ISHIMWE KAREKEZI GUY GAEL

AndrewID: iguygael

MS EAI

Email: iguygael@andrew.cmu.edu

Tel: (+250) 784595484

Course: DIAML

DIAML Assignment 3

Libraries used in the assignment

import numpy as np
import pandas as pd
import seaborn as sns
import scipy
import matplotlib
import sklearn
import shap
from matplotlib import pyplot as plt
from pandas import DataFrame
import statistics as ststc
from scipy import stats
from IPython.display import display
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import SGDClassifier
from sklearn.datasets import make_classification
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay

Q1)

Objective:

The objective for this task was to understand the structure and how our data is distributed and identify types of variables in the dataset whether categorical or numerical. Finally, we had to find skewed variables.

Results

Our dataset

Our Original dataset

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
...	...	...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semiurban	N

614 rows × 13 columns

Summary statistics

Summary statistics

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

Types of variables present:

```
Categorical attributes in our Dataset:
['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'Property_Area', 'Loan_Status']

Numerical attributes in our Dataset:
['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term', 'Credit_History']
```

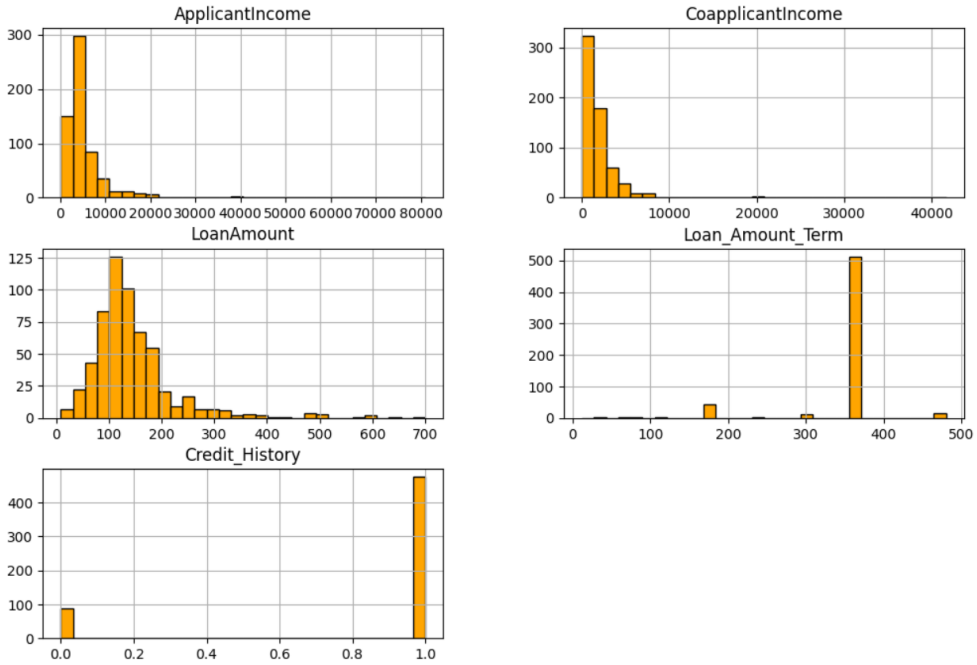
```
Our Dataset Information :
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB

Our Dataset Shape:
(614, 13)
```

Qualitative answer:

The numerical attributes are the one with int64 or float64 in our dataset information output while categorical are the ones with datatype object.

numerical features distribution and which show skewness:



```
Skewness of Numerical attributes:
ApplicantIncome      6.539513
CoapplicantIncome    7.491531
LoanAmount           2.677552
Loan_Amount_Term    -2.362414
Credit_History      -1.882361
dtype: float64

kurtosis of Numerical attributes:
ApplicantIncome      60.540676
CoapplicantIncome    84.956384
LoanAmount           10.401533
Loan_Amount_Term      6.673474
Credit_History        1.548763
dtype: float64
```

Qualitative answer:

Features such as ApplicantIncome, coapplicantIncome, and LoanAmount are highly right skewed as shown in the figure of distributions above, which is also confirmed by skewness and kurtosis calculations

Features such as credit history and loan amount term are non normal because they are discrete variables.

why transforming skewed variables might be necessary before modelling?

Reducing the Impact of Outliers: Some numbers are significantly higher than others, causing the model to misrepresent the population. After applying the log transform, these outliers were reduced, making the model more robust and representative of the population.[1]

Model Assumptions: The ApplicantIncome, coapplicantIncome, and Loan amount are highly right skewed, this can lead to poor performance of models as many assumes a normal distribution. Transforming these numbers help them spread out more evenly, allowing the model to capture the true relationship between income and credit, thus providing more reliable results.[1]

challenges of mixing categorical and numerical features in one predictive model

Datatypes incompatibility: A problem that occurs with mixing categorical and numerical variables in one model can lead to incompatibility as these models uses mathematical functions so they perform addition, multiplications and other operations, therefore they cannot work with text data such Male or Female.

Scaling issues: Another challenge arises from scaling where even after encoding the categorical variables, the models will assume that for example applicant income with a value 4000 is more valuable than Gender (which has become 1 or 0)

Encoding problems: Another issue is misrepresentation of encoded values where for example for Property_Area with more than two categories, the encoder will set rural=1, semiurban= 2, and urban=3, this will make the model to make a false relationship that urban is 3 times greater than rural which is not correct.

