03-credit-risk-model-feature-consumption

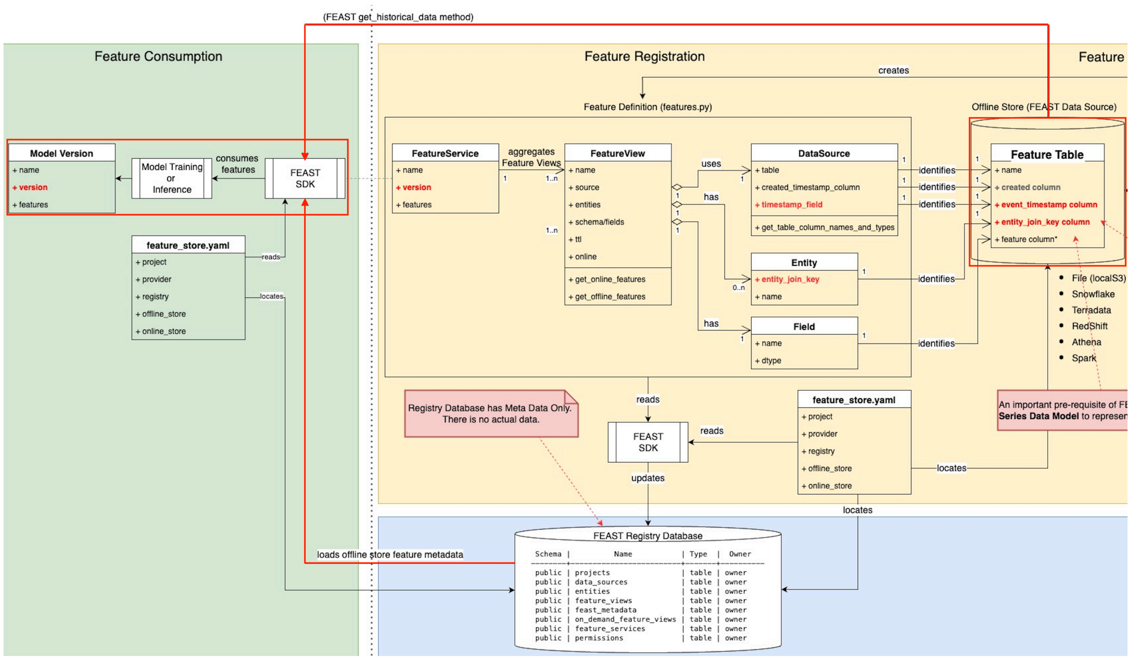
September 22, 2025

1 Feature Retrieval and Consumption for Model Training

The goal is to **Consume Features from the Offline Store** for model training.

1.1 This Notebook (03-feature-consumption)

This notebook goes through the steps of: 1. Load the Feature Metadata from the Feature Registry. 2. Load the Features from the Offline Store. 3. Train the model.



2 Setup

```
<IPython.core.display.HTML object>
```

---

3 FEAST Project

Recreate the FEAST project.

---

4 Metadata from Feature Registry

Load the project feature metadata from the feature registry.

