In [354…

```
'''
IST 692
Final Project
Daylin Hernandez
Title: Evaluating U.S. Mortgage Approval Prediction Models

Option 1.1: (Quantitative Analysis) Replicate and expand Project 2
•       Apply the methods from Project 2 to analyze another sensitive feature (e.g.
•       Integrate key concepts from Project 1, including reflecting on who is suppo
(and other pertinent questions from the data card or datasheets for dataset models)

     Introduction:
Gender disparity is recognized in several areas of our society, including the finan
individuals seeking loans. This project aims to explore gender as a factor in the m
Disclosure Act (HMDA) data from Rocket Mortgage. Using advanced data science tools
gender-based biases exist in loan approval decisions, develops predictive models to
fairness of these models.

    About the data:
The dataset utilized for the project is part of the Modified Loan Application Regis
This dataset is hosted and maintained by the Federal Financial Institutions Examina
Consumer Financial Protection Bureau (CFPB). The dataset is quite large with over 1

The impacted stakeholders include the banks and other institutions providing mortga
regulatory bodies to monitor fair lending practices and assess compliance with HMDA
to analyze the dataset for trends in housing finance and discrimination in lending
this data for insights into lending fairness.

     The overall structure of the project is divided into the following phases.
1. Data analysis and processing: to clean and prepare the dataset for analysis and
or incomplete data entries, transforming numerical variables, and encoding categori
2. Model Development: to build and train predictive models to classify loan applica
model was developed. The dataset was resampled to address class imbalance and the m
accuracy and ROC-AUC.
3. Fairness evaluation: assess the impact of gender on loan approval outcomes and e
using fairness metrics.
4. Explainability: Provide transparency into the factors driving model predictions
understand the role of gender in the model outcome.
5. Data Drift detection: to monitor changes in data distribution over time to ensur

'''
```

Out[354…

```
'\nIST 692\nFinal Project\nDaylin Hernandez\nTitle: Evaluating U.S. Mortgage Approval Prediction Models\n\nOption 1.1: (Quantitative Analysis) Replicate and expand Project 2\n•\tApply the methods from Project 2 to analyze another sensitive feature (e.g., applicant gender or ethnicity).\n•\tIntegrate key concepts from Project 1, including reflecting on who is supporting/hosting/maintaining the dataset \n(and other pertinent questions from the data card or datasheets for dataset models) and who the impacted stakeholders are.\n\n     Introduction:\nGender disparity is recognized in several areas of our society, including the financial industry, with a significant impact for\nindividuals seeking loans. This project aims to explore gender as a factor in the mortgage approval process using 2023 Home Mortgage\nDisclosure Act (HMDA) data from Rocket Mortgage. Using advanced data science tools and techniques, the project explores whether \ngender-based biases exist in loan approval decisions, develops predictive models to streamline the process, and evaluates the \nfairness of these models. \n\n    About the data:\nThe dataset utilized for the project is part of the Modified Loan Application Register (LAR) provided under the HMDA regulations. \nThis dataset is hosted and maintained by the Federal Financial Institutions Examination Council (FFIEC) in collaboration with the\nConsumer Financial Protection Bureau (CFPB). The dataset is quite large with over 100K entries and gets updated frequently. \n\nThe impacted stakeholders include the banks and other institutions providing mortgage loans that use this dataset for compliance and reporting, \nregulatory bodies to monitor fair lending practices and assess compliance with HMDA requirements, academics, policymakers, and data scientists \n to analyze the dataset for trends in housing finance and discrimination in lending practices. The public and civil rights organizations access\nthis data for insights into lending fairness.\n\n     The overall structure of the project is divided into the following phases. \n1. Data analysis and processing: to clean and prepare the dataset for analysis and modeling. This step included removing irrelevant\nor incomplete data entries, transforming numerical variables, and encoding categorical features for compatibility with machine learning models.\n2. Model Development: to build and train predictive models to classify loan applications as approved or denied. A baseline logistic regression \nmodel was developed. The dataset was resampled to address class imbalance and the model performance was evaluated using metrics like balanced \naccuracy and ROC-AUC.\n3. Fairness evaluation: assess the impact of gender on loan approval outcomes and ensure the model treats all demographic groups equitably \nusing fairness metrics.\n4. Explainability: Provide transparency into the factors driving model predictions by using SHAP, LIME, and visualizations that help us to \nunderstand the role of gender in the model outcome.\n5. Data Drift detection: to monitor changes in data distribution over time to ensure model robustness.\n\n'
```

In [356…

```python
#Load necessary packages for the analysis

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import (
    balanced_accuracy_score, roc_auc_score, confusion_matrix, RocCurveDisplay
)
from sklearn.impute import SimpleImputer
from fairlearn.metrics import (
    MetricFrame, selection_rate, false_negative_rate
```

```
)
from lime.lime_tabular import LimeTabularExplainer
import shap
from evidently import ColumnMapping
from evidently.report import Report
from evidently.metric_preset import DataDriftPreset

# Ignore warnings for clean output
import warnings
warnings.filterwarnings('ignore')
```

In [358…

```
#Load data
df = pd.read_csv("https://respai.s3.us-east-2.amazonaws.com/rocket+mortgage_pr.csv"
```