Q2)

Explanations:

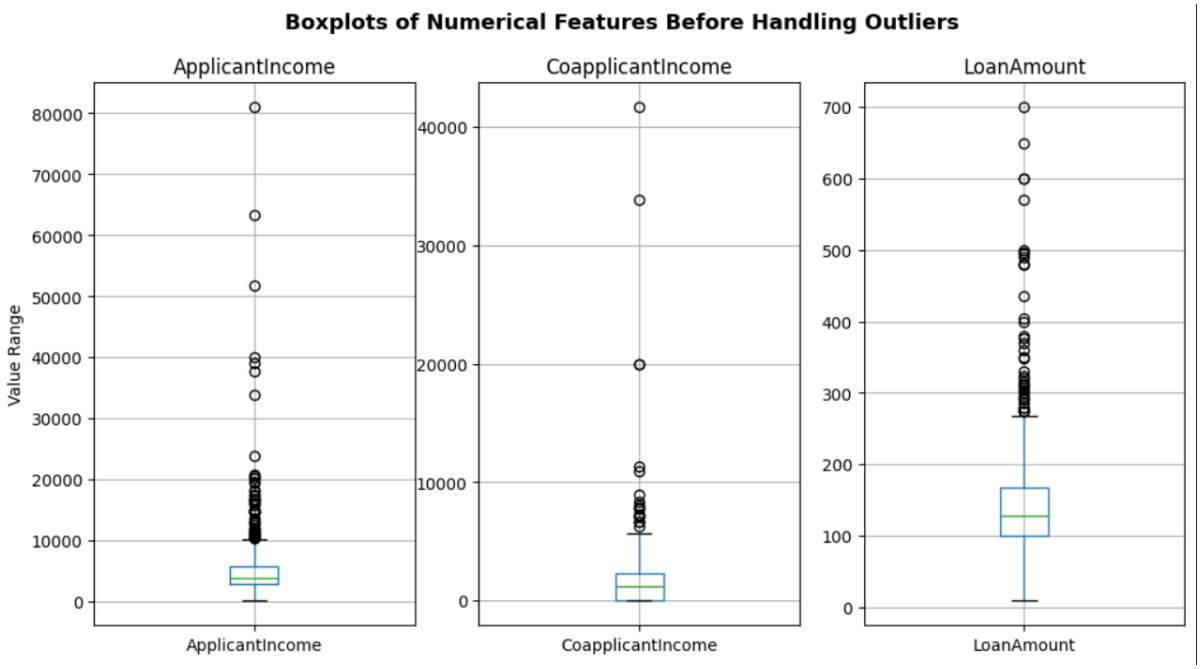
The objective for this task was to find percentage of missing values in each variable and detect unusual data and handle them.

Results:

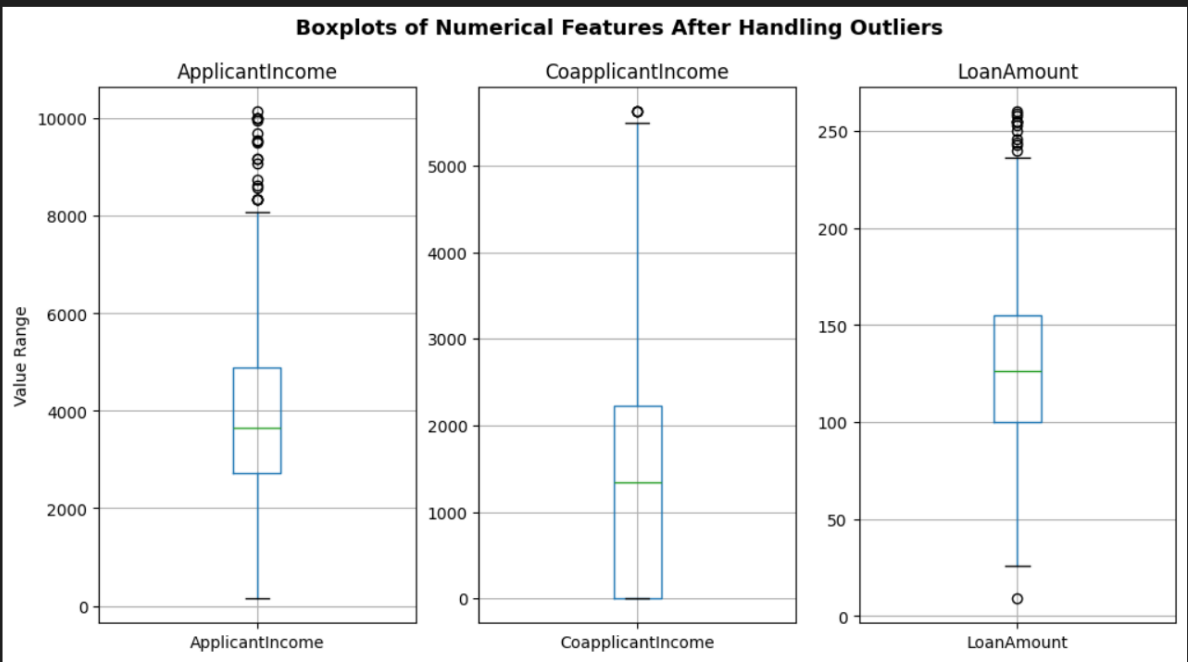
Percentage of missing values for each variable

```
percentage of missing values for each variable
Loan_ID              0.000000
Gender               2.117264
Married              0.488599
Dependents           2.442997
Education            0.000000
Self_Employed        5.211726
ApplicantIncome      0.000000
CoapplicantIncome    0.000000
LoanAmount           3.583062
Loan_Amount_Term     2.280130
Credit_History       8.143322
Property_Area        0.000000
Loan_Status          0.000000
dtype: float64
```

Boxplot before handling outliers



```
Detecting outliers using the IQR method:
ApplicantIncome       8.143322
CoapplicantIncome     2.931596
LoanAmount            6.677524
Loan_Amount_Term     14.332248
Credit_History       14.495114
dtype: float64
Outliers handled
```



Qualitative answers:

Outliers in Machine learning can make our models have high variance and lead to model poor accuracy[2] and in our visualization above, before handling outliers, we had extreme outliers in all the three numerical features, for example applicant income had outliers with 80000. After detecting the outliers using IQR method which is efficient with non-normal data[2], they were handled and the boxplot shows the change where applicant income which had 80000 before now is at 10000.

how our choices affect the data distribution and what biases could be introduced when filling in missing categorical values.

