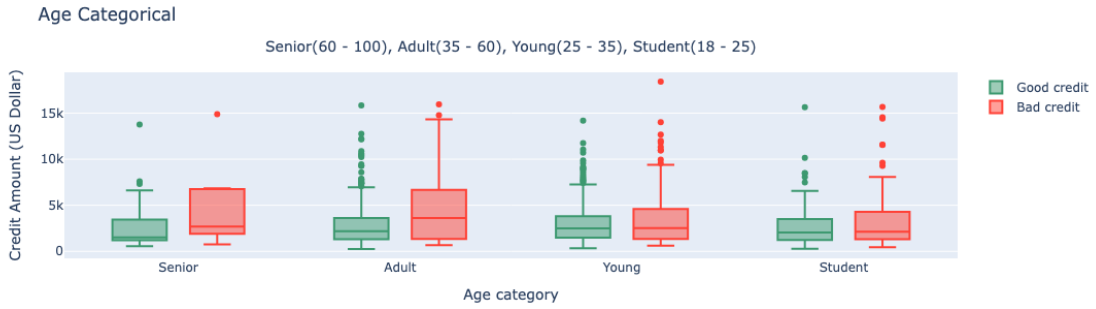
```
   Age   Sex  Job Housing Saving accounts Checking account  Credit amount
Duration   Purpose  Risk Generation Amount
0   67  male    2     own             NaN           little           1169
6  radio/TV   0.0     Senior    <5K
```



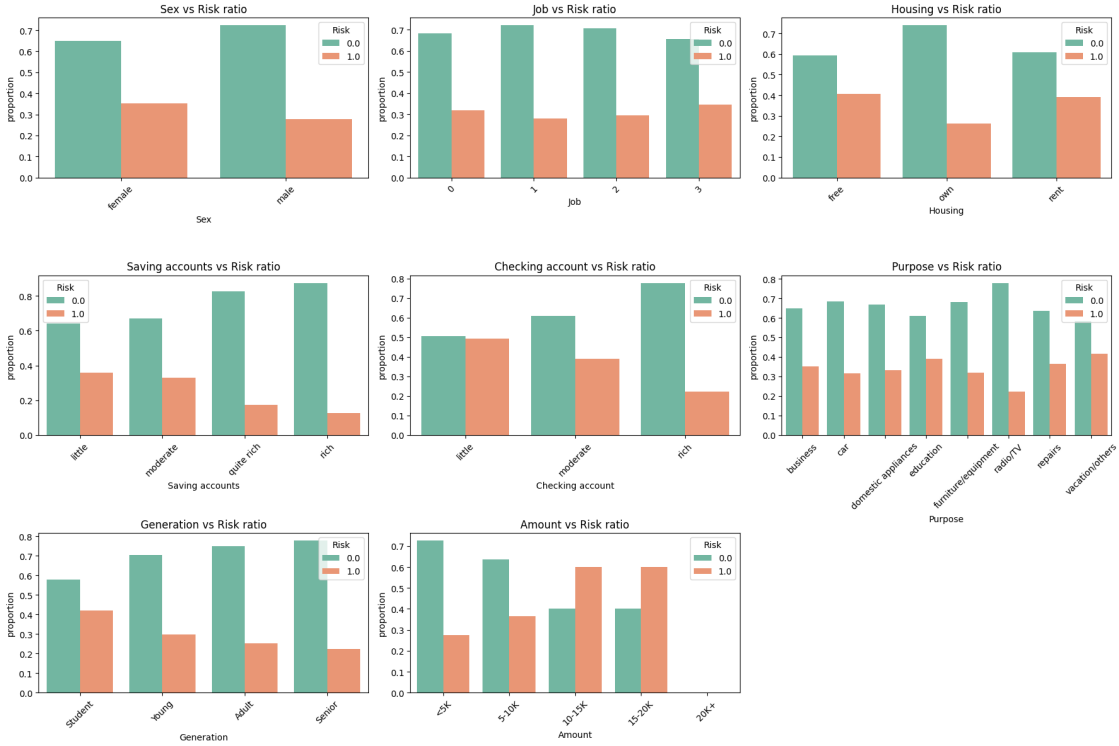
`Non Risky data is [2.33] more than Risky.`