In [359…

```
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 109405 entries, 0 to 109404
Data columns (total 96 columns):
 #   Column                                    Non-Null Count   Dtype  
---  ------                                    --------------   -----  
 0   activity_year                             109405 non-null  int64  
 1   lei                                       109405 non-null  object 
 2   derived_msa-md                            109405 non-null  int64  
 3   state_code                                109405 non-null  object 
 4   county_code                               109396 non-null  float64
 5   census_tract                              109396 non-null  float64
 6   conforming_loan_limit                     109405 non-null  object 
 7   derived_loan_product_type                 109405 non-null  object 
 8   derived_dwelling_category                 109405 non-null  object 
 9   derived_ethnicity                         109405 non-null  object 
 10  derived_race                              109405 non-null  object 
 11  derived_sex                               109405 non-null  object 
 12  action_taken                              109405 non-null  int64  
 13  purchaser_type                            109405 non-null  int64  
 14  preapproval                               109405 non-null  int64  
 15  loan_type                                 109405 non-null  int64  
 16  loan_purpose                              109405 non-null  int64  
 17  lien_status                               109405 non-null  int64  
 18  reverse_mortgage                          109405 non-null  int64  
 19  open-end_line_of_credit                   109405 non-null  int64  
 20  business_or_commercial_purpose            109405 non-null  int64  
 21  loan_amount                               109405 non-null  int64  
 22  loan_to_value_ratio                       109405 non-null  float64
 23  interest_rate                             94469 non-null   float64
 24  rate_spread                               94274 non-null   float64
 25  hoepa_status                              109405 non-null  int64  
 26  total_loan_costs                          93444 non-null   float64
 27  origination_charges                       93444 non-null   float64
 28  discount_points                           66109 non-null   float64
 29  lender_credits                            2960 non-null    float64
 30  loan_term                                 109405 non-null  int64  
 31  intro_rate_period                         640 non-null     float64
 32  negative_amortization                     109405 non-null  int64  
 33  interest_only_payment                     109405 non-null  int64  
 34  balloon_payment                           109405 non-null  int64  
 35  other_nonamortizing_features              109405 non-null  int64  
 36  property_value                            109405 non-null  int64  
 37  construction_method                       109405 non-null  int64  
 38  occupancy_type                            109405 non-null  int64  
 39  manufactured_home_secured_property_type   109405 non-null  int64  
 40  manufactured_home_land_property_interest  109405 non-null  int64  
 41  total_units                               109405 non-null  int64  
 42  income                                    108649 non-null  float64
 43  debt_to_income_ratio                      109405 non-null  object 
 44  applicant_credit_score_type               109405 non-null  int64  
 45  co-applicant_credit_score_type            109405 non-null  int64  
 46  applicant_ethnicity-1                     109405 non-null  int64  
 47  applicant_ethnicity-2                     13417 non-null   float64
 48  applicant_ethnicity-3                     190 non-null     float64
 49  applicant_ethnicity-4                     4 non-null       float64
 50  applicant_ethnicity-5                     1 non-null       float64
```

```
 51  co-applicant_ethnicity-1                109405 non-null  int64
 52  co-applicant_ethnicity-2                5769 non-null    float64
 53  co-applicant_ethnicity-3                71 non-null      float64
 54  co-applicant_ethnicity-4                1 non-null       float64
 55  co-applicant_ethnicity-5                0 non-null       float64
 56  applicant_ethnicity_observed            109405 non-null  int64
 57  co-applicant_ethnicity_observed         109405 non-null  int64
 58  applicant_race-1                        109405 non-null  int64
 59  applicant_race-2                        14300 non-null   float64
 60  applicant_race-3                        793 non-null     float64
 61  applicant_race-4                        88 non-null      float64
 62  applicant_race-5                        33 non-null      float64
 63  co-applicant_race-1                     109405 non-null  int64
 64  co-applicant_race-2                     5819 non-null    float64
 65  co-applicant_race-3                     285 non-null     float64
 66  co-applicant_race-4                     28 non-null      float64
 67  co-applicant_race-5                     12 non-null      float64
 68  applicant_race_observed                 109405 non-null  int64
 69  co-applicant_race_observed              109405 non-null  int64
 70  applicant_sex                           109405 non-null  int64
 71  co-applicant_sex                        109405 non-null  int64
 72  applicant_sex_observed                  109405 non-null  int64
 73  co-applicant_sex_observed               109405 non-null  int64
 74  applicant_age                           109405 non-null  object
 75  co-applicant_age                        109405 non-null  object
 76  applicant_age_above_62                  109405 non-null  object
 77  co-applicant_age_above_62               45774 non-null   object
 78  submission_of_application               109405 non-null  int64
 79  initially_payable_to_institution        109405 non-null  int64
 80  aus-1                                   109405 non-null  int64
 81  aus-2                                   0 non-null       float64
 82  aus-3                                   0 non-null       float64
 83  aus-4                                   0 non-null       float64
 84  aus-5                                   0 non-null       float64
 85  denial_reason-1                         109405 non-null  int64
 86  denial_reason-2                         1871 non-null    float64
 87  denial_reason-3                         120 non-null     float64
 88  denial_reason-4                         0 non-null       float64
 89  tract_population                        109405 non-null  int64
 90  tract_minority_population_percent       109405 non-null  float64
 91  ffiec_msa_md_median_family_income       109405 non-null  int64
 92  tract_to_msa_income_percentage          109405 non-null  float64
 93  tract_owner_occupied_units              109405 non-null  int64
 94  tract_one_to_four_family_homes          109405 non-null  int64
 95  tract_median_age_of_housing_units       109405 non-null  int64
dtypes: float64(36), int64(47), object(13)
memory usage: 80.1+ MB
```

Out[359...

	activity_year	lei	derived_msa-md	state_code	county_code	census_t
0	2023	549300FGXN1K3HLB1R50	41884	CA	6081.0	6.081609e
1	2023	549300FGXN1K3HLB1R50	26420	TX	48201.0	4.820145e
2	2023	549300FGXN1K3HLB1R50	16984	IL	17043.0	1.704385e
3	2023	549300FGXN1K3HLB1R50	42644	WA	53033.0	5.303303e
4	2023	549300FGXN1K3HLB1R50	40220	VA	51161.0	5.116103e

5 rows × 96 columns

In [360...

```
#Data cleaning and preparation

'''
The outcome variable, "Approval," will be binary:
1: Includes approved loans and applications approved but not accepted.
0: Includes denied applications. Other cases (e.g., withdrawals, incompleteness, an
the analysis.
'''

approval_mapping = {
    1: 1,  # Approved loans
    2: 1,  # Approved but not accepted
    3: 0,  # Denied
}
df['approval'] = df['action_taken'].map(approval_mapping)
df = df.dropna(subset=['approval'])
df['approval'] = df['approval'].astype(int)
```

In [361...

```
# Categorize applicant sex
sex_mapping = {1: 'Male', 2: 'Female'}
df['applicant_sex'] = df['applicant_sex'].map(sex_mapping)
```