Filling missing categorical values with the mode may lead to representation bias, for example, filling missing married values can lead to strengthening the major category and will decrease our dataset variance. This can also lead to mislabelling the minor category.

Q3)

Difference between Label Encoding and One-Hot Encoding, considering their trade-offs for different types of categorical variables.

Label Encoding: This is a transformation of categorical variables that are binary (True/False) or ordinal (low, medium, high) into numeric variables which will be in the form 0 or 1 for binary data and sequence form for example 0,1, 2...., for ordinal data. This works best when there is an order.[3]

Trade-offs:

- This encoding technique cannot be used for nominal data as it can lead to assigning an order which can be a problem when using models like linear regression [3]

- Many models that depend on distance or gradient descent metrics can perform poorly when label encoding is performed as it can create false distance measurements.

One-Hot Encoding: This is a transformation of categorical nominal data into numeric data and works best if there is no ranking or order[3]

Trade-offs:

- Curse of dimensionality: when we apply one-hot encoding it increases number of columns in our dataset for example if we had column named countries with 150 countries, if we apply one hot encoding it creates 150 columns[4].

- Sparsity: As the number of columns increases, our dataset becomes sparse as many values are filled with zeros due to high dimensionality[4].

Differentiate the following scaling methods: Min-Max Scaling (Normalization), Standardization (Z-score Scaling), Robust Scaling, Unit Vector Scaling (L2 Norm Scaling)

Min-Max Scaling

This is a type of scaling where the maximum value gets assigned with '1' and the minimum value assigned with a '0' and all other values between the minimum and maximum will be assigned with a decimal value between 0 and 1[5].

Z-score Scaling

This scaling method that is used to normalize data where it re-centers the entire dataset and the new average (mean) is 0 and the new standard deviation is 1[6].

Robust Scaling

This is a method used when our data contains a lot of extremes or outliers and there is need of maintaining distances between non- extremes data points. This method uses the median to recenter our data and IQR to scale it.[7]

Unit Vector Scaling

This scaling method divides each data point by its magnitude or length to change it into a unit vector with unit length[8].

Now let us move on the coding part

Objective: The aim is to add two columns; total income and loan amount per income and identify skewed variables, then transform those skewed variables using log transformation. Additionally, we must transform categorical variables into numerical. The other task is to scale our LoanAmount, 'Total_income, loan_amount_per_income features. Finally, we must drop irrelevant variables and visualize heatmap of independent variables.

Results:

The two new added columns (Total income and Loan amount per income)

ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	Total_income	loan_amount_per_income
5849	0.0	128.0	360.0	1.0	Urban	Y	5849.0	0.021880
4583	1508.0	128.0	360.0	1.0	Rural	N	6091.0	0.021011
3000	0.0	66.0	360.0	1.0	Urban	Y	3000.0	0.021993
2583	2358.0	120.0	360.0	1.0	Urban	Y	4941.0	0.024282
6000	0.0	141.0	360.0	1.0	Urban	Y	6000.0	0.023496

Where total income is the sum of applicant and coapplicant income, and loan amount per income is loan amount divided by total income.

Skewed variables

we identified four skewed columns: ApplicantIncome, CoapplicantIncome,LoanAmount, and TotalIncome.