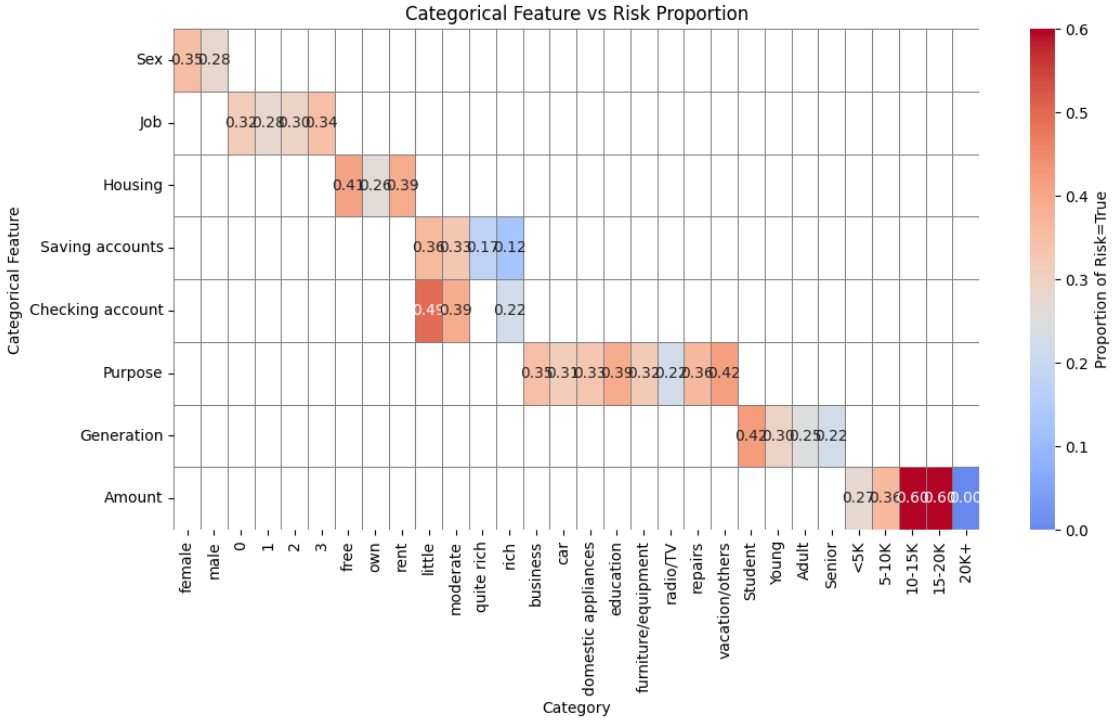
4.2 Risk by Credit Amount

Higher the amount, higher the risk.



5 Risk Correlation





6 Featur Engineering

Supposing the EDA has been done and concluded:

1. Not to use `Duration` column.
2. Convert all other columns to categorical.
3. One Hot Encode all the categorical columns.