In [362...

```
# Filter the dataset to keep only the relevant columns

relevant_columns = [
    'approval',  # Outcome variable
    'loan_to_value_ratio',
    'state_code',
    'applicant_sex',
    'income',
    'loan_amount',
    'preapproval',
    'loan_type',
```

```
    'occupancy_type'
]

df = df[relevant_columns]
```

In [363…

```
# Apply logarithmic transformation for skewed data income and loan amount
df['log_income'] = np.log(df['income'].replace(0, np.nan))
df['log_loan_amount'] = np.log(df['loan_amount'].replace(0, np.nan))

# Drop the original columns after transformation
df = df.drop(columns=['income', 'loan_amount'])
```

In [364…

```
# Group states into broader regions (e.g., Northeast, Midwest, South, West) using U

state_to_region = {
    'AL': 'South', 'AK': 'West', 'AZ': 'West', 'AR': 'South', 'CA': 'West',
    'CO': 'West', 'CT': 'Northeast', 'DE': 'South', 'FL': 'South', 'GA': 'South',
    'HI': 'West', 'ID': 'West', 'IL': 'Midwest', 'IN': 'Midwest', 'IA': 'Midwest',
    'KS': 'Midwest', 'KY': 'South', 'LA': 'South', 'ME': 'Northeast', 'MD': 'South'
    'MA': 'Northeast', 'MI': 'Midwest', 'MN': 'Midwest', 'MS': 'South', 'MO': 'Midw
    'MT': 'West', 'NE': 'Midwest', 'NV': 'West', 'NH': 'Northeast', 'NJ': 'Northeas
    'NM': 'West', 'NY': 'Northeast', 'NC': 'South', 'ND': 'Midwest', 'OH': 'Midwest
    'OK': 'South', 'OR': 'West', 'PA': 'Northeast', 'RI': 'Northeast', 'SC': 'South
    'SD': 'Midwest', 'TN': 'South', 'TX': 'South', 'UT': 'West', 'VT': 'Northeast',
    'VA': 'South', 'WA': 'West', 'WV': 'South', 'WI': 'Midwest', 'WY': 'West', 'DC'
}
df['region'] = df['state_code'].map(state_to_region)

df = df.dropna()
len(df)
df
```

Out[364...

	approval	loan_to_value_ratio	state_code	applicant_sex	preapproval	loan_type	o
0	1	80.0	CA	Female	2	1	
1	1	95.0	TX	Male	2	1	
2	1	97.0	IL	Female	2	1	
3	1	85.0	WA	Male	1	1	
4	1	80.0	VA	Male	2	1	
...	...	...	...	...	...	...	
109400	0	86.0	MI	Male	2	1	
109401	0	70.0	CA	Female	2	1	
109402	0	75.0	CA	Male	2	1	
109403	0	97.0	FL	Female	2	1	
109404	0	80.0	ME	Male	2	1	

107918 rows × 10 columns

In [365...

```
# Encode categorical features
categorical_features = ['applicant_sex',
                         'preapproval',
                         'loan_type',
                         'occupancy_type']

for col_name in categorical_features:
    df[col_name] = df[col_name].astype('category')
```

In [374...

```
#Descriptive Statistics
df["approval"].value_counts(normalize=True)
```

Out[374...

```
approval
1    0.868493
0    0.131507
Name: proportion, dtype: float64
```

In [376...

```
'''
Most loan applications (~87%) in the dataset were approved (approval = 1) meaning t
Resampling the training set would be necessary for effective model training.
'''
```

Out[376...

```
'\nMost loan applications (~87%) in the dataset were approved (approval = 1) meani
ng that the approval variable is imbalanced.\nResampling the training set would be
necessary for effective model training.\n'
```

In [378...

```
df["applicant_sex"].value_counts(normalize=True)
```

Out[378…

```
applicant_sex
Male      0.633944
Female    0.366056
Name: proportion, dtype: float64
```

In [380…

```
'''
Similarly to the outcome variable, the applicant_sex variable is imbalanced with mo
'''
```

Out[380…

```
'\nSimilarly to the outcome variable, the applicant_sex variable is imbalanced wit
h most of the applicants being Male. \n'
```

In [382…

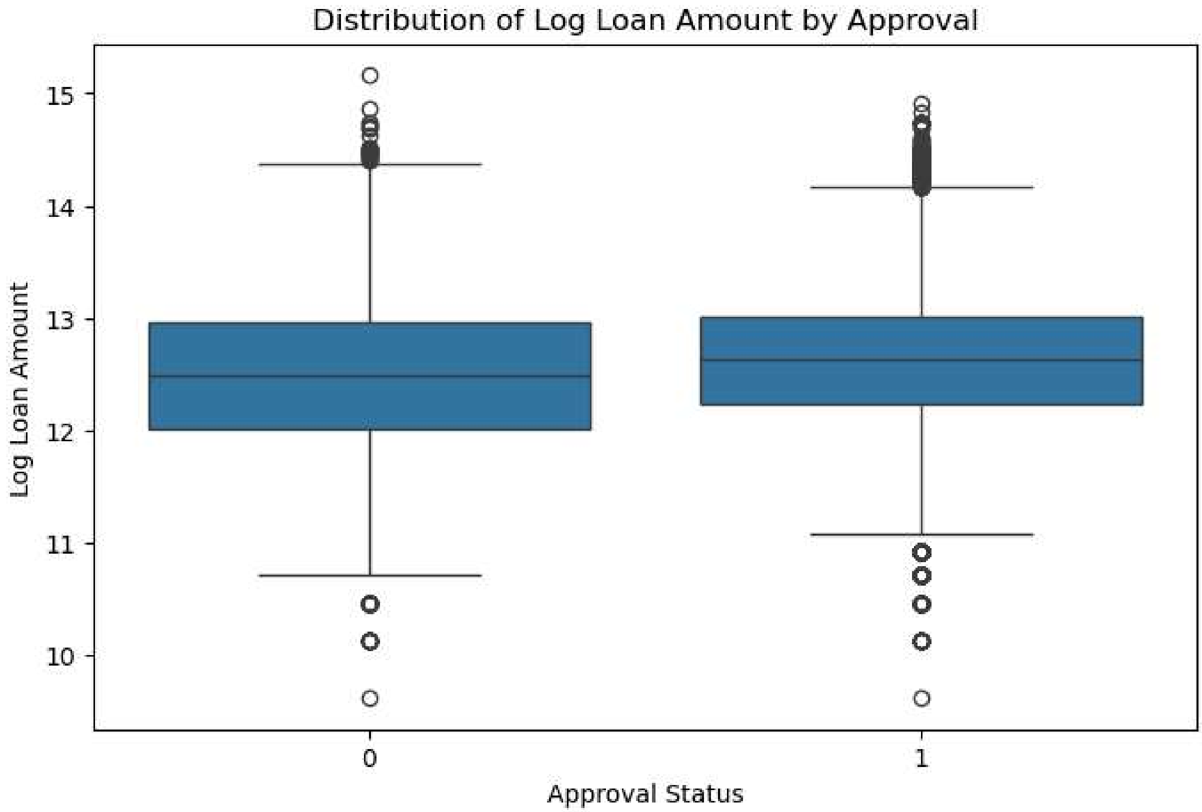
```
# Summary Statistics by Applicant Sex
summary_stats = df.groupby("applicant_sex").agg({
    "log_income": ["mean", "median"],
    "log_loan_amount": ["mean", "median"],
    "approval": "mean"
}).reset_index()
summary_stats.columns = ["Applicant Sex", "Mean Log Income", "Median Log Income", "
print("Summary Statistics by Applicant Sex:")
print(summary_stats)
```

```
Summary Statistics by Applicant Sex:
  Applicant Sex  Mean Log Income  Median Log Income  Mean Log Loan Amount  \
0        Female         4.599590           4.584967             12.482948
1          Male         4.760087           4.753590             12.610505

   Median Log Loan Amount  Approval Rate
0               12.524526       0.863432
1               12.660328       0.871415
```