First 5 columns displayed:

```
our skewed variables:
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Total_income
0	5849	0.0	128.0	5849.0
1	4583	1508.0	128.0	6091.0
2	3000	0.0	66.0	3000.0
3	2583	2358.0	120.0	4941.0
4	6000	0.0	141.0	6000.0

Applying logarithmic transformation

Log_ApplicantIncome	Log_CoapplicantIncome	Log_LoanAmount	Log_Total_income
8.674197	0.000000	4.859812	8.674197
8.430327	7.319202	4.859812	8.714732
8.006701	0.000000	4.204693	8.006701
7.857094	7.765993	4.795791	8.505525
8.699681	0.000000	4.955827	8.699681

Transforming categorical columns

Before transformation:

Categorical columns in our dataset

	Gender	Married	Dependents	Education	Self_Employed	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	Male	No	0	Graduate	No	360.0	1.0	Urban	Y
1	Male	Yes	1	Graduate	No	360.0	1.0	Rural	N
2	Male	Yes	0	Graduate	Yes	360.0	1.0	Urban	Y
3	Male	Yes	0	Not Graduate	No	360.0	1.0	Urban	Y
4	Male	No	0	Graduate	No	360.0	1.0	Urban	Y

After transformation:

i) Label encoding

Here label encoding was used to transform the categorical variables into 0 or 1. Label encoding was applied to features ('Gender','Married','Education','Self_Employed','Credit_History','Loan_Status')

Gender	Married	Education	Self_Employed
1	0	0	0
1	1	0	0
1	1	0	1
1	1	1	0
1	0	0	0

Now we can observe that the categorical variables have been encoded into 1 or 0s.

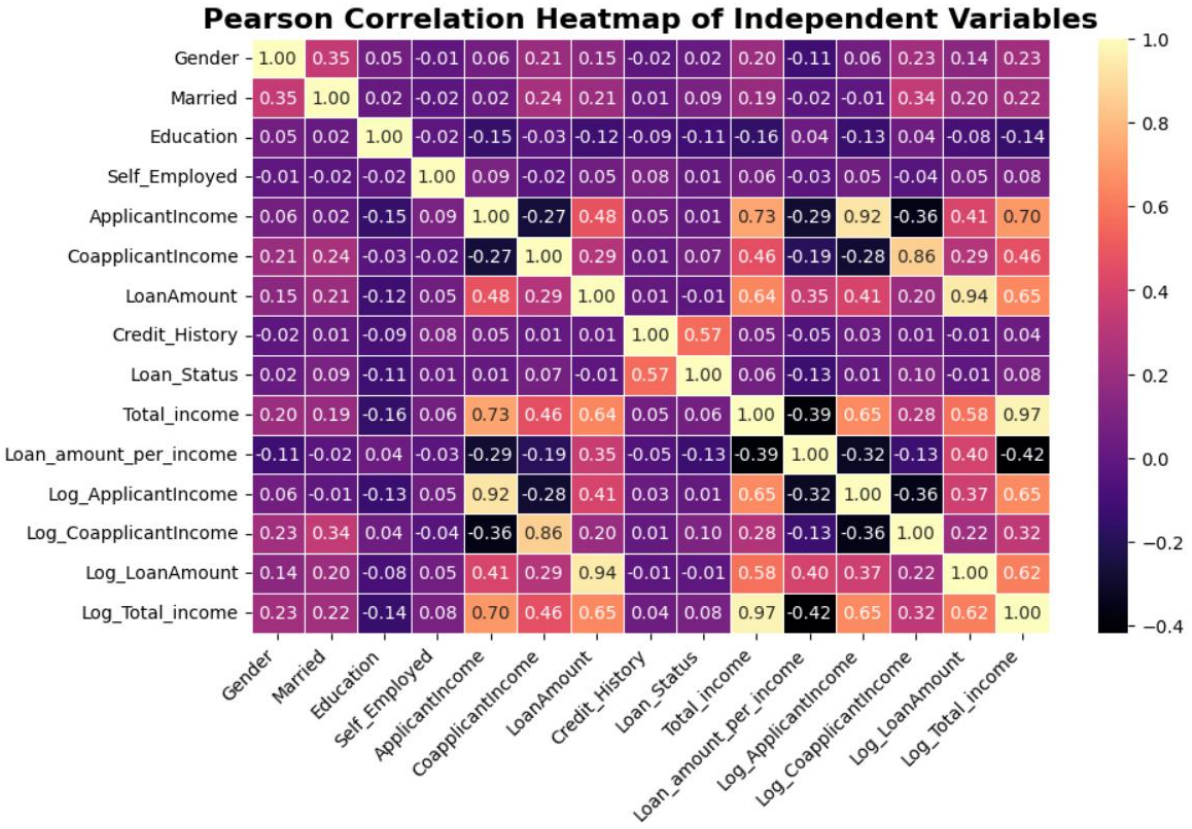
ii) One-Hot encoding

One-hot encoding was used for 'Property_Area', 'Dependents', 'Loan_Amount_Term' as it had more than two categories and were nominal. One-hot encoding was used for these features because these features don't have a ranking therefore, we create dummy columns so that models like logistic regression don't assumes a numeric or ordinal relationship between categories.

Property_Area_Semiurban	Property_Area_Urban	Dependents_1	Dependents_2	Dependents_3+	Loan_Amount_Term_36.0	Loan_Amount_Term_60.0
False	True	False	False	False	False	False
False	False	True	False	False	False	False
False	True	False	False	False	False	False
False	True	False	False	False	False	False
False	True	False	False	False	False	False

Pearson correlation Matrix

This correlation matrix uses a scale of -1 to 1 where -1 shows a strong negative relationship, 0 shows no linear relationship between two variables and 1 demonstrates strong positive relationship between two variables.



Qualitative answers:

The matrix shows that our diagonal is perfectly correlated which makes sense as it is comparing a variable with itself.

The most key insight from the matrix is that there is no significant multicollinearity that occurs when two variables are too similar and this can confuse the model.

Even though we have some positive correlation of +0.73 for total income and applicant income which makes sense as total income is derived from applicant income so there is some relationship and +0.64 for total income and loan amount which also makes sense because people with higher incomes tend to have larger loans.

Another key insight from the correlation matrix is the strong positive relationship between log transformed variables and their pre-log transformation variables. This shows us that we have to drop irrelevant and redundant variables such as 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Total_income'. This helps us avoid multicollinearity.

The final scaled dataset

the scaled final dataset is:

	Loan_ID	Gender	Married	Education	Self_Employed	Credit_History	Loan_Status	Loan_amount_per_income	Log_ApplicantIncome	Log_CoapplicantIncome	...	Dependents_3+	Loan_Amount_T
0	LP001002	1	0	0	0	1	1	-0.344771	0.990468	-1.148115	...	False	
1	LP001003	1	1	0	0	1	0	-0.450847	0.475332	0.781100	...	False	
2	LP001005	1	1	0	1	1	1	-0.331062	-0.419512	-1.148115	...	False	
3	LP001006	1	1	1	0	1	1	-0.051707	-0.735533	0.898866	...	False	