7 Features for Model Consumption

Verify the features to be consumed by the Model Training

```
<class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, 0 to 999
Data columns (total 36 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   risk                          1000 non-null   float64
 1   gender_female                 1000 non-null   float32
 2   gender_male                   1000 non-null   float32
 3   job_0                         1000 non-null   float32
 4   job_1                         1000 non-null   float32
 5   job_2                         1000 non-null   float32
 6   job_3                         1000 non-null   float32
 7   housing_free                  1000 non-null   float32
 8   housing_own                   1000 non-null   float32
 9   housing_rent                  1000 non-null   float32
 10  saving_accounts_little        1000 non-null   float32
 11  saving_accounts_moderate      1000 non-null   float32
 12  saving_accounts_no_inf        1000 non-null   float32
 13  saving_accounts_quite_rich    1000 non-null   float32
 14  saving_accounts_rich          1000 non-null   float32
 15  checking_account_little       1000 non-null   float32
 16  checking_account_moderate     1000 non-null   float32
 17  checking_account_no_inf       1000 non-null   float32
 18  checking_account_rich         1000 non-null   float32
 19  purpose_business              1000 non-null   float32
 20  purpose_car                   1000 non-null   float32
 21  purpose_domestic_appliances   1000 non-null   float32
 22  purpose_education             1000 non-null   float32
 23  purpose_furniture_equipment   1000 non-null   float32
 24  purpose_radio_tv              1000 non-null   float32
 25  purpose_repairs               1000 non-null   float32
 26  purpose_vacation_others       1000 non-null   float32
```

```
 27  generation_student           1000 non-null   float32
 28  generation_young             1000 non-null   float32
 29  generation_adult             1000 non-null   float32
 30  generation_senior            1000 non-null   float32
 31  amount_0                     1000 non-null   float32
 32  amount_1                     1000 non-null   float32
 33  amount_2                     1000 non-null   float32
 34  amount_3                     1000 non-null   float32
 35  amount_4                     1000 non-null   float32
dtypes: float32(35), float64(1)
memory usage: 152.3 KB
```

```
   risk  gender_female  gender_male  job_0  job_1  job_2  job_3  housing_free
housing_own  housing_rent  saving_accounts_little  saving_accounts_moderate
saving_accounts_no_inf  saving_accounts_quite_rich  saving_accounts_rich
checking_account_little  checking_account_moderate  checking_account_no_inf
checking_account_rich  purpose_business  purpose_car
purpose_domestic_appliances  purpose_education  purpose_furniture_equipment
purpose_radio_tv  purpose_repairs  purpose_vacation_others  generation_student
generation_young  generation_adult  generation_senior  amount_0  amount_1
amount_2  amount_3  amount_4
0   0.0            0.0          1.0    0.0    0.0    1.0    0.0           0.0
1.0           0.0                      0.0                         0.0
1.0                          0.0                    0.0                       1.0
0.0                       0.0                    0.0               0.0
0.0                           0.0                0.0                           0.0
1.0              0.0                       0.0                 0.0
0.0                0.0                1.0       1.0       0.0       0.0       0.0
0.0
1   1.0            1.0          0.0    0.0    0.0    1.0    0.0           0.0
1.0           0.0                      1.0                         0.0
0.0                          0.0                    0.0                       0.0
1.0                       0.0                    0.0               0.0
0.0                           0.0                0.0                           0.0
1.0              0.0                       0.0                 1.0
0.0                0.0                0.0       0.0       1.0       0.0       0.0
0.0
2   0.0            0.0          1.0    0.0    1.0    0.0    0.0           0.0
1.0           0.0                      1.0                         0.0
0.0                          0.0                    0.0                       0.0
0.0                       1.0                    0.0               0.0
0.0                           0.0                1.0                           0.0
0.0              0.0                       0.0                 0.0
0.0                1.0                0.0       1.0       0.0       0.0       0.0
0.0
3   0.0            0.0          1.0    0.0    0.0    1.0    0.0           1.0
0.0           0.0                      1.0                         0.0
```

```
0.0                         0.0                   0.0                         1.0
0.0                       0.0                   0.0               0.0
0.0                           0.0               0.0                           1.0
0.0               0.0                       0.0                 0.0
0.0                1.0                0.0       0.0       1.0       0.0       0.0
0.0
4   1.0             0.0          1.0    0.0    0.0    1.0    0.0           1.0
0.0           0.0                       1.0                       0.0
0.0                           0.0                   0.0                         1.0
0.0                       0.0                   0.0               0.0
1.0                           0.0               0.0                           0.0
0.0               0.0                       0.0                 0.0
0.0                1.0                0.0       1.0       0.0       0.0       0.0
0.0
```