4.0.1 Verify Retrieved Metadata


```
{
  "spec": {
    "name": "customer_credit_risk",
    "description": "A project for customer credit risk"
  },
  "meta": {
    "createdTimestamp": "2025-09-18T02:27:25.053090Z",
    "lastUpdatedTimestamp": "2025-09-18T02:27:25.053090Z"
  }
}
```


```
FeatureView: customer_credit_risk_feature_view
  Feature: risk (Float32)
  Feature: purpose_business (Float32)
  Feature: purpose_car (Float32)
  Feature: purpose_domestic_appliances (Float32)
  Feature: purpose_education (Float32)
  Feature: purpose_furniture_equipment (Float32)
  Feature: purpose_radio_tv (Float32)
  Feature: purpose_repairs (Float32)
  Feature: purpose_vacation_others (Float32)
  Feature: gender_female (Float32)
  Feature: gender_male (Float32)
  Feature: housing_free (Float32)
  Feature: housing_own (Float32)
  Feature: housing_rent (Float32)
  Feature: saving_accounts_little (Float32)
  Feature: saving_accounts_moderate (Float32)
  Feature: saving_accounts_no_inf (Float32)
  Feature: saving_accounts_quite_rich (Float32)
  Feature: saving_accounts_rich (Float32)
  Feature: checking_account_little (Float32)
  Feature: checking_account_moderate (Float32)
  Feature: checking_account_no_inf (Float32)
  Feature: checking_account_rich (Float32)
  Feature: generation_student (Float32)
  Feature: generation_young (Float32)
```

```
  Feature: generation_adult (Float32)
  Feature: generation_senior (Float32)
  Feature: job_0 (Float32)
  Feature: job_1 (Float32)
  Feature: job_2 (Float32)
  Feature: job_3 (Float32)
  Feature: amount_0 (Float32)
  Feature: amount_1 (Float32)
  Feature: amount_2 (Float32)
  Feature: amount_3 (Float32)
  Feature: amount_4 (Float32)

FeatureService: customer_credit_risk_feature_service
  FeatureProjection: customer_credit_risk_feature_view
    Feature: risk (Float32)
    Feature: purpose_business (Float32)
    Feature: purpose_car (Float32)
    Feature: purpose_domestic_appliances (Float32)
    Feature: purpose_education (Float32)
    Feature: purpose_furniture_equipment (Float32)
    Feature: purpose_radio_tv (Float32)
    Feature: purpose_repairs (Float32)
    Feature: purpose_vacation_others (Float32)
    Feature: gender_female (Float32)
    Feature: gender_male (Float32)
    Feature: housing_free (Float32)
    Feature: housing_own (Float32)
    Feature: housing_rent (Float32)
    Feature: saving_accounts_little (Float32)
    Feature: saving_accounts_moderate (Float32)
    Feature: saving_accounts_no_inf (Float32)
    Feature: saving_accounts_quite_rich (Float32)
    Feature: saving_accounts_rich (Float32)
    Feature: checking_account_little (Float32)
    Feature: checking_account_moderate (Float32)
    Feature: checking_account_no_inf (Float32)
    Feature: checking_account_rich (Float32)
    Feature: generation_student (Float32)
    Feature: generation_young (Float32)
    Feature: generation_adult (Float32)
    Feature: generation_senior (Float32)
    Feature: job_0 (Float32)
    Feature: job_1 (Float32)
    Feature: job_2 (Float32)
    Feature: job_3 (Float32)
    Feature: amount_0 (Float32)
    Feature: amount_1 (Float32)
    Feature: amount_2 (Float32)
```

```
Feature: amount_3 (Float32)
Feature: amount_4 (Float32)
```

---

5 Feature Retrieval from Offline Store

- get_historical_features

  This method joins historical feature data from one or more feature views to an entity dataframe by using a time travel join. Each feature view is joined to the entity dataframe using all entities configured for the respective feature view.

  Parameters
  * `entity_df`: a collection of rows containing all entity columns (e.g., driver_id) on which features need to be joined, as well as a event_timestamp column used to ensure point-in-time correctness.

  Returns: RetrievalJob which can be used to materialize the results.

- RetrievalJob

  A RetrievalJob manages the execution of a query to retrieve data from the offline store.
  Methods
  * to_df: Synchronously executes the underlying query and returns the result as a pandas dataframe. On demand transformations will be executed.

5.0.1 Use SQL as entity_df

- Example: entity SQL query for generating training data

```
    SELECT
        entity_id,
        event_timestamp
    FROM (SELECT * FROM credit.customer_credit_risk_offline_features) AS
ENTITY_SQL
```

- FEAST Feature Store - What is event_timestamp in entity_df parameter of FeatureStore.get_historical_features method

```
   entity_id             event_timestamp  risk  purpose_business  purpose_car
 purpose_domestic_appliances  purpose_education  purpose_furniture_equipment
 purpose_radio_tv  purpose_repairs  purpose_vacation_others  gender_female
 gender_male  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  generation_student
 generation_young  generation_adult  generation_senior  job_0  job_1  job_2
 job_3  amount_0  amount_1  amount_2  amount_3  amount_4
```

```
0         1 2025-09-18 12:27:23.123026   0.0               0.0          0.0
0.0               0.0                         0.0               1.0
0.0                     0.0           0.0         1.0           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               1.0     0.0     0.0     1.0     0.0        1.0
0.0       0.0       0.0       0.0
1         2 2025-09-18 12:27:23.123026   1.0               0.0          0.0
0.0               0.0                         0.0               1.0
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                   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
2         3 2025-09-18 12:27:23.123026   0.0               0.0          0.0
0.0               1.0                         0.0               0.0
0.0                     0.0           0.0         1.0           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               1.0               0.0     0.0     1.0     0.0     0.0        1.0
0.0       0.0       0.0       0.0
```

5.1 Online Store

Need to run the materialization command to get columns from the online store.