In [384…

```
# Distribution of Loan Amount by Approval
plt.figure(figsize=(8, 5))
sns.boxplot(x="approval", y="log_loan_amount", data=df)
plt.title("Distribution of Log Loan Amount by Approval")
plt.xlabel("Approval Status")
plt.ylabel("Log Loan Amount")
plt.show()
```



In [386…
```
#Training the initial model
# Target and feature separation
target_variable = 'approval'
sensitive_feature = ['applicant_sex']
```

In [388…
```
Y = df.loc[:, target_variable]
```

In [390…
```
#Adjust binary variable
binary_columns = ["preapproval"]
for col in binary_columns:
    df[col] = df[col].astype(int) - 1
```

In [392…
```
X = pd.get_dummies(df.drop(columns=[target_variable, *sensitive_feature, 'state_cod
X
```

Out[392…

	loan_to_value_ratio	preapproval	log_income	log_loan_amount	loan_type_1	loan_
0	80.0	1	4.804021	13.028053	True	
1	95.0	1	4.174387	12.230765	True	
2	97.0	1	4.343805	12.560244	True	
3	85.0	0	5.081404	13.279367	True	
4	80.0	1	3.688879	12.128111	True	
...	...	...	...	...	...	
109400	86.0	1	5.273000	12.524526	True	
109401	70.0	1	5.407172	13.493927	True	
109402	75.0	1	6.712956	13.493927	True	
109403	97.0	1	3.828641	11.736069	True	
109404	80.0	1	4.779123	12.660328	True	

107918 rows × 14 columns

In [394…

```
# Handle missing values in features
imputer = SimpleImputer(strategy='mean')
X = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)
```

In [396…

```
# Train-test split
random_seed = 123
np.random.seed(random_seed)
X_train, X_test, Y_train, Y_test, df_train, df_test = train_test_split(
    X,
    Y,
    df,
    test_size=0.20,
    stratify=Y,
    random_state=random_seed
)
```

In [398…

```
# Resample to balance the dataset

def resample_dataset(X_train, Y_train):
    # Reset indices to ensure alignment
    X_train = X_train.reset_index(drop=True)
    Y_train = Y_train.reset_index(drop=True)

    # Get indices of positive and negative samples
    negative_ids = Y_train[Y_train == 0].index
    positive_ids = Y_train[Y_train == 1].index

    # Resample negative indices to match the count of positives
    balanced_negative_ids = np.random.choice(a=negative_ids, size=len(positive_ids)
```

```
    balanced_ids = np.concatenate([positive_ids, balanced_negative_ids])
    X_train_bal = X_train.loc[balanced_ids]
    Y_train_bal = Y_train.loc[balanced_ids]

    return X_train_bal, Y_train_bal
```

In [400…

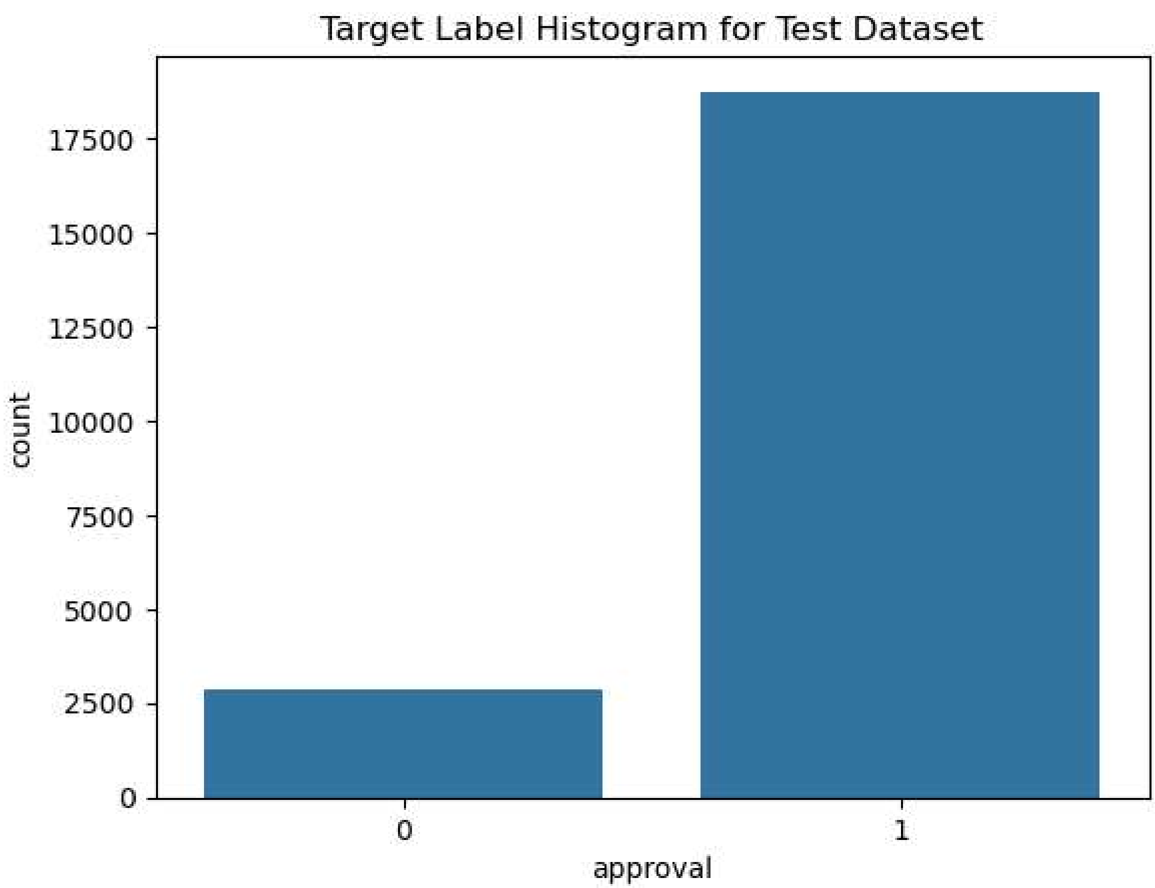
```
X_train_bal, Y_train_bal = resample_dataset(X_train, Y_train)
```