4	LP001008	1	0	0	0	1	1	-0.147581	1.044300	-1.148115	...	False	

Qualitative answer:

We need to scale our data before training our models to ensure that all our features contribute equally. Features that need scaling are 'Log_ApplicantIncome', 'Log_CoapplicantIncome', 'Log_LoanAmount', 'Log_Total_income', and 'Loan_amount_per_income'.

Using standardScaler (), we obtain a scaled dataset that will scale everything to prevent our models to be biased towards certain features, for example, in first dataset we see total income with 8.6 this will cause the model to think this feature is very important than self employed (0 or 1). Therefore, scaling our dataset was needed to remove biases when we start predicting.

Q4)

Imbalanced dataset: This occurs when a category in dataset is unequally represented than other categories, for example in fraud detection most of transactions will legitimate while illegitimate class will have few examples. This imbalance can lead the model to bias towards the majority class and ignoring the minor category. This leads to reduction in model's performance and leads to skewness of evaluation metrics[9]

Metrics used for evaluating models on imbalanced datasets
F1 score: Metric used to measure accuracy of binary classification model where it calculates harmonic mean of precision and recall[10]

$$\boldsymbol{F1\ Score = 2 * \frac{precision * recall}{precision + recall}}\ [10]$$

Precision: Metric used to measure how many of the model's positive predictions were correct and is useful when a false positive is very bad. For example: It would be acceptable for a spam email to be classified as important rather than an important email being classified as spam.[10]

$$\boldsymbol{Precision = \frac{TP}{TP+FP}} \quad \textbf{[10]}$$

Recall: Metric used to measure how many actual predictions that were correctly identified by the model. This is used when we have sensitive cases where getting a false negative is more costly than getting a false positive. For example, model predicting you don't have cancer when you have cancer.

$$\boldsymbol{Recall = \frac{TP}{TP + FN}}\ [10]$$

AUC score (AUC ROC)
This is a plot that shows the True Positive Rate (TPR) and the False Positive Rate (FPR) at different categorization criteria in a graph.[10]

Confusion Matrix

This is table used to evaluate a classification model performance by comparing the model's predictions against actual data.[11]

Detecting bias or variance issues in our models.

i) Testing a Model with High Variance (Overfitting)

This happens when a model learns so well from the data it has been trained on that it remembers that information rather than learning a general pattern of decision making.

How to detect: Compare how it performed on the training set with how it performed on new data (test set). If the score on the training set is very high but on the new data it is very low, it indicates that the model is not performing well on data it has not seen before.

ii) Testing for High Bias (Underfitting)

This occurs when the model used is too simple to capture the relationships between different elements in the data.
How to detect: You look at the model's scores on the data it trained on and on new data. If the scores are low on both sides (like 52% for training and 49% for testing), it means that the model is under-fitting.

iii) Checking for Algorithmic / Fairness Bias

This occurs when a model is biased or biased against different groups (such as men and women or educated and uneducated).

How to detect:

Compare the ratios between the classes and using SHAP to detect classes that are misrepresented and understand why they are misrepresented.

Now back on coding part

Results:

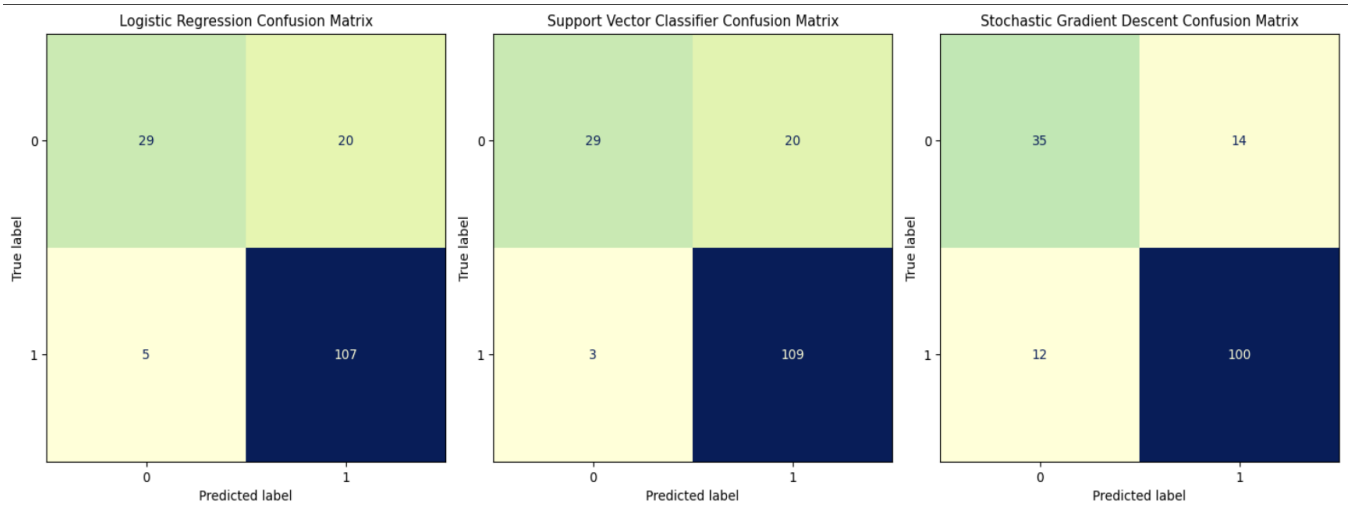
i) Trained models