5.1.1 Materialise

```
Materializing 1 feature views from 2025-09-18
12:27:23+00:00 to 2025-09-18 12:27:45+00:00 into the
sqlite online store.

customer_credit_risk_feature_view:
```

5.1.2 Issue

FeatureStore.get_online_features() throws FeatureViewNotFoundException #5606

> FeatureViewNotFoundException: Feature view customer_credit_risk_customer_credit_risk_feature_view does not exist in project customer_credit_risk

```
   entity_id  gender_female  checking_account_moderate
 purpose_domestic_appliances  saving_accounts_moderate  amount_4  purpose_repairs
 amount_0  generation_senior  housing_rent  checking_account_little  risk
```

```
purpose_business  amount_3  generation_student  housing_own
saving_accounts_little  checking_account_rich  checking_account_no_inf
purpose_education  generation_adult  saving_accounts_quite_rich
generation_young  amount_1  housing_free  job_1  amount_2
purpose_vacation_others  purpose_radio_tv  gender_male  saving_accounts_no_inf
purpose_car  purpose_furniture_equipment  saving_accounts_rich  job_0  job_2
job_3
0          1            0.0                         0.0
0.0                     0.0       0.0              0.0       1.0
1.0          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          0.0    0.0       0.0                     0.0
1.0         1.0                    1.0         0.0
0.0                  0.0    0.0    1.0    0.0
1          2            1.0                         1.0
0.0                     0.0       0.0              0.0       0.0
0.0          0.0                    0.0   1.0              0.0       0.0
1.0         1.0                    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
```

6 Model Training - Feature Consumption

```
   purpose_business  purpose_car  purpose_domestic_appliances  purpose_education
purpose_furniture_equipment  purpose_radio_tv  purpose_repairs
purpose_vacation_others  gender_female  gender_male  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  generation_student  generation_young  generation_adult
generation_senior  job_0  job_1  job_2  job_3  amount_0  amount_1  amount_2
amount_3  amount_4
0              0.0          0.0                          0.0                0.0
0.0              1.0              0.0                        0.0            0.0
1.0           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                1.0    0.0    0.0    1.0
0.0       1.0       0.0       0.0       0.0       0.0
1              0.0          0.0                          0.0                0.0
0.0              1.0              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
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
2              0.0          0.0                          0.0              1.0
0.0              0.0             0.0                      0.0            0.0
1.0          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              1.0              0.0    0.0    1.0    0.0
0.0       1.0       0.0       0.0       0.0       0.0
```

6.1 Algorithsm Comparision

```
XGB: 0.486110 (0.104101)
Gaussian Naive Bayes: 0.525474 (0.082965)
Logistic Regression: 0.599749 (0.080013)
```

7 Train Models

7.1 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)
```

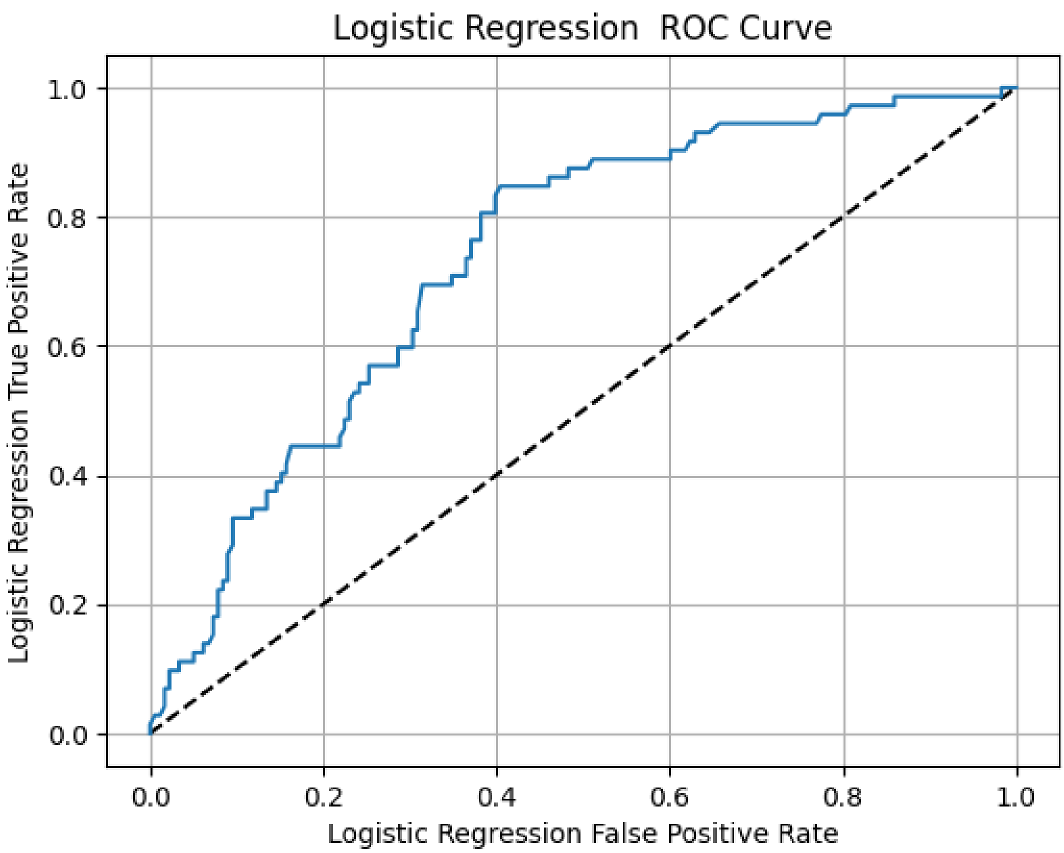
```
Logistic regression best score: 0.5894372219310449
{
    "C": 0.01,
    "class_weight": "balanced",
    "penalty": "l2"
}
Best parameters: {'C': 0.01, 'class_weight': 'balanced', 'penalty': 'l2'}
Best ROC: 0.5894372219310449

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]]
```

7.2 GNB

- Scikit-Learn GridSearchCV GaussianNB

```
0.6946666666666668
```

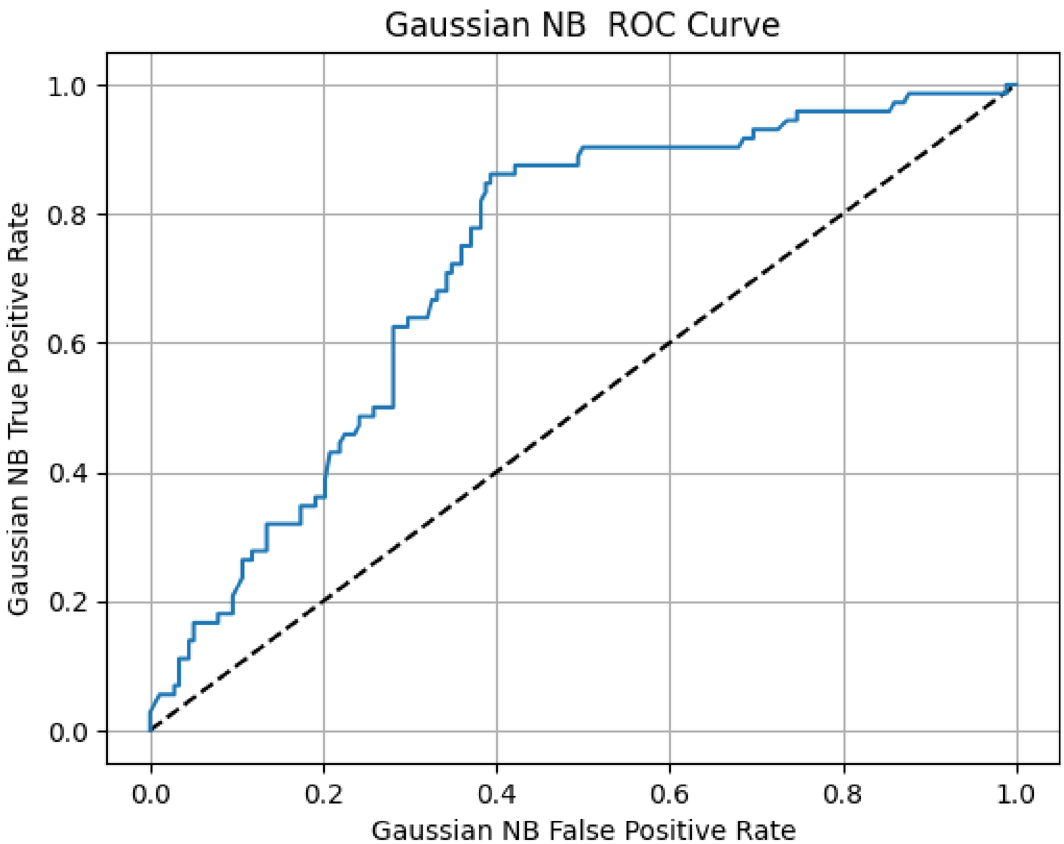
```
GridSearchCV(cv=KFold(n_splits=10, random_state=7, shuffle=True),
             estimator=Pipeline(steps=[('feature_union',
                                        FeatureUnion(transformer_list=[('pca',
PCA(n_components=2)),
('select_best',
SelectKBest(k=6))])),
                                       ('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
```



7.3 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],
```

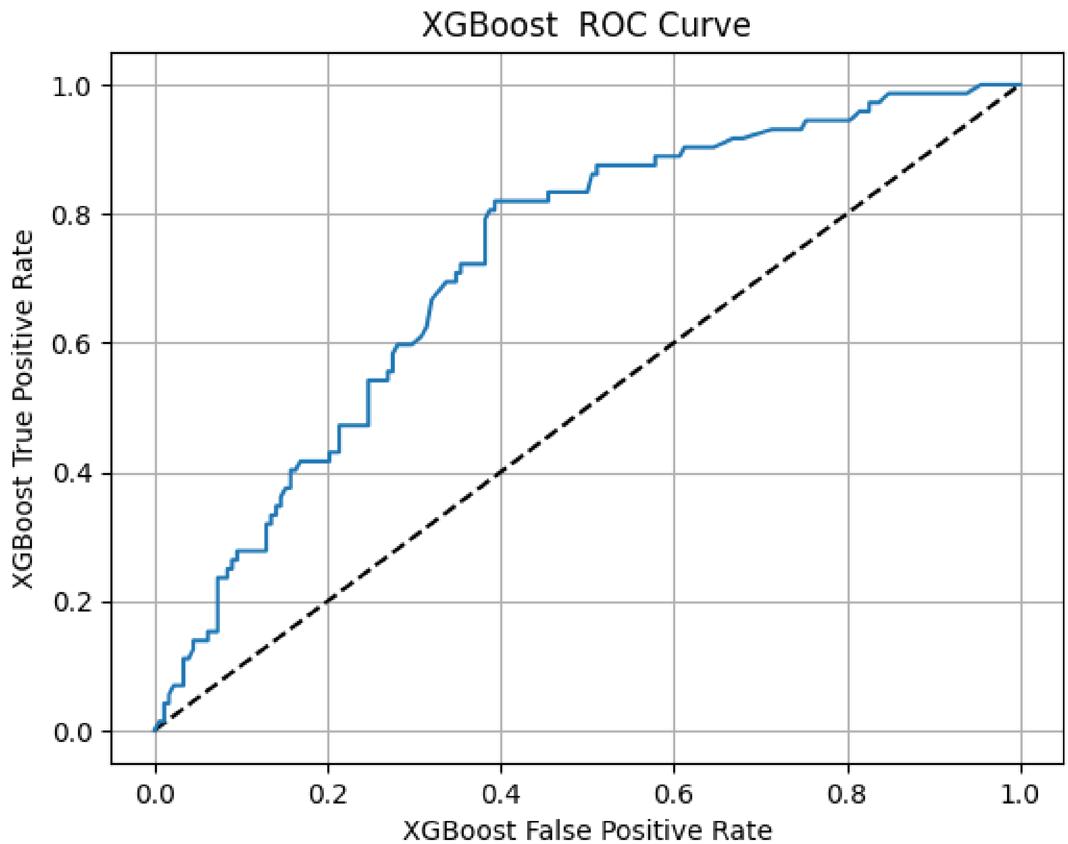
```
                         'learning_rate': [0.005, 0.01, 0.015],
                         'max_depth': [3, 4], 'min_child_weight': [8, 12],
                         'n_estimators': [100, 200, 300],
                         'scale_pos_weight': [2.3333333333333335]},
             scoring='f1')

grid_xb.best_score_: 0.5782563728585046
grid_xb.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
```



	feature	importance
20	checking_account_no_inf	0.283878
21	checking_account_rich	0.064194
14	saving_accounts_moderate	0.058702
29	job_3	0.058189
18	checking_account_little	0.048575
22	generation_student	0.046834
15	saving_accounts_no_inf	0.046445
24	generation_adult	0.044408
11	housing_own	0.043040
30	amount_0	0.038106