In [402…

```
#Evaluate statistics of the training and test data set
sns.countplot(x=Y_test)
plt.title("Target Label Histogram for Test Dataset")
```

Out[402…

```
Text(0.5, 1.0, 'Target Label Histogram for Test Dataset')
```

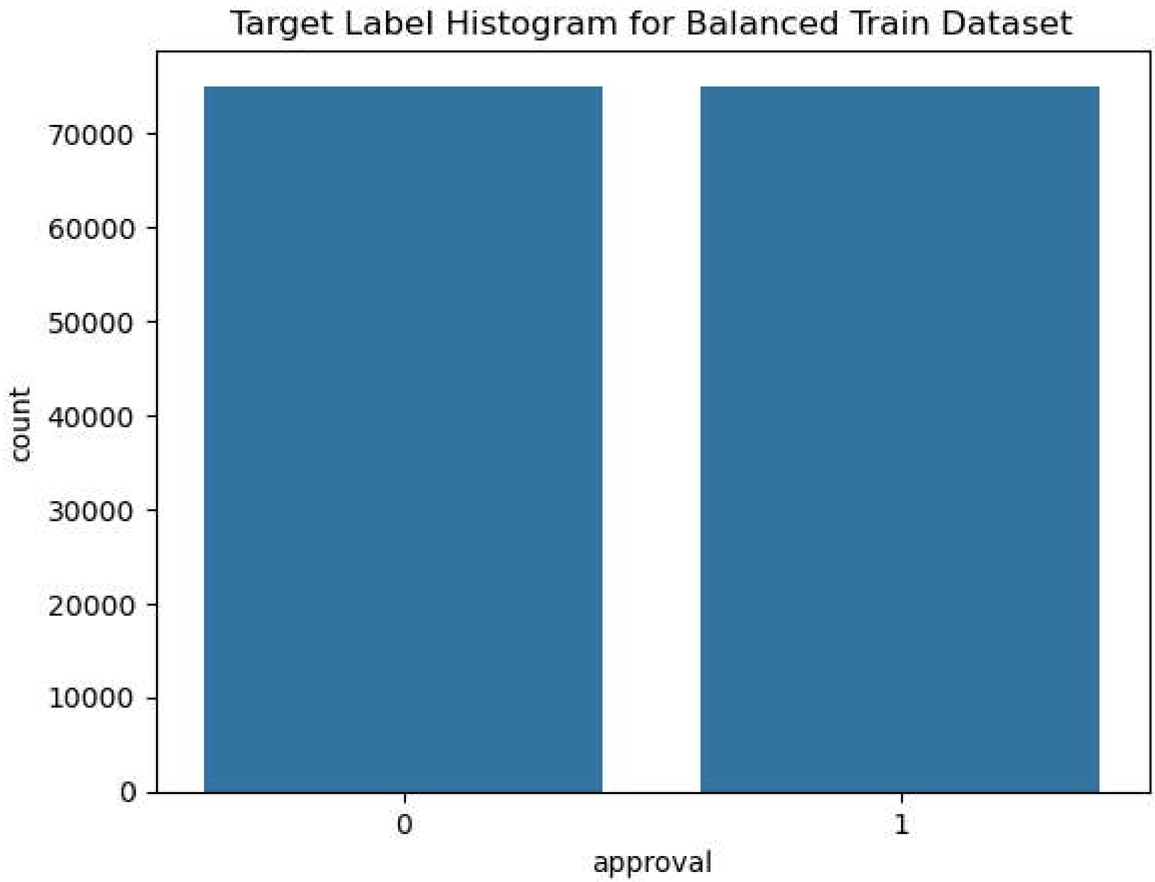


In [404…

```
sns.countplot(x=Y_train_bal)
plt.title("Target Label Histogram for Balanced Train Dataset")
```

Out[404…

```
Text(0.5, 1.0, 'Target Label Histogram for Balanced Train Dataset')
```



In [ ]:
```
'''
The train data set es balanced now so both groups are equally represented.
'''
```

In [406…
```
#Model pipeline to train a logistic regression model
pipeline = Pipeline(steps=[
    ("preprocessing", StandardScaler()),
    ("logistic_regression", LogisticRegression(max_iter=1000))
])
pipeline.fit(X_train_bal, Y_train_bal)
```