---

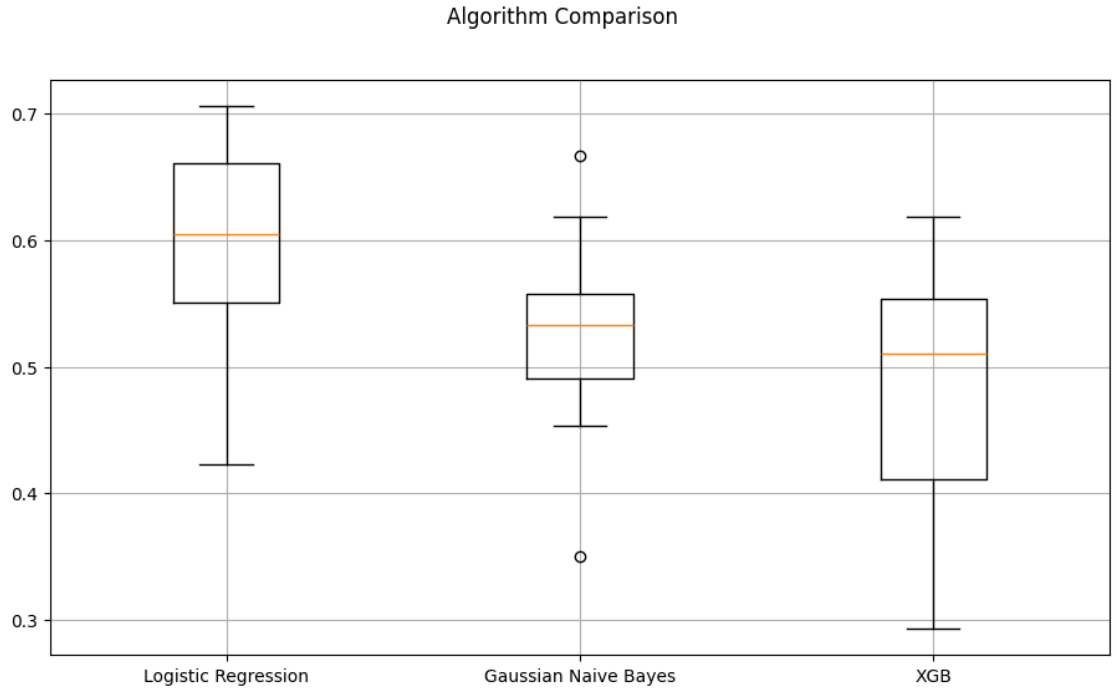
8 Model Training

```
   gender_female  gender_male  job_0  job_1  job_2  job_3  housing_free
housing_own  housing_rent  saving_accounts_little  saving_accounts_moderate
saving_accounts_no_inf  saving_accounts_quite_rich  saving_accounts_rich
checking_account_little  checking_account_moderate  checking_account_no_inf
checking_account_rich  purpose_business  purpose_car
purpose_domestic_appliances  purpose_education  purpose_furniture_equipment
purpose_radio_tv  purpose_repairs  purpose_vacation_others  generation_student
generation_young  generation_adult  generation_senior  amount_0  amount_1
amount_2  amount_3  amount_4
0            0.0          1.0    0.0    0.0    1.0    0.0           0.0
1.0           0.0                       0.0                       0.0
1.0                           0.0                   0.0                         1.0
0.0                       0.0                   0.0               0.0
0.0                           0.0               0.0                           0.0
1.0               0.0                       0.0                 0.0
0.0                0.0                1.0       1.0       0.0       0.0       0.0
0.0
1            1.0          0.0    0.0    0.0    1.0    0.0           0.0
1.0           0.0                       1.0                       0.0
0.0                           0.0                   0.0                         0.0
1.0                       0.0                   0.0               0.0
0.0                           0.0               0.0                           0.0
1.0               0.0                       0.0                 1.0
0.0                0.0                0.0       0.0       1.0       0.0       0.0
0.0
2            0.0          1.0    0.0    1.0    0.0    0.0           0.0
1.0           0.0                       1.0                       0.0
0.0                           0.0                   0.0                         0.0
```

```
0.0                    1.0                    0.0              0.0
0.0                        0.0                1.0                        0.0
0.0              0.0                    0.0                0.0
0.0               1.0               0.0       1.0       0.0       0.0       0.0
0.0
3           0.0          1.0    0.0    0.0    1.0    0.0          1.0
0.0          0.0                    1.0                    0.0
0.0                       0.0                   0.0                       1.0
0.0                    0.0                    0.0              0.0
0.0                        0.0                0.0                        1.0
0.0              0.0                    0.0                0.0
0.0               1.0               0.0       0.0       1.0       0.0       0.0
0.0
4           0.0          1.0    0.0    0.0    1.0    0.0          1.0
0.0          0.0                    1.0                    0.0
0.0                       0.0                   0.0                       1.0
0.0                    0.0                    0.0              0.0
1.0                        0.0                0.0                        0.0
0.0              0.0                    0.0                0.0
0.0               1.0               0.0       1.0       0.0       0.0       0.0
0.0
```