```
Logistic Regression Model Accuracy on Test Set in %: 84.472
Support Vector Classifier Model Accuracy on Test Set in %: 85.714
SGD Classifier Model Accuracy on Test Set in %: 83.851
```

Qualitative answer:

SVM model with 85.714% had higher accuracy than logistic regression and SGD model.

ii) Confusion matrices



Qualitative answer:

Logistic regression predicted correctly 107 ‘approved’ cases while incorrectly predicting 20 ‘approved’ cases, it also predicted correctly 29 ‘denied’ cases while the model incorrectly predicted ‘denied’ for 5 people who were approved.

Support vector Machine predicted correctly 109 ‘approved’ cases while incorrectly predicting 20 ‘approved’ cases, it also predicted correctly 29 ‘denied’ cases while the model incorrectly predicted ‘denied’ for 3 people who were approved.

Stochastic gradient descent model predicted correctly 100 ‘approved’ cases while incorrectly predicting 14 ‘approved’ cases, it also predicted correctly 35 ‘denied’ cases while the model incorrectly predicted ‘denied’ for 12 people who were approved.

Classification Report:

Classification Report for Logistic Regression Model:

	precision	recall	f1-score	support
0	0.852941	0.591837	0.698795	49.00000
1	0.842520	0.955357	0.895397	112.00000
accuracy	0.844720	0.844720	0.844720	0.84472
macro avg	0.847730	0.773597	0.797096	161.00000
weighted avg	0.845691	0.844720	0.835562	161.00000

Classification Report for Support Vector Classifier Model:

	precision	recall	f1-score	support
0	0.906250	0.591837	0.716049	49.000000
1	0.844961	0.973214	0.904564	112.000000
accuracy	0.857143	0.857143	0.857143	0.857143
macro avg	0.875606	0.782526	0.810307	161.000000
weighted avg	0.863614	0.857143	0.847190	161.000000

Classification Report for Stochastic Gradient Descent Model:

	precision	recall	f1-score	support
0	0.744681	0.714286	0.729167	49.000000
1	0.877193	0.892857	0.884956	112.000000
accuracy	0.838509	0.838509	0.838509	0.838509
macro avg	0.810937	0.803571	0.807061	161.000000
weighted avg	0.836863	0.838509	0.837542	161.000000

Best Model

Referring to our classification model, Support vector classifier (SVM) is our best model as it has the highest accuracy of 85.714% and highest recall of 0.97 for approved (1) which means it is great at predicting approved loans. The SVM has also the highest precision for denied cases (0) of 0.904 which provides a good balance between sensitivity and specificity.

The SVM having the best macro avg F1- score among the three demonstrates that it is the best-balanced model among the three.

Q5)

Bias detection: In machine learning, bias detection is the process of verifying if the model's predictions are unfair or biased toward specific groups. A model is considered biased if it works significantly worse for one group than another, for example, if it makes Hate decisions more often for one gender than another, or if it uses specific information such as race or gender to make decisions in a biased way. [12]

How to detect biases?

Compare Results for Different Groups: dividing the test data into groups and compare different metrics like accuracy, recall, and F1-score among the groups. This will help identify low scoring groups and that can tell us the model performs worse for those groups.

Use Explainability Tools (like SHAP): Using SHAP helps us indicate how each feature influence predictions for different groups.[13]

Check for false positives and false negatives: After predictions, check for false results for example when the model's output is denying loans for unmarried people, which can be an indicator of bias.

Bias mitigation strategies

Re-sampling and Re-weighting: During pre-processing stage, increasing the number of examples to a smaller class or decreasing the number of examples to a larger class. Re-weighting can also be applied to give more meaning to the minority group during the training phase.

Adversarial Debiasing: this is in-training process that uses another model to check if the main model is working fairly and neutral.