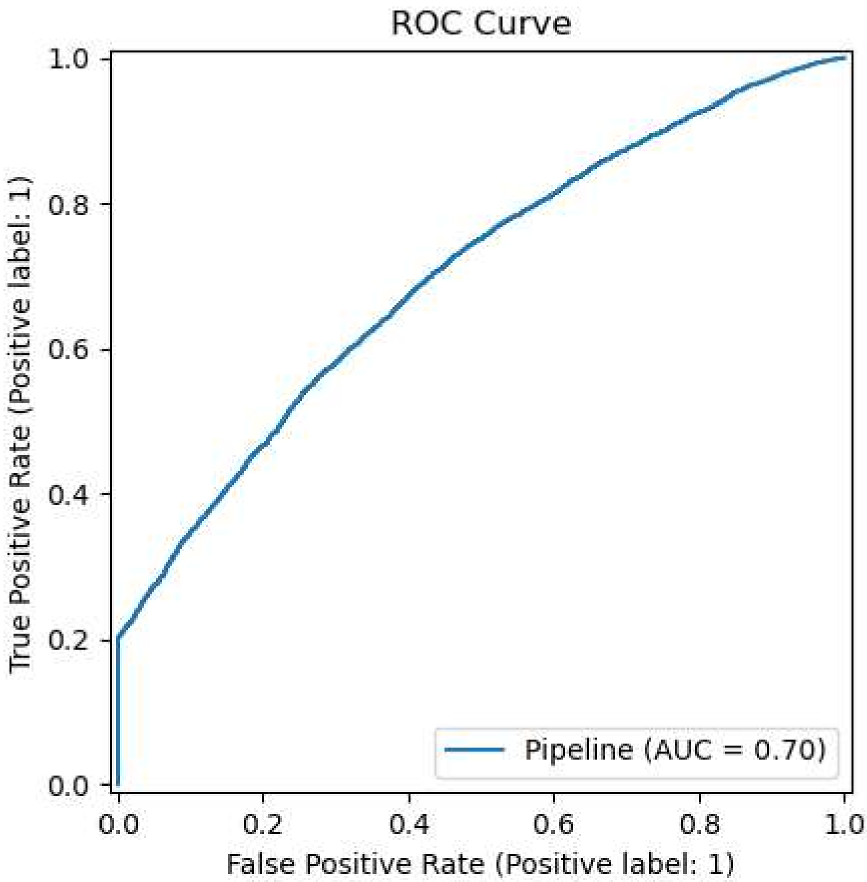
Out[406…
```
Pipeline
  StandardScaler
  LogisticRegression
```

In [408…
```
#predictions
Y_pred_proba = pipeline.predict_proba(X_test)[:,1]
Y_pred = pipeline.predict(X_test)
```

In [410…
```
#Model Evaluation Metrics
```

In [412…

```
RocCurveDisplay.from_estimator(pipeline, X_test, Y_test)
plt.title("ROC Curve")
plt.show()
```



In [414…

```
balanced_accuracy = balanced_accuracy_score(Y_test, Y_pred)
print(f"Balanced Accuracy: {balanced_accuracy:.2f}")
```

```
Balanced Accuracy: 0.64
```

In [416…

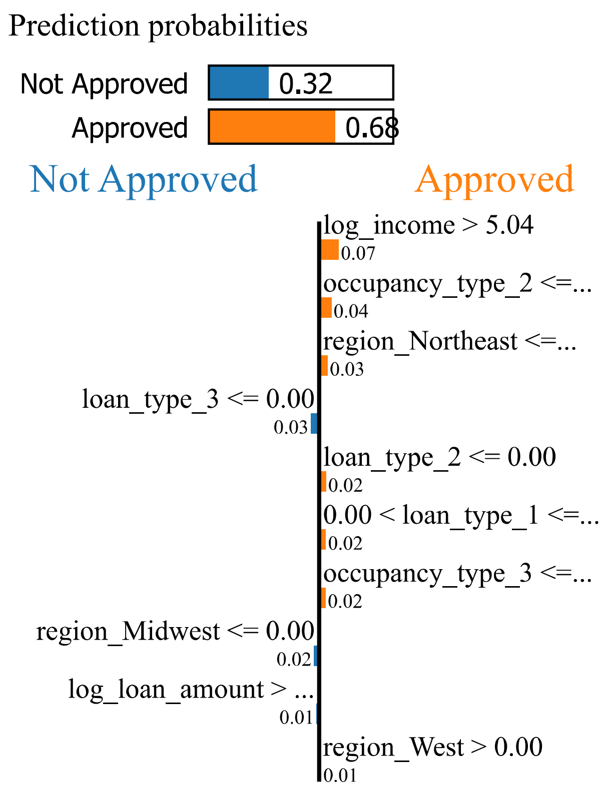
```
# Fairness metrics
mf = MetricFrame(
    metrics={
        "balanced_accuracy": lambda y_true, y_pred: balanced_accuracy_score(y_true,
        "false_negative_rate": false_negative_rate,
        "selection_rate": selection_rate,
    },
    y_true=Y_test,
    y_pred=Y_pred,
    sensitive_features=df.loc[Y_test.index, sensitive_feature].squeeze()
)

print("Fairness Metrics:")
print(mf.by_group)
```

```
Fairness Metrics:
               balanced_accuracy  false_negative_rate  selection_rate
applicant_sex
Female                  0.643331             0.512034        0.448967
Male                    0.637150             0.449212        0.515441
```

In [ ]:
```
'''Overall, the balanced accuracy was similar for males and females (approximately
 However, the selection rate showed that males had a higher selection rate than fem
```