8.1 Algorithsm Comparision

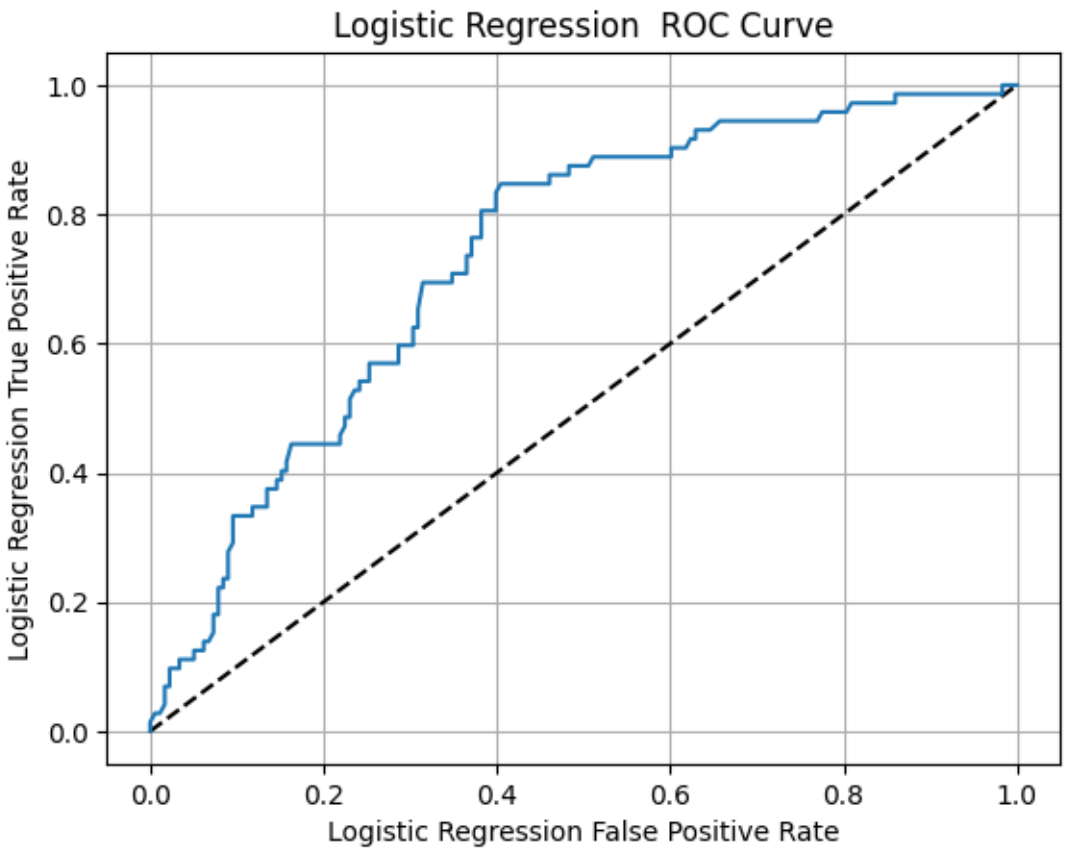
```
XGB: 0.485379 (0.095631)
Gaussian Naive Bayes: 0.525474 (0.082965)
Logistic Regression: 0.599749 (0.080013)
```



9 Train Models

9.1 Logistic Regression

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits

 GridSearchCV(cv=5,
              estimator=LogisticRegression(class_weight='balanced',
                                           max_iter=1000, random_state=42,
                                           solver='liblinear'),
              n_jobs=-1,
              param_grid={'C': [0.01, 0.1, 1, 10, 100],
                          'class_weight': ['balanced', None],
                          'penalty': ['l2']},
              scoring='f1', verbose=2)

Logistic regression best score: 0.5894372219310449
{
    "C": 0.01,
    "class_weight": "balanced",
    "penalty": "l2"
}
Best parameters: {'C': 0.01, 'class_weight': 'balanced', 'penalty': 'l2'}
Best ROC: 0.5894372219310449

Logistic Regression Results:
Confusion Matrix
[[115  63]
 [ 21  51]]
Accuracy: 0.6640
Recall: 0.7083
Precision: 0.4474
F1: 0.5484
AUC: 0.7365
classification report
              precision    recall  f1-score   support

         0.0       0.85      0.65      0.73       178
         1.0       0.45      0.71      0.55        72

    accuracy                           0.66       250
   macro avg       0.65      0.68      0.64       250
weighted avg       0.73      0.66      0.68       250
```



NOTE: SKLearn Confusion Matrix format:

```
[[TN FP]
 [FN TP]]
```

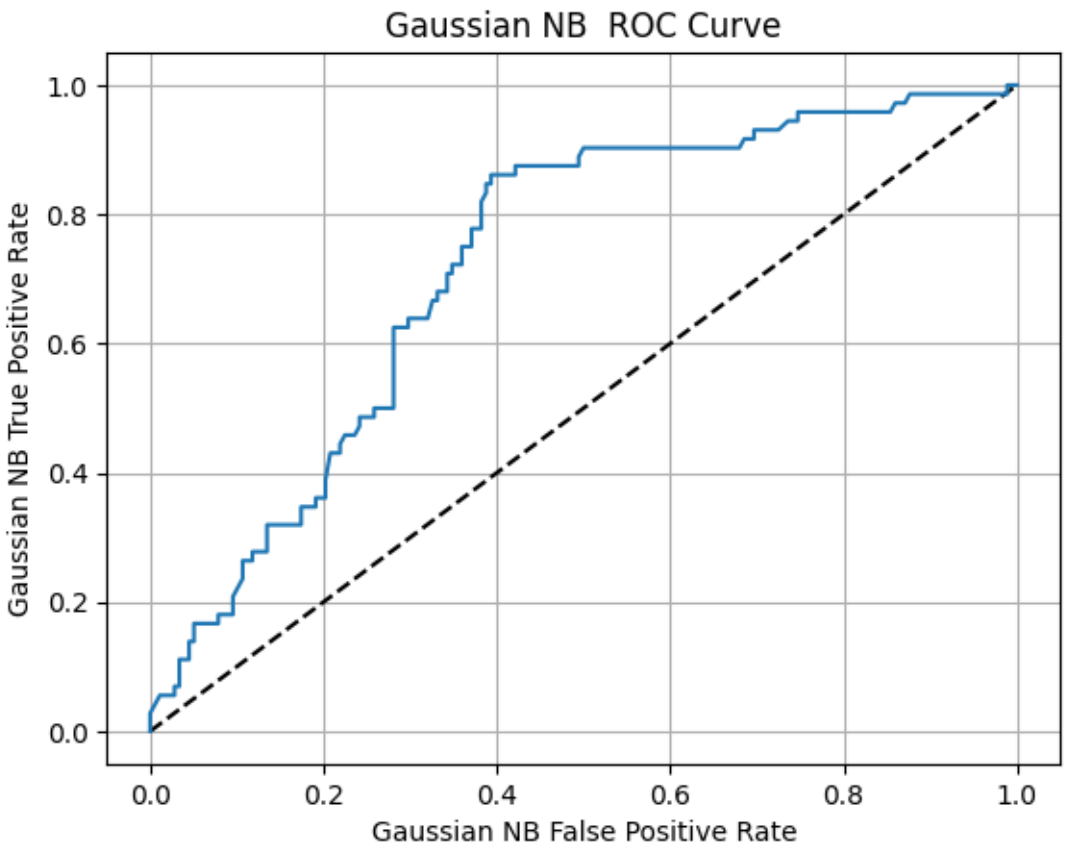
9.2 GNB

- Scikit-Learn GridSearchCV GaussianNB

```
0.6946666666666668
```

```
GridSearchCV(cv=KFold(n_splits=10, random_state=7, shuffle=True),
             estimator=Pipeline(steps=[('feature_union',
                                        FeatureUnion(transformer_list=[('pca',
PCA(n_components=2)),
('select_best',
SelectKBest(k=6))])),
                                       ('logistic', GaussianNB())]),
             n_jobs=-1,
             param_grid={'feature_union__pca__n_components': [1, 2, 3],
                         'feature_union__select_best__k': [4, 6, 8],
```

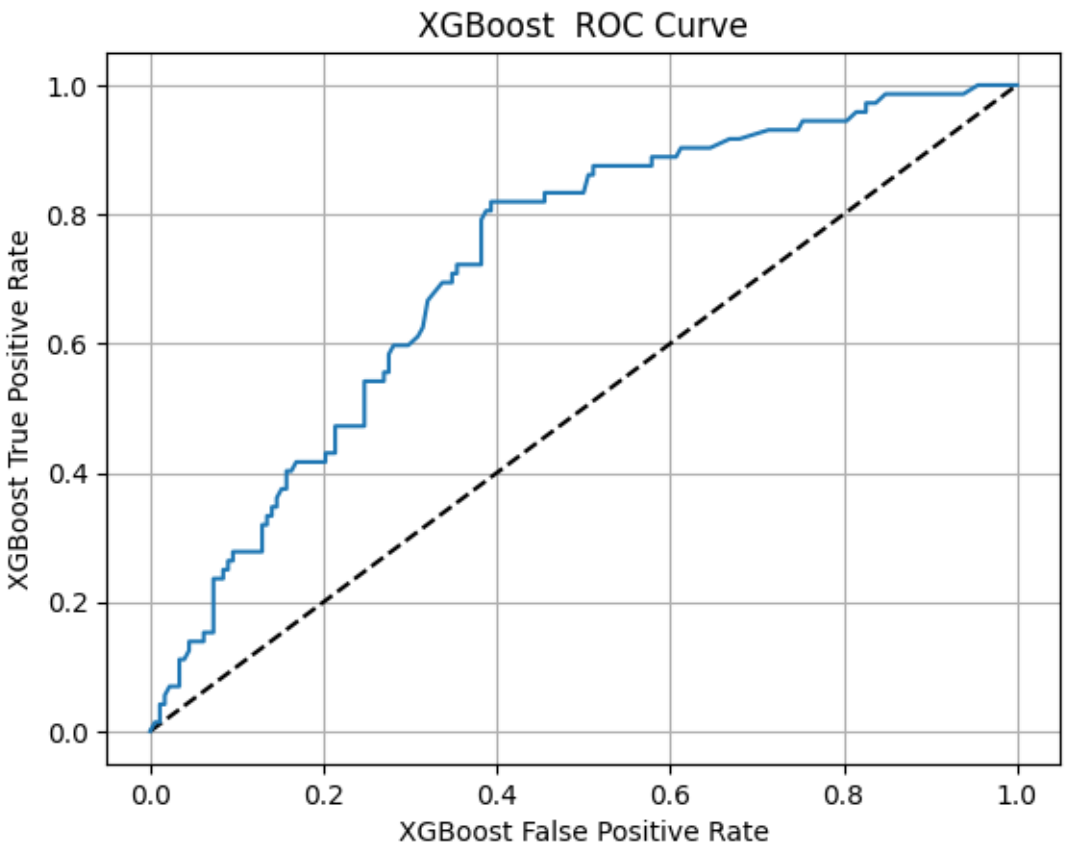
```
                        'logistic__var_smoothing': [1e-09, 1e-08, 1e-07,
                                                    1e-06]},
             scoring='f1')

GNB best score: 0.5516661003209277
{
    "feature_union__pca__n_components": 3,
    "feature_union__select_best__k": 4,
    "logistic__var_smoothing": 1e-09
}

Gaussian NB Results:
Confusion Matrix
[[128  50]
 [ 27  45]]
Accuracy: 0.6920
Recall: 0.6250
Precision: 0.4737
F1: 0.5389
AUC: 0.7239
classification report
              precision    recall  f1-score   support

         0.0       0.83      0.72      0.77       178
         1.0       0.47      0.62      0.54        72

    accuracy                           0.69       250
   macro avg       0.65      0.67      0.65       250
weighted avg       0.72      0.69      0.70       250
```



9.3 XGB

```
GridSearchCV(cv=5,
             estimator=XGBClassifier(base_score=None, booster=None,
                                     callbacks=None, colsample_bylevel=None,
                                     colsample_bynode=None,
                                     colsample_bytree=None, device=None,
                                     early_stopping_rounds=None,
                                     enable_categorical=False,
                                     eval_metric='auc', feature_types=None,
                                     feature_weights=None, gamma=None,
                                     grow_policy=None, importance_type=None,
                                     interaction_constraints=No…
                                     max_delta_step=None, max_depth=None,
                                     max_leaves=None, min_child_weight=None,
                                     missing=nan, monotone_constraints=None,
                                     multi_strategy=None, n_estimators=None,
                                     n_jobs=-1, num_parallel_tree=None, …),
             param_grid={'gamma': [0.1, 0.2],
```

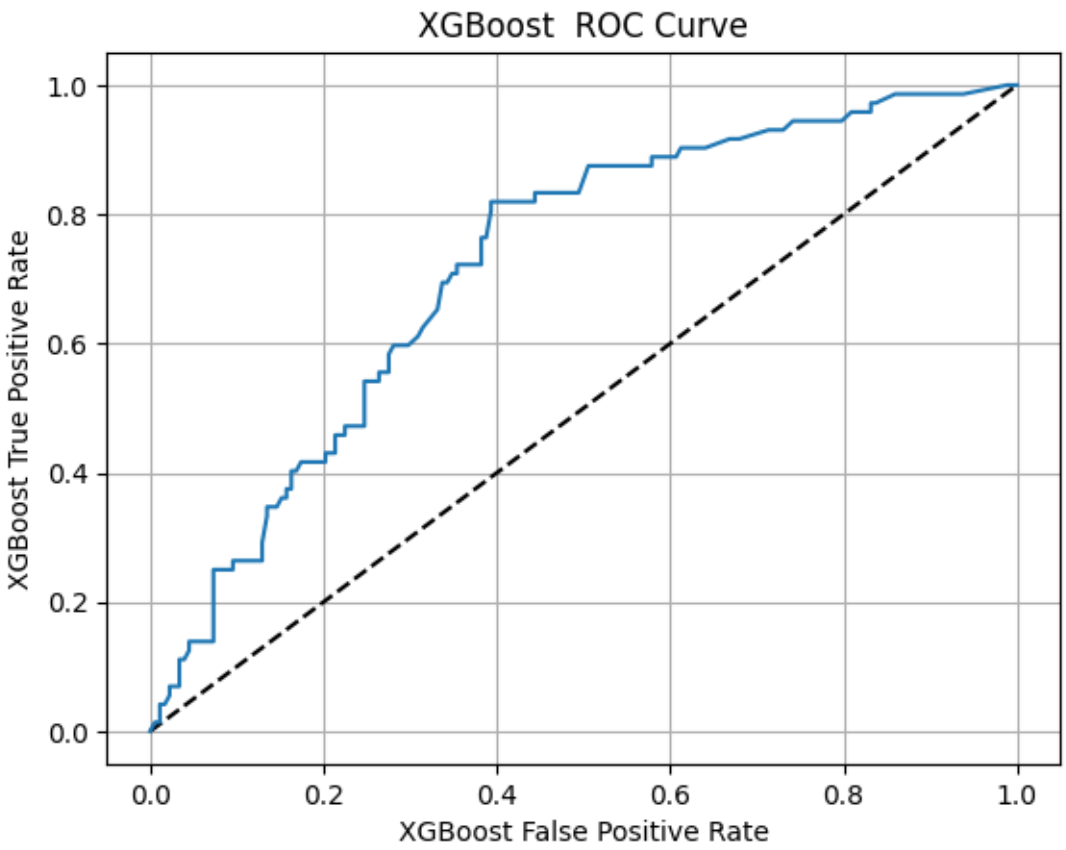
```
                          'learning_rate': [0.005, 0.01, 0.015],
                          'max_depth': [3, 4], 'min_child_weight': [8, 12],
                          'n_estimators': [100, 200, 300],
                          'scale_pos_weight': [2.3333333333333335]},
             scoring='f1')

stage1 grid_xb.best_score_:0.5782563728585046
stage1 best params:{
    "gamma": 0.1,
    "learning_rate": 0.01,
    "max_depth": 3,
    "min_child_weight": 12,
    "n_estimators": 200,
    "scale_pos_weight": 2.3333333333333335
}

XGBoost Results:
Confusion Matrix
[[110  68]
 [ 17  55]]
Accuracy: 0.6600
Recall: 0.7639
Precision: 0.4472
F1: 0.5641
AUC: 0.7231
classification report
              precision    recall  f1-score   support

         0.0       0.87      0.62      0.72       178
         1.0       0.45      0.76      0.56        72

    accuracy                           0.66       250
   macro avg       0.66      0.69      0.64       250
weighted avg       0.75      0.66      0.68       250
```



9.3.1 Stage2

```
GridSearchCV(cv=5,
             estimator=XGBClassifier(base_score=None, booster=None,
                                     callbacks=None, colsample_bylevel=None,
                                     colsample_bynode=None,
                                     colsample_bytree=None, device=None,
                                     early_stopping_rounds=None,
                                     enable_categorical=False,
                                     eval_metric='auc', feature_types=None,
                                     feature_weights=None, gamma=None,
                                     grow_policy=None, importance_type=None,
                                     interaction_constraints=No…
                                     max_leaves=None, min_child_weight=None,
                                     missing=nan, monotone_constraints=None,
                                     multi_strategy=None, n_estimators=None,
                                     n_jobs=-1, num_parallel_tree=None, …),
             param_grid={'gamma': [0, 0.1, 0.2], 'learning_rate': [0.01],
                         'max_depth': [3], 'min_child_weight': [12],
```

```
                        'n_estimators': [200], 'reg_alpha': [0, 0.1, 0.5],
                        'reg_lambda': [0.1, 1.0, 2.0],
                        'scale_pos_weight': [2.3333333333333335]},
             scoring='f1')

stage2 grid_xb.best_score_:0.5782563728585046
stage2 best params:{
    "gamma": 0,
    "learning_rate": 0.01,
    "max_depth": 3,
    "min_child_weight": 12,
    "n_estimators": 200,
    "reg_alpha": 0.1,
    "reg_lambda": 1.0,
    "scale_pos_weight": 2.3333333333333335
}

XGBoost Results:
Confusion Matrix
[[110  68]
 [ 17  55]]
Accuracy: 0.6600
Recall: 0.7639
Precision: 0.4472
F1: 0.5641
AUC: 0.7215
classification report
              precision    recall  f1-score   support

         0.0       0.87      0.62      0.72       178
         1.0       0.45      0.76      0.56        72

    accuracy                           0.66       250
   macro avg       0.66      0.69      0.64       250
weighted avg       0.75      0.66      0.68       250
```



	feature	importance
16	checking_account_no_inf	0.279368
17	checking_account_rich	0.063518
10	saving_accounts_moderate	0.057739
5	job_3	0.056073
14	checking_account_little	0.047422
26	generation_student	0.046623
11	saving_accounts_no_inf	0.045262
28	generation_adult	0.044990
7	housing_own	0.042318
30	amount_0	0.036817