Use of balanced training: applying classweight='balanced' makes our model to consider even the underrepresented groups.

Threshold adjustment: In this post-processing phase, we set a decision threshold for performance stability. For example, if the model is heavily denying a certain group, then lower the approval threshold.

SHAP vs LIME

SHAP (SHapley Addictive exPlanations): is a method for giving each feature a value that shows how much it adds to the output of a model. Its great advantage is that it can provide information on a single factor and also provide a picture of the performance of the entire model.[13]

How does it work

SHAP calculates the value of the contribution of each feature to the final result. It performs a deep analysis that tries every possible combination of data, seeing how the result changes

when a single feature is present or absent. It is a very reliable method for determining the actual impact of each factor.[13]

Advantages of SHAP:

It is accurate and reliable: It is based on solid data, making its interpretation reliable.[13]

It provides a complete picture: It shows both local such as individual decisions and global such as the values of the model as a whole interpretations.[13]

It has good graphics: It offers many ways to display data such as summary plots and force plots that help to understand the model in depth.[13]

Disadvantages of SHAP:

It is time-consuming: Because of this in-depth analysis, it is slow, especially for large data sets.[13]

It is difficult to understand: The mathematical model is very difficult to understand in depth.[13]

LIME (Local Interpretable Model-agnostic Explanations): LIME is a method used to explain why a Machine Learning model gave a certain answer. It is called Local because it focuses on explaining only one decision at a time and does not explain the performance of the entire model.

How does it work

LIME runs a small experiment. It takes the original data for example, the data of a person who applied for a loan, makes small changes to it such as a small increase in salary, a reduction in debt, etc., and observes how the model's results change. This helps it find features that played a significant role in that single decision.

Advantages of LIME:

It is easy to understand: Its mechanism is easy to explain and understand.

It works on all models: It is model agnostic, so you can use it on any model.

It is fast: Since it only evaluates one thing, it doesn't take much time.

Disadvantages of LIME:

It is not consistent: Its meaning can change if you use it repeatedly on the same thing.

It is not a global view: It only shows the reason for a single decision; it cannot tell you the global view of the whole model.

Now on the coding part

Objectives:

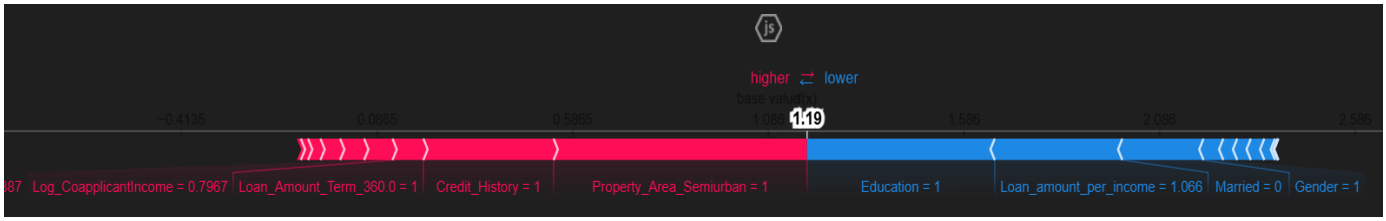
Use SHAP to identify individual contributions for each feature.

Results

i) Feature contributions to individual predictions in RL

Below is a force plot which will show us why a model chose a specific prediction for a single person.

The first plot is for the first person in our dataset

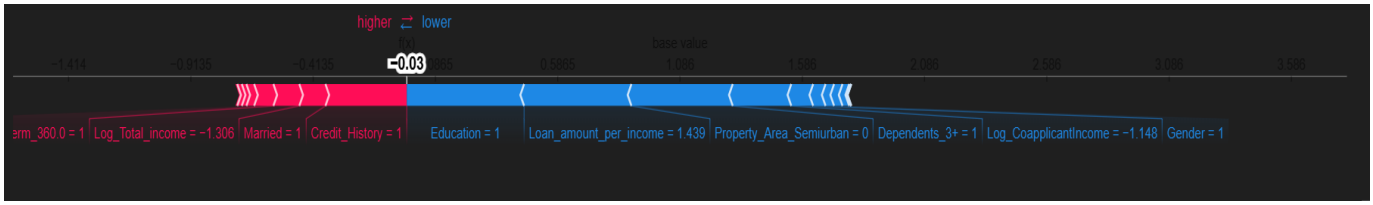


Qualitative answer:

The plot shows that for this individual; the loan was approved because the red factors outweighed the blue.

The applicant income, loan amount term, and credit history made the model approve the loan.

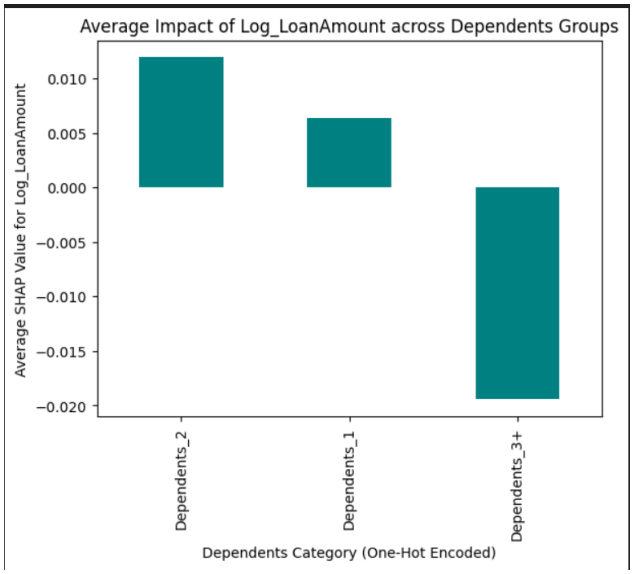
Now for the tenth person:



Qualitative answer:

The plot shows that for this individual; the loan was denied because the blue factors outweighed the red. This led to the model denying him the loan.

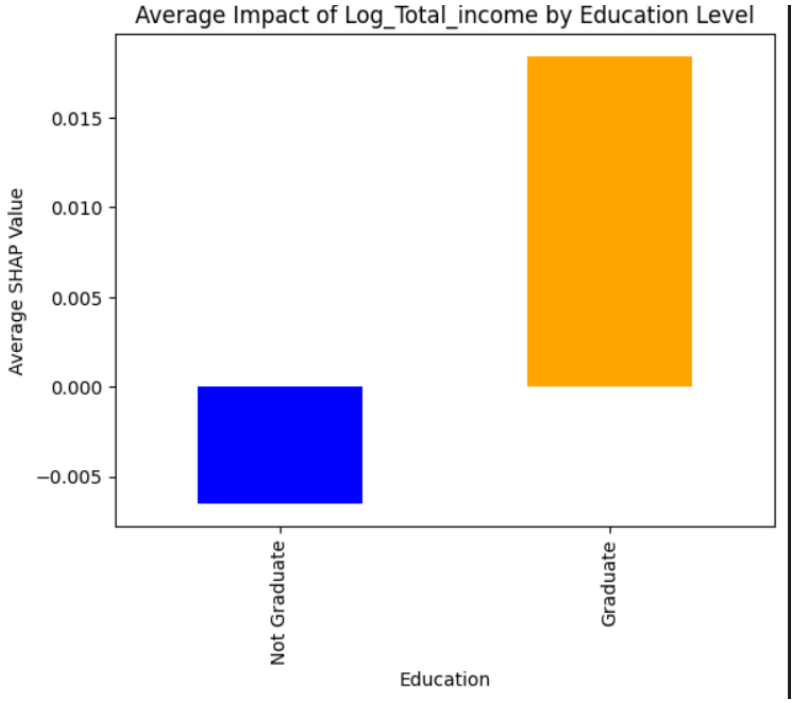
ii) Compare feature contributions across groups



Qualitative answer:

This plot shows that if an applicant had more dependents, the model treated them as a major risk factor and pushed hard towards denied.

This proves model's bias against this group because if they have 1 or 2 dependents it will approve but for 3 or more dependents, it will penalize them.

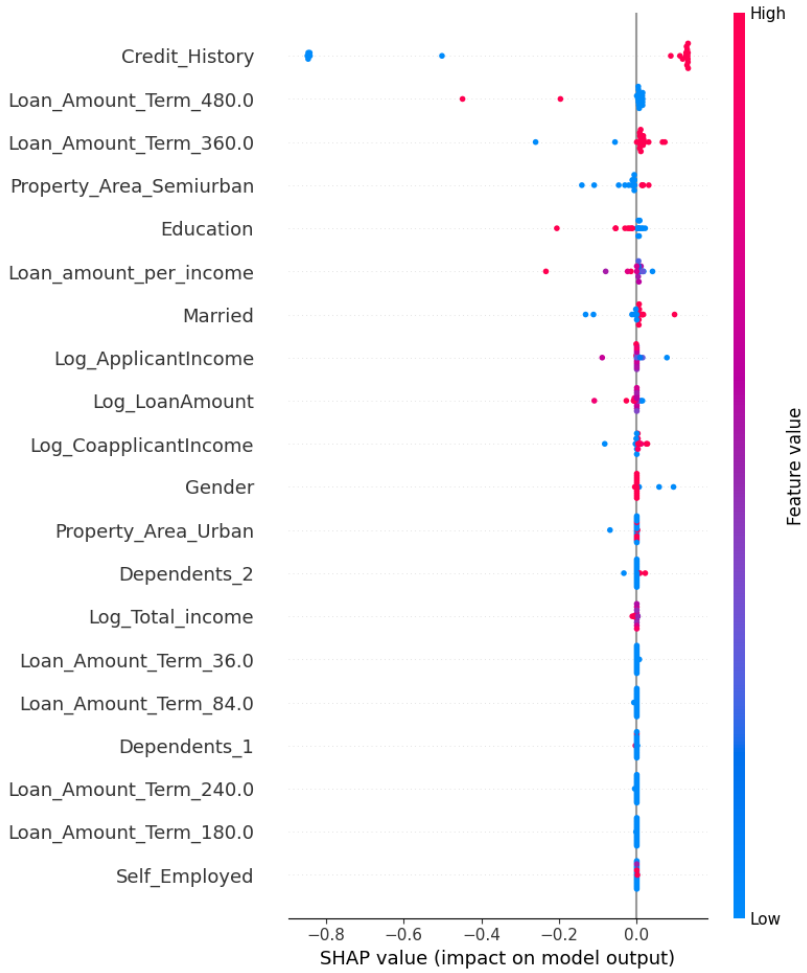


Qualitative answer:

The plot above demonstrates that the not graduate group is heavily penalized as the model consider their income as negative factor, which makes the model to lean towards Denied for not graduates.

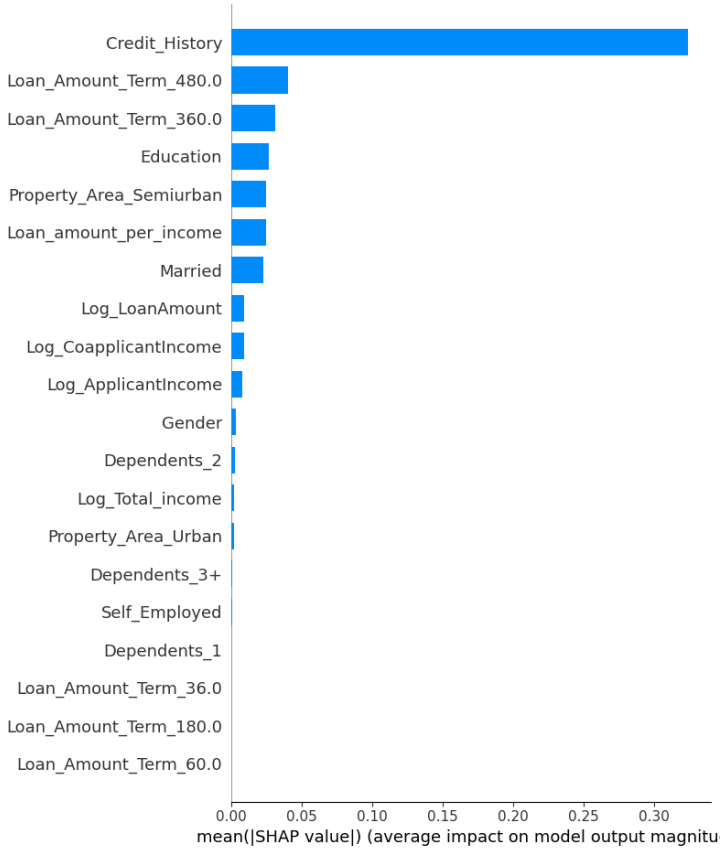
This proves the model has learned an unfair rule, a non graduate's income is a bad sign while a graduate's income is good factor. This shows the model's biasness towards non-graduates, even if they have higher income.

iii) Identify features driving prediction bias (ones that unfairly influence the outcome)



Qualitative answer:

The model's predictions are heavily relying on financial status indicators like credit history, loan term, and education rather than demographic features.



This plot shows us the most important feature is credit history. A good credit history will influence heavily the model's decision on whether an applicant's request for loan is approved or denied. This aligns with real life where a poor credit score can lead to low loan approval.

How to make the model fairer and more transparent

The model showed discrimination against the uneducated and those with more than three children and therefore needed to be corrected by adding data for these groups or by setting different criteria for each group to ensure equality. SHAP analysis showed that credit history plays a significant role in the model's decisions and helped explain why a particular person was denied a loan. The group diagrams also revealed hidden discrimination in the model's performance, which helped to make an important step towards combating it and improving fairness in its operation.

References:

[1] "Is skewness something to worry about? | Kaggle." Accessed: Nov. 09, 2025. [Online]. Available: https://www.kaggle.com/questions-and-answers/539888

[2] "How to Detect Outliers in Machine Learning," GeeksforGeeks. Accessed: Nov. 09, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/machine-learning-outlier/

[3] "One Hot Encoding vs Label Encoding," GeeksforGeeks. Accessed: Nov. 09, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/one-hot-encoding-vs-label-encoding/

[4] T. @ DataMantra, "Issues with One-Hot Encoding," Medium. Accessed: Nov. 09, 2025. [Online]. Available: https://datamantra.medium.com/issues-with-one-hot-encoding-b54aa7368589

[5] "Normalization: Min-Max and Z-Score Normalization," Codecademy. Accessed: Nov. 09, 2025. [Online]. Available: https://www.codecademy.com/article/min-max-zscore-normalization

[6] "Z-Score Normalization: Definition and Examples," GeeksforGeeks. Accessed: Nov. 09, 2025. [Online]. Available: https://www.geeksforgeeks.org/data-analysis/z-score-normalization-definition-and-examples/

[7] "StandardScaler, MinMaxScaler and RobustScaler techniques - ML," GeeksforGeeks. Accessed: Nov. 09, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/

[8] "Feature Engineering: Scaling, Normalization and Standardization," GeeksforGeeks. Accessed: Nov. 09, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/Feature-Engineering-Scaling-Normalization-and-Standardization/

[9] D. Rathi, "Handling Imbalanced Data: Key Techniques for Better Machine Learning," Medium. Accessed: Nov. 09, 2025. [Online]. Available: https://medium.com/@dakshrathi/handling-imbalanced-data-key-techniques-for-better-machine-learning-6e33b466f8b7

[10] "Evaluation Metrics in Machine Learning," GeeksforGeeks. Accessed: Nov. 09, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/metrics-for-machine-learning-model/

[11] "Understanding the Confusion Matrix in Machine Learning - GeeksforGeeks." Accessed: Nov. 11, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/confusion-matrix-machine-learning/

[12] "Bias in Machine Learning: Identifying, Mitigating, and Preventing Discrimination," GeeksforGeeks. Accessed: Nov. 11, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/bias-in-machine-learning-identifying-mitigating-and-preventing-discrimination/

[13] "LIME vs SHAP: A Comparative Analysis of Interpretability Tools." Accessed: Nov. 11, 2025. [Online]. Available: https://www.markovml.com/blog/lime-vs-shap