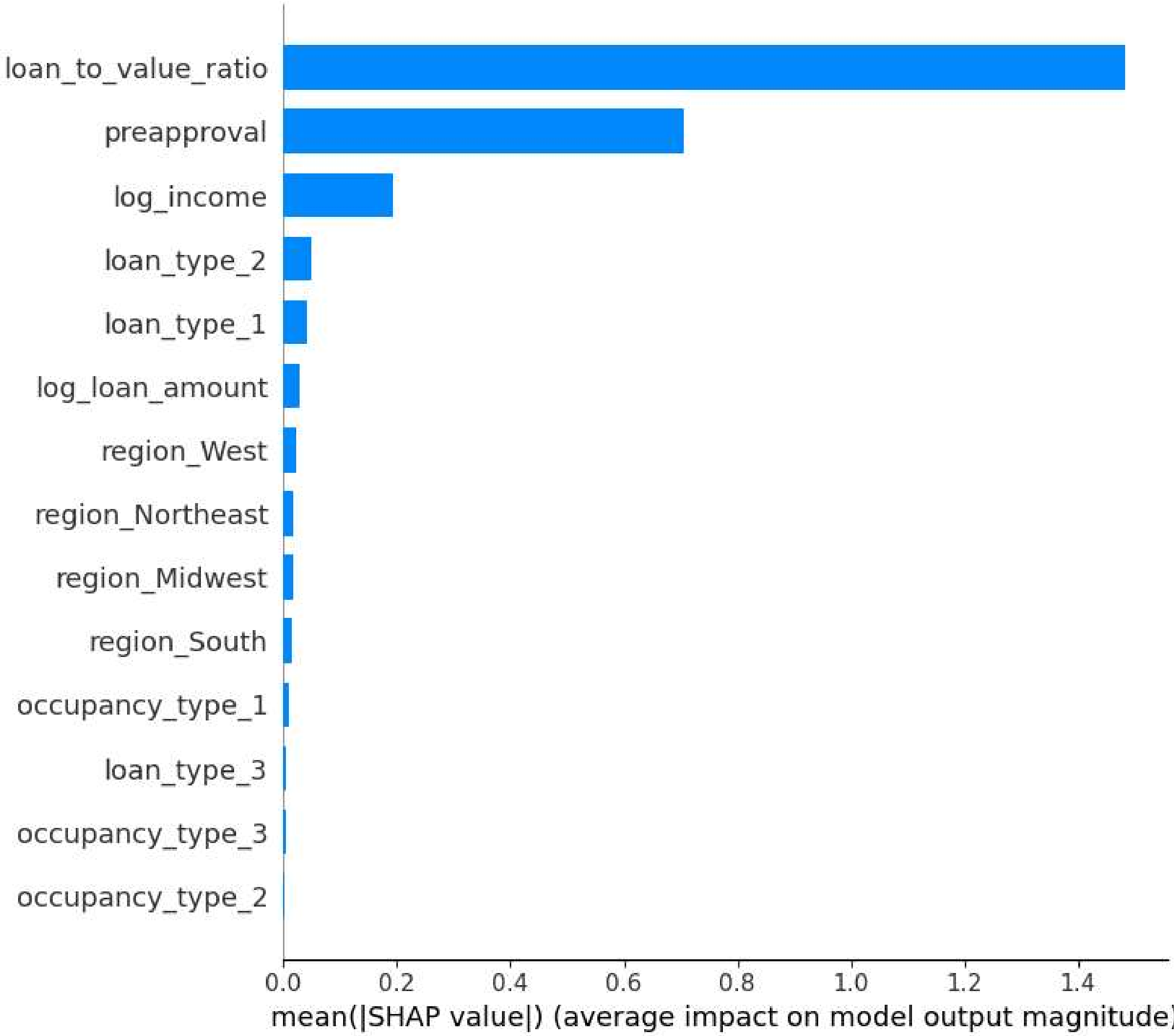
In [418…
```
# LIME explainer for the second loan application as an example
explainer_lime = LimeTabularExplainer(
    X_train_bal.values,
    feature_names=X_train_bal.columns,
    class_names=["Not Approved", "Approved"],
    discretize_continuous=True
)
lime_exp = explainer_lime.explain_instance(
    X_test.iloc[1].values, pipeline.predict_proba, num_features=10
)
lime_exp.show_in_notebook()
```



Feature	Value
log_income	5.81
occupancy_type_2	0.00
region_Northeast	0.00
loan_type_3	0.00
loan_type_2	0.00
loan_type_1	1.00
occupancy_type_3	0.00
region_Midwest	0.00

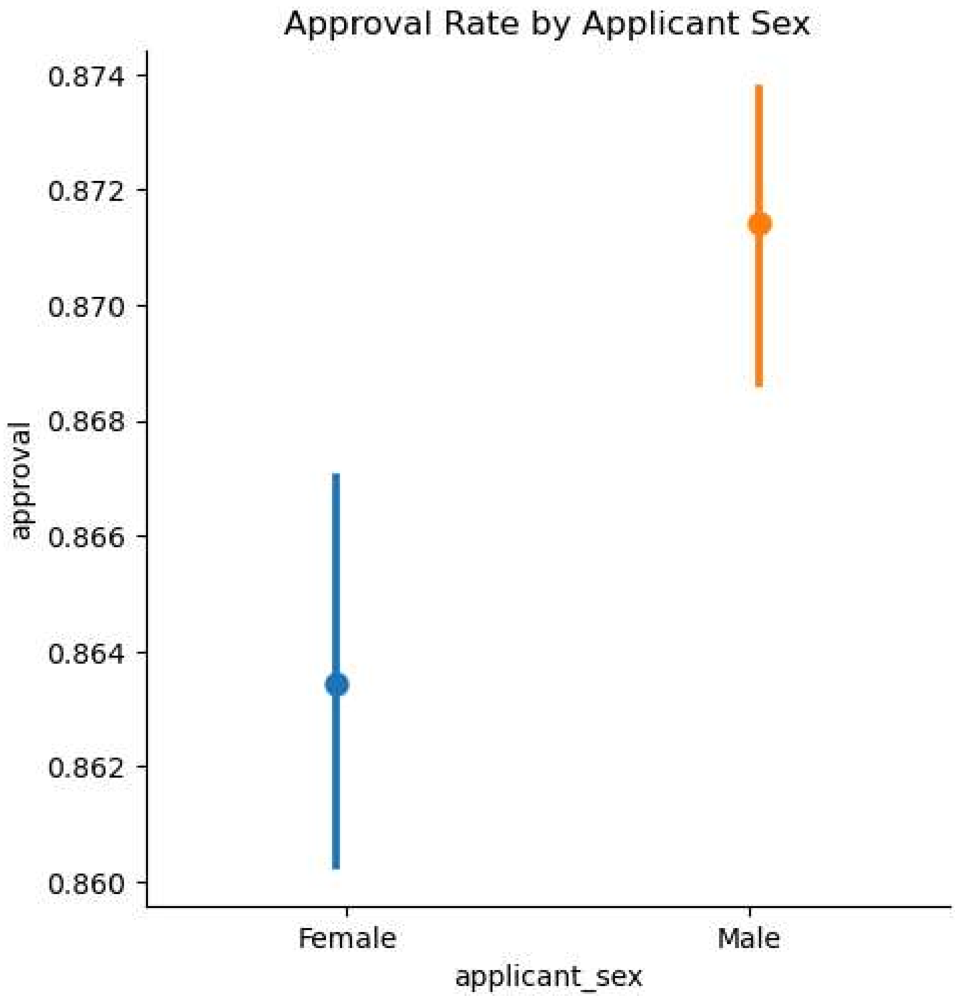
In [ ]:
```
'''
 This application is predicted to be approved with a prediction probability of 0.68
 The main factors that positively impacting the decision are the income, occupancy,
 role in the application.
 '''
```

In [419…
```
# SHAP explainer
logistic_model = pipeline.named_steps["logistic_regression"]
explainer_shap = shap.LinearExplainer(logistic_model, X_train_bal)
shap_values = explainer_shap.shap_values(X_test)
shap.summary_plot(shap_values, X_test, plot_type="bar")
```



In [ ]:
```
'''
SHAP allows us to visualize global insights and evaluate the role of each feature
loan_to_value_ratio, preapproval, and low income are among the variables with high
The loan_to_value_ratio demonstrated a higher impact, followed by the preapproval
Other features such as loan type y regions demonstrated to have some impact as wel
here.
```

In [422…
```
# Visualization: Approval by Sex
sns.catplot(y="approval", x="applicant_sex", hue="applicant_sex", data=df, kind="po
plt.title("Approval Rate by Applicant Sex")
plt.show()
```



In [ ]:
```
'''
The graph above serves as a comparison of the approval rate based on the applicant'
they were very close with approximately 0.86 females approval rate and 0.87 for mal
'''
```

In [424…
```
# Dataset Drift Report
reference = X_train_bal.copy()
reference['prediction'] = pipeline.predict(X_train_bal)
current = X_test.copy()
current['prediction'] = pipeline.predict(X_test)

report = Report(metrics=[
    DataDriftPreset()
])
report.run(reference_data=reference, current_data=current)
report.show()
```

Out[424…

Dataset Drift

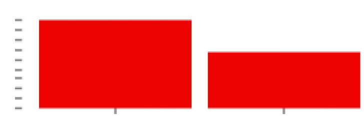
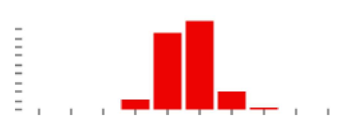
Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

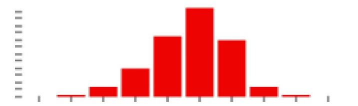
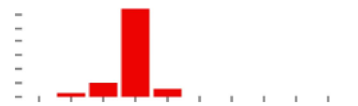
15	1	0.0667
Columns	Drifted Columns	Share of Drifted Columns

Data Drift Summary

Drift is detected for 6.667% of columns (1 out of 15).

Search

	Column	Type	Reference Distribution
›	prediction	cat	
›	log_income	num	

›	preapproval	num	
›	log_loan_amount	num	
›	loan_to_value_ratio	num	
›	loan_type_2	num	
›	loan_type_1	num	

In [ ]:

```
'''
Only 6.667% of columns (1 out of 15) showed a drift, specifically the log_income va
'''
```

In [263…

```
'''
Conclusions:
The project covered a structured approach to analyzing loan approval processes with
(applicant sex).
The developed logistic regression model effectively predicts loan approval outcomes
score. These results indicate that the model performs well in distinguishing betwee
The use of balanced training data through resampling helped address the gender imba
not disproportionately favor the one gender.
Fairness metrics calculated using the Fairlearn library revealed disparities in fal
Women had a slightly higher rate of being incorrectly denied loans compared to men,
men were marginally higher, suggesting disparities in access to credit.
These findings highlight the need for further bias mitigation strategies to ensure
SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic E
Features like income and loan-to-value ratio were significant predictors of loan ap
Gender (applicant sex) had a measurable but relatively minor impact on predictions,
characteristics.
While machine learning models can improve efficiency and accuracy in processes like
```

```
ethical implications, particularly when decisions affect people’s lives. By combini
this project offers a robust framework for ethical AI in financial services.
'''
```

Out[263…

```
'\nConclusions:\nThe project covered a structured approach to analyzing loan approval processes with a specific focus on the potential influence of gender\n(applicant sex).\nThe developed logistic regression model effectively predicts loan approval outcomes with high balanced accuracy and a strong ROC-AUC score. \nThese results indicate that the model performs well in distinguishing between approved and denied applications.\nThe use of balanced training data through resampling helped address class imbalance, ensuring that the model did \nnot disproportionately favor the majority class.\nFairness metrics calculated using the Fairlearn library revealed disparities in false negative rates and selection rates across gender groups.\nFalse Negative Rate: Women had a slightly higher rate of being incorrectly denied loans compared to men, indicating a potential bias in decision-making.\nSelection Rate: Approval rates for men were marginally higher, suggesting disparities in access to credit.\nThese findings highlight the need for further bias mitigation strategies to ensure equitable outcomes.\nSHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) provided critical insights into the model’s behavior:\nFeatures like income, loan-to-value ratio, and loan amount were significant predictors of loan approval.\nGender (applicant sex) had a measurable but relatively minor impact on predictions, suggesting that the model primarily relied on financial characteristics.\nExplainability tools offered transparency into how individual predictions were made, building trust in the model’s decisions.\nThe Evidently AI library detected minor data drifts in certain features (e.g., income distribution) between the training dataset and a newer dataset of loan applications.\nAlthough the drift was not severe, it underscores the importance of continuous monitoring and periodic retraining to maintain model validity over time.\nThis project underscores the importance of balancing predictive accuracy with fairness and transparency in AI-driven decision-making systems. \nWhile machine learning models can improve efficiency and accuracy in processes like loan approvals, their real-world deployment must consider \nethical implications, particularly when decisions affect people’s lives. By combining fairness evaluations, explainability, and data drift monitoring,\nthis project offers a robust framework for ethical AI in financial services.\n\n'
```

In [ ]:

In [ ]:

In [ ]: