Enhancing Credit Risk Assessment through Advanced Machine Learning Techniques

Phase 2 Review 2

Project Group No : 244

Group Supervisor – Dr . Preetam Suman

Group Members

- ❑ 21BCE10443 Harshit Varshney
- ❑ 21BCE11188 Adarsh Kanungo
- ❑ 21BCE10470 Giya Ambasta
- ❑ 21BCE10503 Saumadipta Chatterjee

Objective

Aim: The objective is to predict whether to approve or deny a loan application using customer data.

Purpose:

- Optimize the bank’s lending strategy.
- Reduce the occurrence of non-performing loans.
- Assist the bank in making data-driven credit decisions.

Motivation:

- Address existing gaps in credit risk modeling.

Clarity and Significance of the Problem Statement

- **Problem Statement:**
  - The bank faces challenges in accurately predicting credit risk for new and existing customers, leading to increased bad loans and impacting profitability.
- **Significance:**
  - Addressing this issue with advanced modeling will reduce loan defaults, improve profitability, and enhance customer profiling.

Feasibility and Scope of the Project

- **Feasibility:**
  - Leverages available internal and external datasets (Bureau dataset, Internal product dataset).
  - Machine learning models can be deployed to automate risk assessment.
- **Scope:**
  - Improves risk assessment for credit cards, personal loans, housing loans, etc.
  - Can be scaled to other products and sectors of the bank's business.

Existing Projects

Similar Projects:

- Previous credit scoring systems rely on rule-based models or traditional scoring techniques like logistic regression.

Difference:

- Used 2 datasets 1.Bank Internal dataset 2.Cibil dataset for more accurate results
- Proposed model uses advanced ML techniques (e.g., random forest, XGBoost) to enhance prediction accuracy.

Initial Research Findings:

- Higher accuracy in predicting delinquencies compared to existing models.
- Improved risk segmentation based on customer behavior and profile data.

Some Basic Banking Terms

Asset: Loans like housing, personal, vehicle, and education loans. These are the bank's assets because they generate income through interest payments.

Liability: Products like current accounts, savings accounts, fixed deposits, and RDs. These are considered liabilities because the bank must pay interest on these accounts.

NPA (Non-Performing Asset): A loan is considered NPA when payments are overdue (Days Past Due, DPD) for more than 90 days. It indicates that the loan is in default.

Disbursed Amount: The amount of the loan granted to the customer.

Outstanding Principal (OSP): The remaining loan balance that the customer still owes. Ideally, it should be zero at the end of the loan term.

DPD (Days Past Due): The number of days a payment has been delayed. DPD should ideally be zero, meaning no payment delay.

PAR (Portfolio at Risk): The outstanding principal at risk when DPD is greater than zero, indicating potential for default.

Credit Risk Types:

- **Non-delegate account (NDA)**: DPD = 0, meaning no payment delays.
- **SMA1 (Special Mention Account 1)**: DPD is between 0-30 days.
- **SMA2 (Special Mention Account 2)**: DPD is between 31-60 days.
- **SMA3 (Special Mention Account 3)**: DPD is between 61-90 days.
- **NPA (Non-Performing Asset)**: DPD is between 90-180 days.
- **Written Off**: DPD is greater than 180 days, meaning the loan is deleted from the bank's books, but recovery may still be pursued.

NPA Categories:

- **GNPA (Gross NPA)**: The total amount of NPA, typically between 3-5% of the bank’s outstanding principal.
- **NNPA (Net NPA)**: This is GNPA minus the provisioning amount (the amount set aside to cover potential losses), generally ranging from 0.01-0.06%. Lower NNPA reflects better financial health of the bank.

Datasets & Their Feature Description

Case Study 1: Internal product dataset containing 25 features related to the bank's customer loan product holdings.

Case Study 2: CIBIL dataset with features to predict loan priority using CIBIL scoring rules (P1, P2, P3, P4).

Datasets & Their Feature Description

Case Study 1: Internal product dataset containing 25 features related to the bank's customer loan product holdings.

Variable Name	Description
Total_TL	Total trade lines/accounts in Bureau
Tot_Closed_TL	Total closed trade lines/accounts
Tot_Active_TL	Total active accounts
Total_TL_opened_L6M	Total accounts opened in last 6 Months
Tot_TL_closed_L6M	Total accounds closes in last 6 months
pct_tl_open_L6M	Percent accounts opened in last 6 months
pct_tl_closed_L6M	percent accounts closed in last 6 months
pct_active_tl	Percent active accounts
pct_closed_tl	Percent closed accounts
Total_TL_opened_L12M	Total accounts opened in last 12 Months
Tot_TL_closed_L12M	Total accounts closed in last 12 months

pct_tl_open_L12M	Percent accounts opened in last 12 months
pct_tl_closed_L12M	percent accounts closed in last 12 months
Tot_Missed_Pmnt	Total missed Payments
Auto_TL	Count Automobile accounts
CC_TL	Count of Credit card accounts
Consumer_TL	Count of Consumer goods accounts
Gold_TL	Count of Gold loan accounts
Home_TL	Count of Housing loan accounts
PL_TL	Count of Personal loan accounts
Secured_TL	Count of secured accounts
Unsecured_TL	Count of unsecured accounts
Other_TL	Count of other accounts
Age_Oldest_TL	Age of oldest opened account
Age_Newest_TL	Age of newest opened account

Case Study 2: CIBIL dataset with features to predict loan priority using CIBIL scoring rules (P1, P2, P3, P4).

time_since_recent_payment	Time Since recent Payment made
time_since_first_deliquency	Time since first Deliquency (missed payment)
time_since_recent_deliquency	Time Since recent Delinquency
num_times_delinquent	Number of times delinquent
max_delinquency_level	Maximum delinquency level
max_recent_level_of_deliq	Maximum recent level of delinquency
num_deliq_6mts	Number of times delinquent in last 6 months
num_deliq_12mts	Number of times delinquent in last 12 months
num_deliq_6_12mts	Number of times delinquent between last 6 months and last 12 months
max_deliq_6mts	Maximum delinquency level in last 6 months
max_deliq_12mts	Maximum delinquency level in last 12 months
num_times_30p_dpd	Number of times 30+ dpd
num_times_60p_dpd	Number of times 60+ dpd
num_std	Number of standard Payments
num_std_6mts	Number of standard Payments in last 6 months
num_std_12mts	Number of standard Payments in last 12 months
num_sub	Number of sub standard payments - not making full payments
num_sub_6mts	Number of sub standard payments in last 6 months
num_sub_12mts	Number of sub standard payments in last 12 months
num_dbt	Number of doubtful payments
num_dbt_6mts	Number of doubtful payments in last 6 months
num_dbt_12mts	Number of doubtful payments in last 12 months

num_lss	Number of loss accounts
num_lss_6mts	Number of loss accounts in last 6 months
num_lss_12mts	Number of loss accounts in last 12 months
recent_level_of_deliq	Recent level of delinquency
tot_enq	Total enquiries
CC_enq	Credit card enquiries
CC_enq_L6m	Credit card enquiries in last 6 months
CC_enq_L12m	Credit card enquiries in last 12 months
PL_enq	Personal Loan enquiries
PL_enq_L6m	Personal Loan enquiries in last 6 months
PL_enq_L12m	Personal Loan enquiries in last 12 months
time_since_recent_enq	Time since recent enquiry
enq_L12m	Enquiries in last 12 months
enq_L6m	Enquiries in last 6 months
enq_L3m	Enquiries in last 3 months
MARITALSTATUS	Marital Status
EDUCATION	Education level
AGE	Age
GENDER	
NETMONTHLYINCOME	
Time_With_Curr_Empr	Time with current Employer
pct_of_active_TLs_ever	Percent active accounts ever
pct_opened_TLs_L6m_of_L12m	Percent accounts opened in last 6 months to last 12 months
pct_currentBal_all_TL	Percent current balance of all accounts
CC_utilization	Credit card utilization
CC_Flag	Credit card Flag
PL_utilization	Peronal Loan utilization
PL_Flag	Personal Loan Flag
pct_PL_enq_L6m_of_L12m	Percent enquiries PL in last 6 months to last 12 months
pct_CC_enq_L6m_of_L12m	Percent enquiries CC in last 6 months to last 12 months

pct_PL_enq_L6m_of_ever	Percent enquiries PL in last 6 months to last 6 months
pct_CC_enq_L6m_of_ever	Percent enquiries CC in last 6 months to last 6 months
max_unsec_exposure_inPct	Maximum unsecured exposure in percent
HL_Flag	Housing Loan Flag
GL_Flag	Gold Loan Flag
last_prod_enq2	Lates product enquired for
first_prod_enq2	First productd enquired for
Credit_Score	Applicant's credit score
Approved_Flag	Priority levels

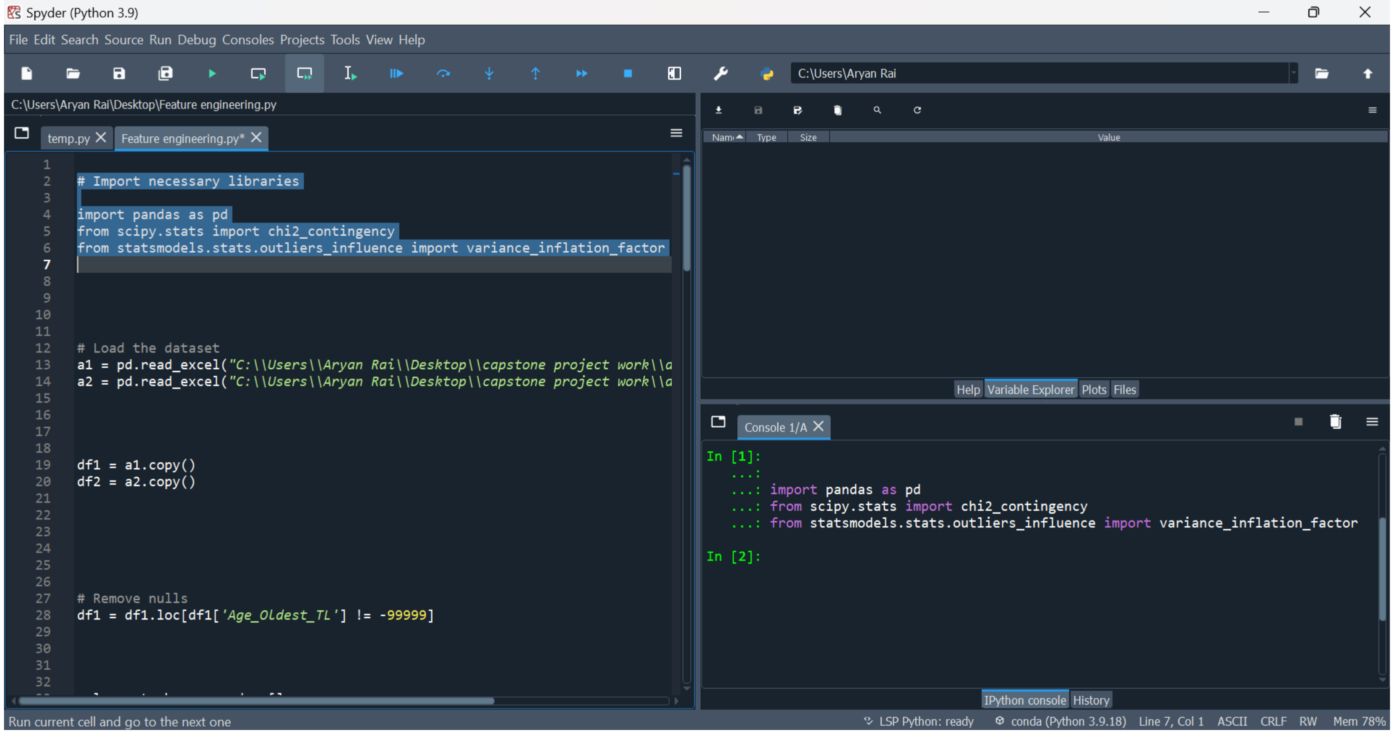
Feature Engineering & EDA

The Phase 1 of capstone project so far has focused on the initial and crucial steps of data processing, which include dataset cleaning, feature engineering, and exploratory data analysis (EDA). The main aim was to prepare the dataset for further analysis by handling missing values, identifying important features, and gaining insights into the data through visualizations and statistical methods.

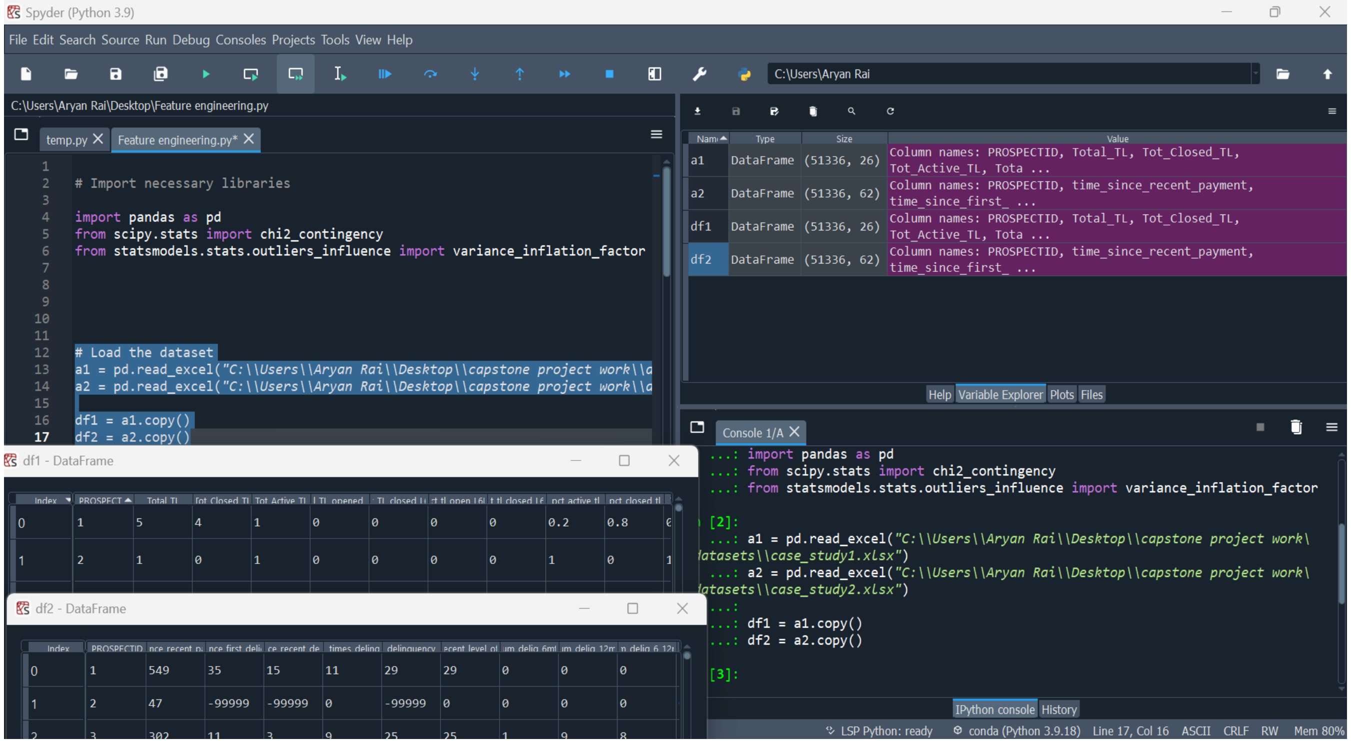
The code primarily involves:

- **Data Cleaning**: Null values, represented as -99999, were identified and appropriately handled based on the banking requirements.
- **Feature Engineering**: Multiple features were transformed and prepared for analysis, ensuring the integrity and relevance of data.
- **Exploratory Data Analysis**: Insights were drawn from both categorical and numerical features, testing associations with the target variable using chi-square tests, and multicollinearity was checked with VIF analysis.

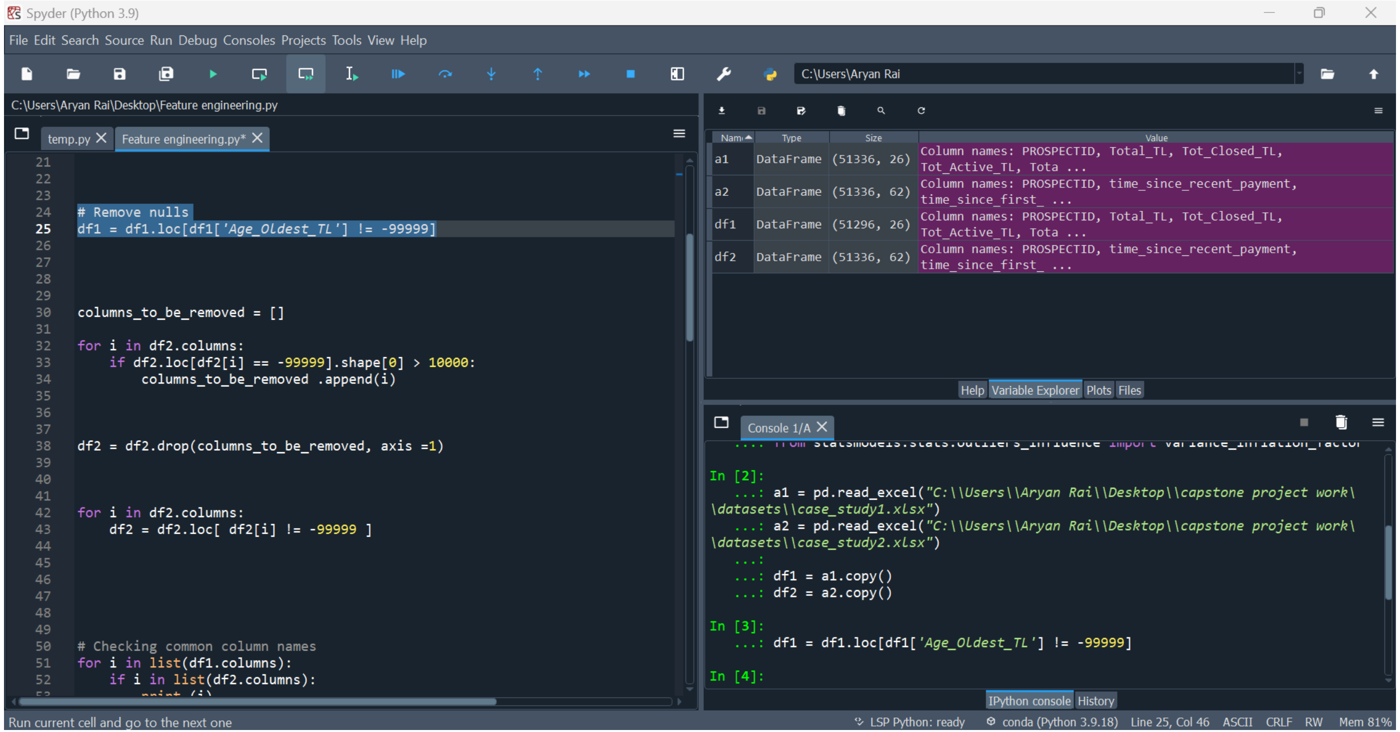
➢ The first step is to load the necessary Python libraries that help with data handling, statistical tests, and multicollinearity checks.



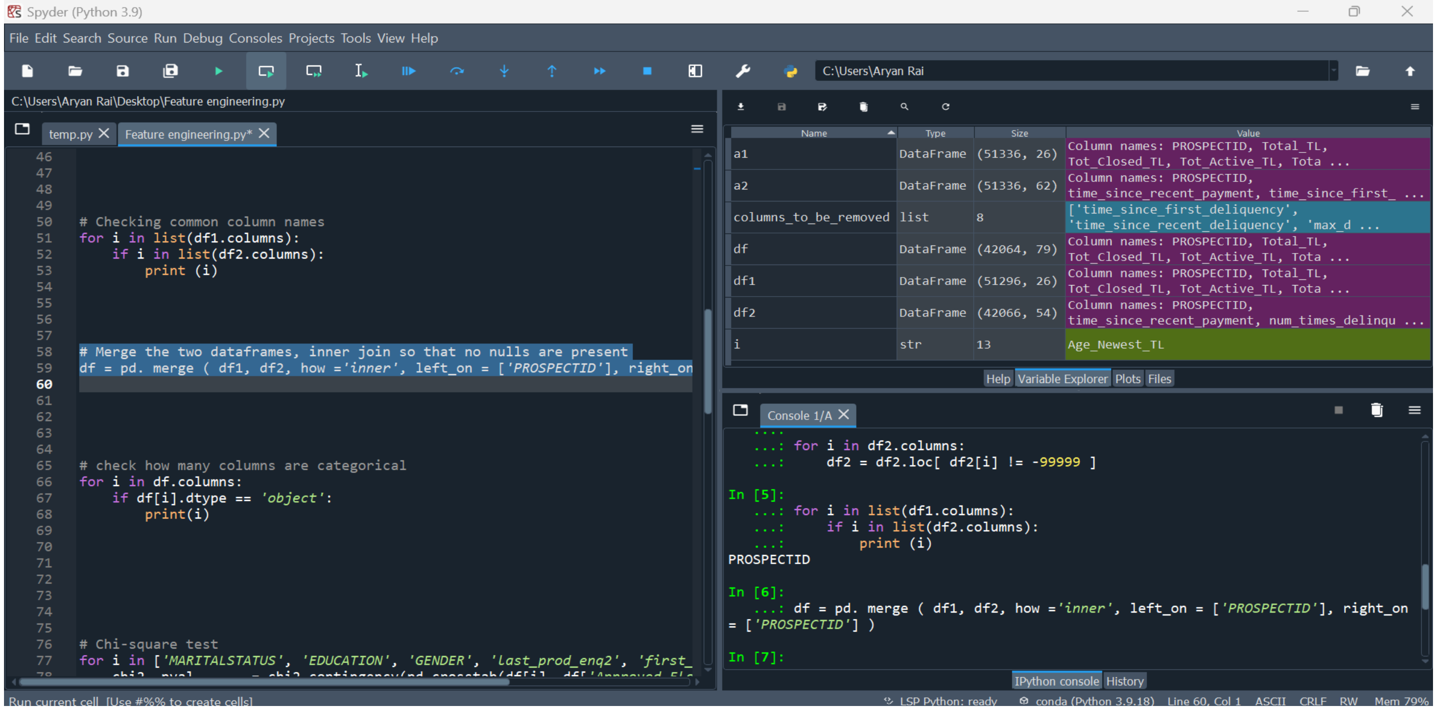
- Two Excel files, case_study1.xlsx (internal product data) and case_study2.xlsx (CIBIL data), are loaded into Python and copied into dataframes df1 and df2 for further analysis.



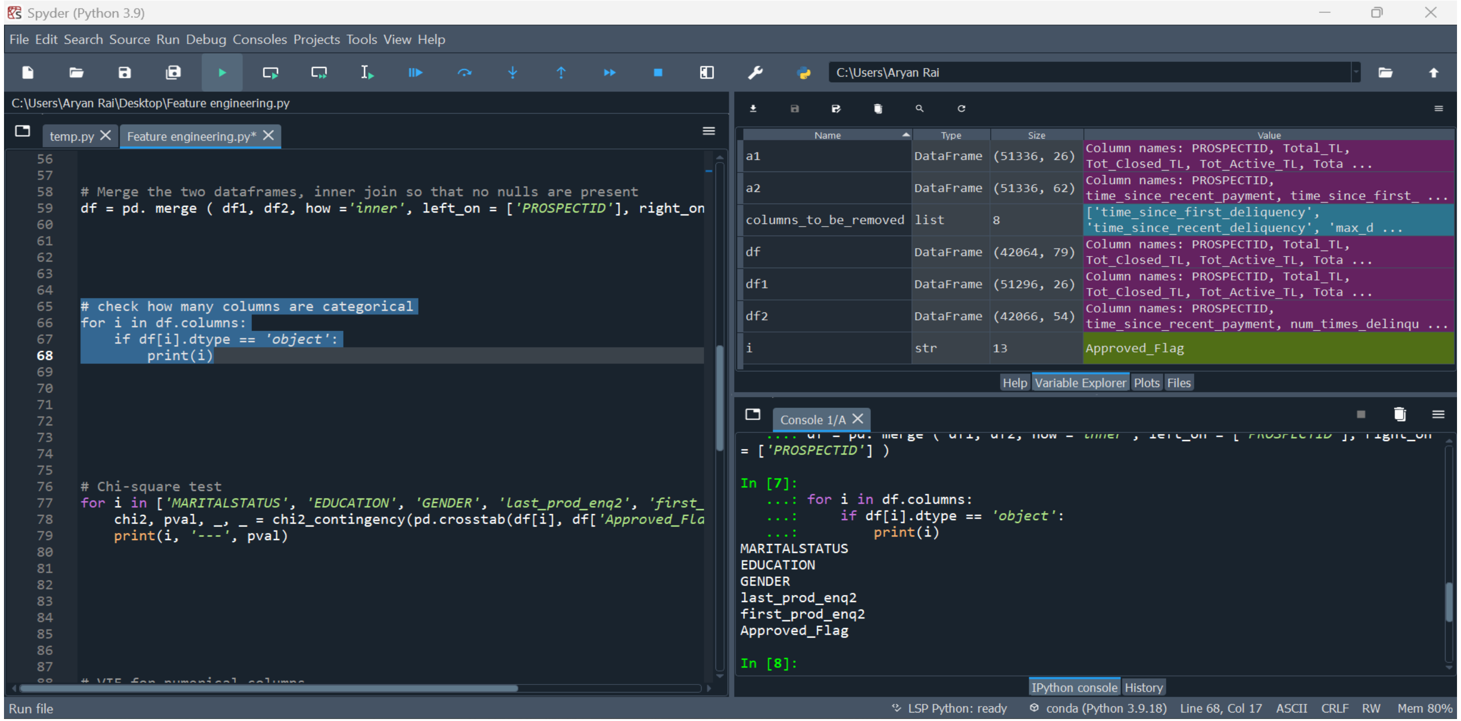
- The code identifies missing values (represented as -99999) in both datasets. In df1, rows where the oldest account age is -99999 are removed. In df2, columns with more than 10,000 missing values are dropped, and remaining rows with null values are also removed.



➢ The two datasets are merged based on a common column PROSPECTID. This ensures that we have a unified dataset with no missing values, ready for analysis.



- The code then checks which columns contain categorical data (e.g., "Marital Status"). This is important for statistical tests later.



Multicollinearity

- Happens when two or more Independent Variables are correlated
- This leads to lower the accuracy of our model
- Needs to be removed by some method (eg- VIF)

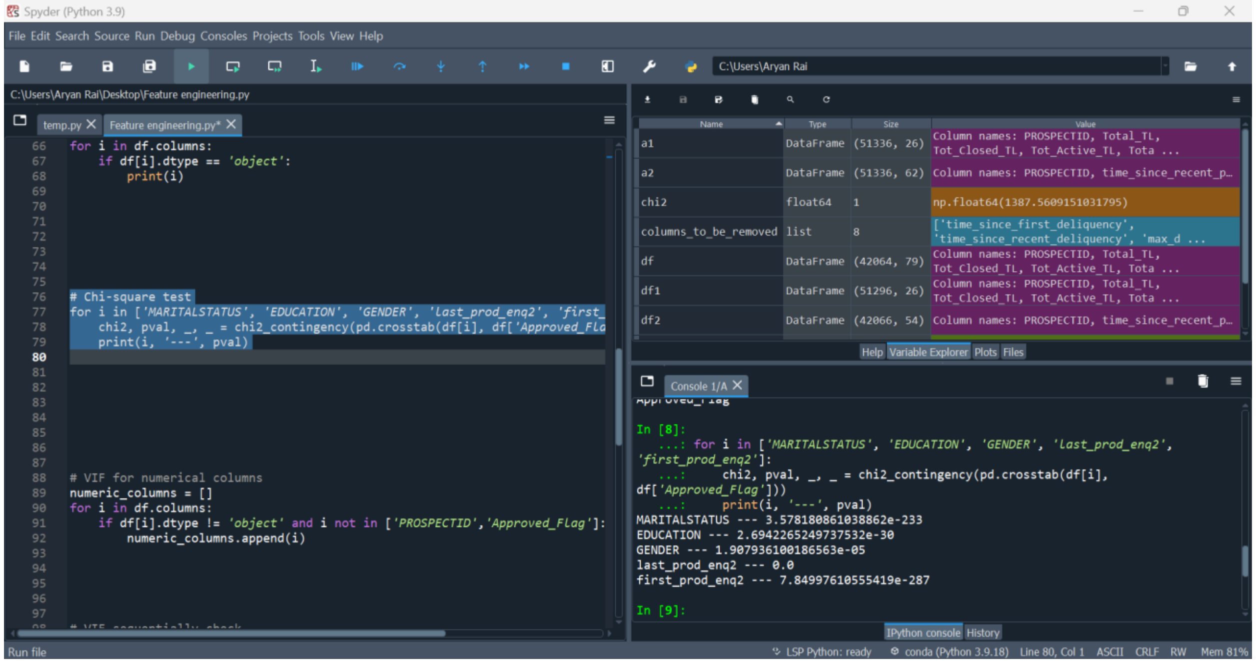
Example

House Price = (3 x Floor_Area) – (4 x Distance_City_Centre) – (8 x House_Age)

House Price = (3 x Floor_Area) – (4 x Distance_City_Centre) + (7 x Carpet_Area)

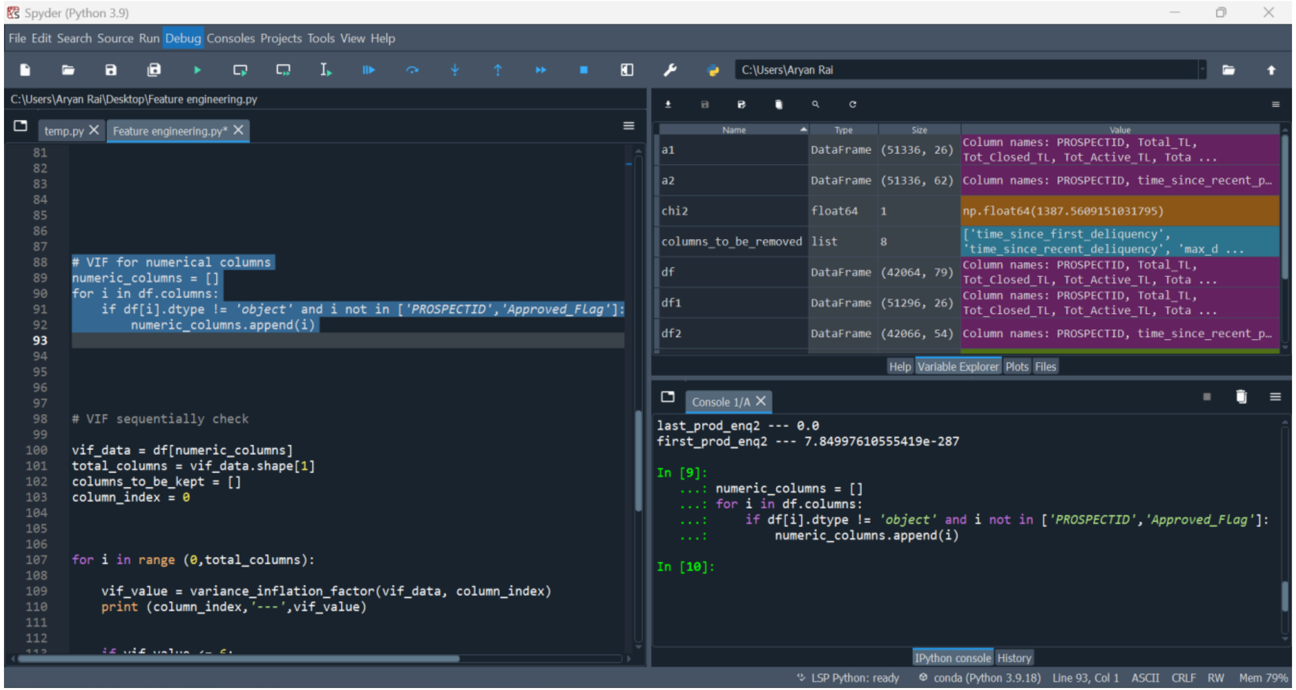
Chi-Square Test for Categorical Features:

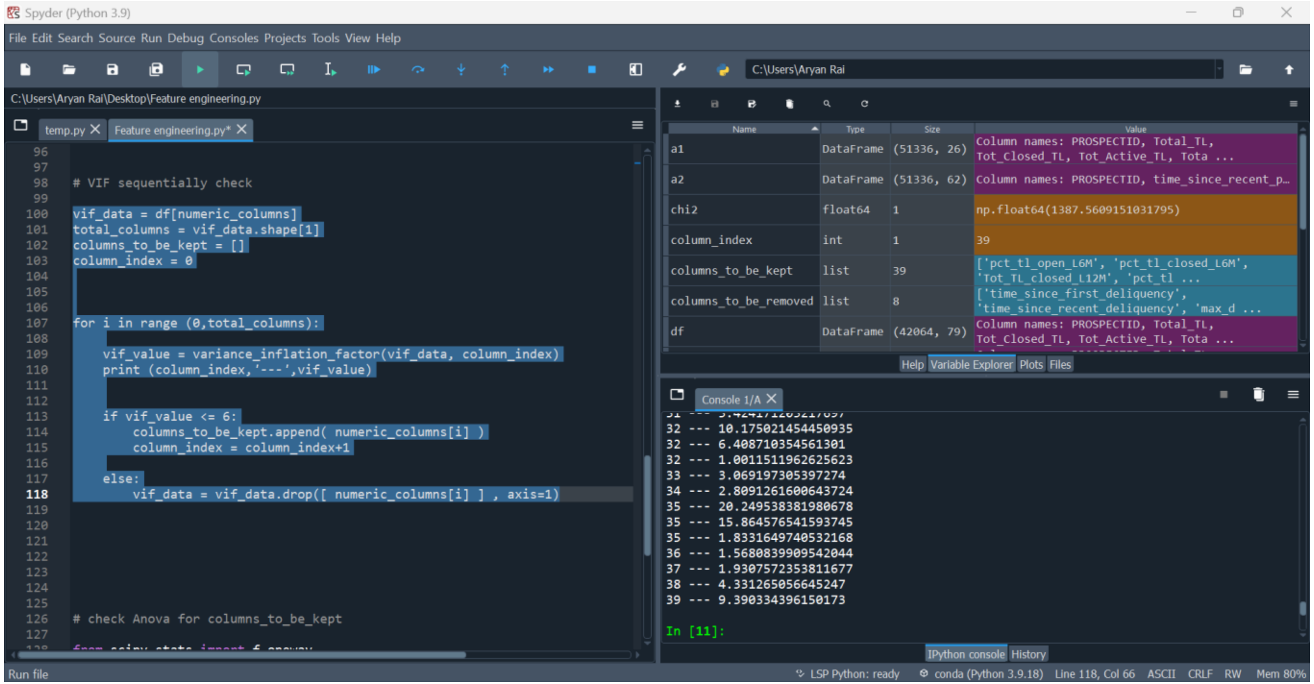
- For each categorical feature (like Marital Status, Education, Gender, etc.), a chi-square test is performed to check if it is associated with the loan approval flag (Approved_Flag). This test helps determine which categorical features have a significant relationship with the target variable.



Checking Multicollinearity with VIF:

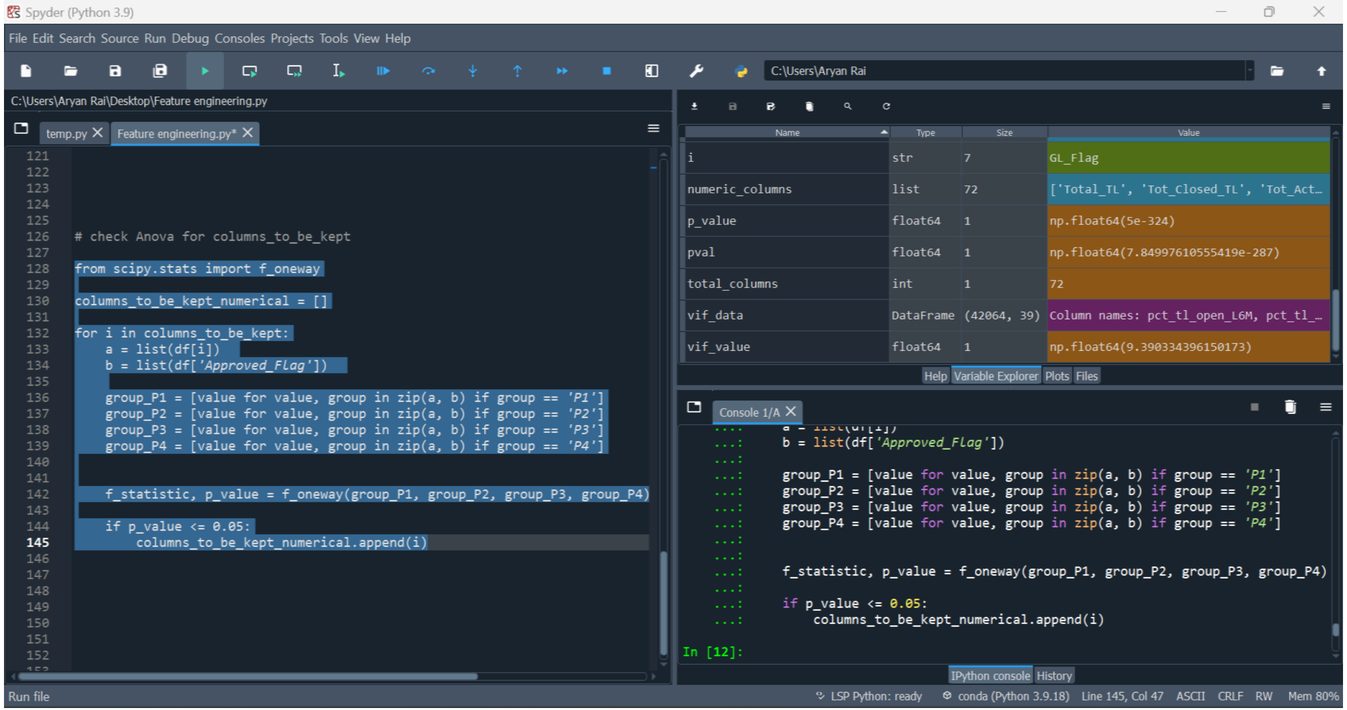
- For numerical columns, multicollinearity is checked using the Variance Inflation Factor (VIF). Multicollinearity occurs when two or more features are highly correlated, and removing such features can improve the model's accuracy. The code removes any features with a VIF value greater than 6.




Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Aryan Rai
C:\Users\Aryan Rai\Desktop\Feature engineering.py
temp.py
Feature engineering.py*
# VIF sequentially check
vif_data = df[numeric_columns]
total_columns = vif_data.shape[1]
columns_to_be_kept = []
column_index = 0
for i in range (0,total_columns):
vif_value = variance_inflation_factor(vif_data, column_index)
print (column_index,'---',vif_value)
if vif_value <= 6:
columns_to_be_kept.append(numeric_columns[i])
column_index = column_index+1
else:
vif_data = vif_data.drop([numeric_columns[i]] , axis=1)
# check Anova for columns_to_be_kept
Name
Type
Size
Value
a1
DataFrame
(51336, 26)
Column names: PROSPECTID, Total_TL, Tot_Closed_TL, Tot_Active_TL, Tota ...
a2
DataFrame
(51336, 62)
Column names: PROSPECTID, time_since_recent_p…
chi2
float64
1
np.float64(1387.5609151031795)
column_index
int
1
39
columns_to_be_kept
list
39
['pct_tl_open_L6M', 'pct_tl_closed_L6M', 'Tot_TL_closed_L12M', 'pct_tl ...
columns_to_be_removed
list
8
['time_since_first_deliquency', 'time_since_recent_deliquency', 'max_d ...
df
DataFrame
(42064, 79)
Column names: PROSPECTID, Total_TL, Tot_Closed_TL, Tot_Active_TL, Tota ...
Help
Variable Explorer
Plots
Files
Console 1/A
32 --- 10.175021454450935
32 --- 6.408710354561301
32 --- 1.0011511962625623
33 --- 3.069197305397274
34 --- 2.8091261600643724
35 --- 20.249538381980678
35 --- 15.864576541593745
35 --- 1.8331649740532168
36 --- 1.5680839909542044
37 --- 1.9307572353811677
38 --- 4.331265056645247
39 --- 9.390334396150173
In [11]:
IPython console
History
Run file
LSP Python: ready
conda (Python 3.9.18)
Line 118, Col 66
ASCII
CRLF
RW
Mem 80%

ANOVA Test for Numerical Features:

➢ Finally, the code performs an ANOVA test on numerical features to check whether they have a significant relationship with the target variable. Only features with a p-value less than or equal to 0.05 are kept for further analysis.



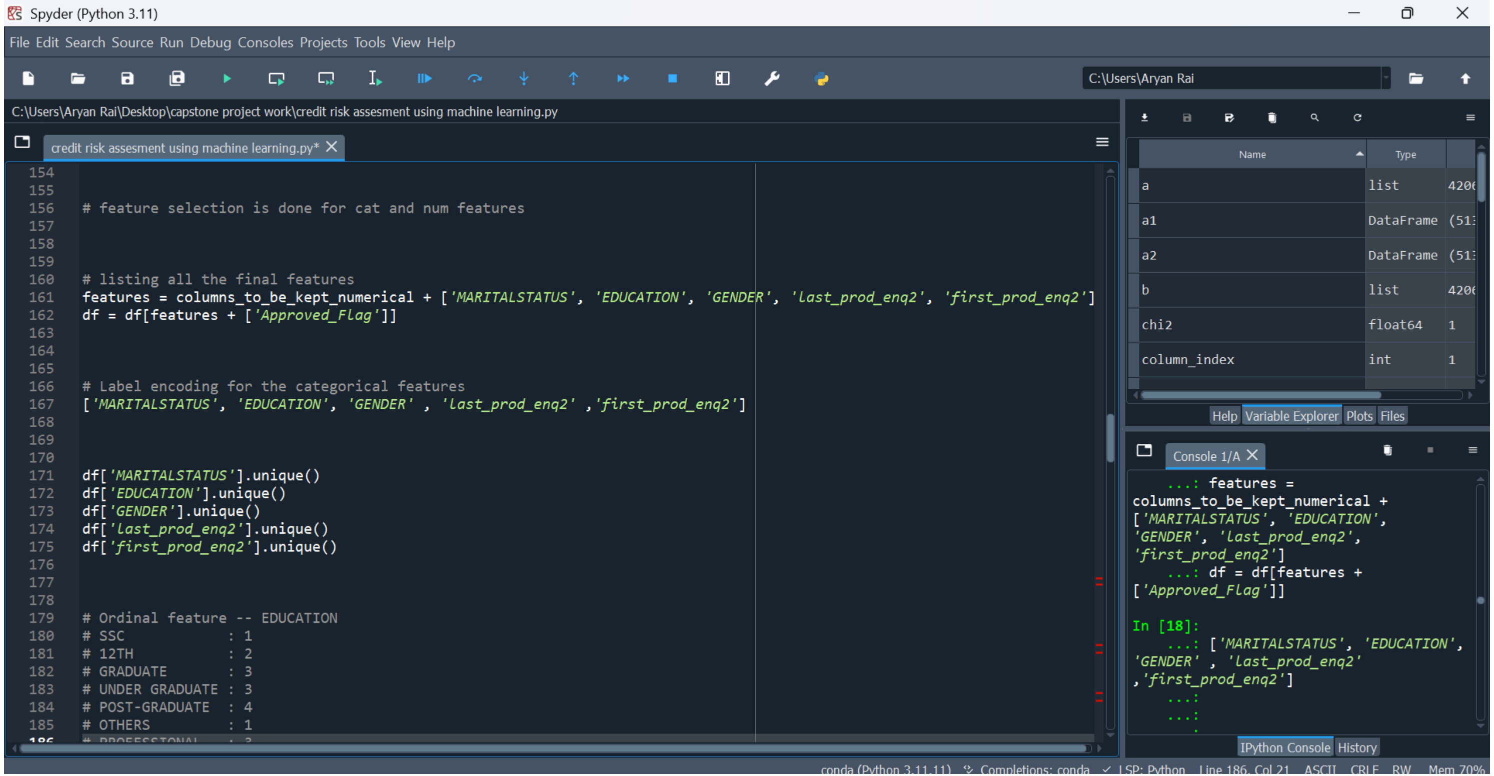
➢ Finally we have our single Dataset Ready for Modeling

df - DataFrame

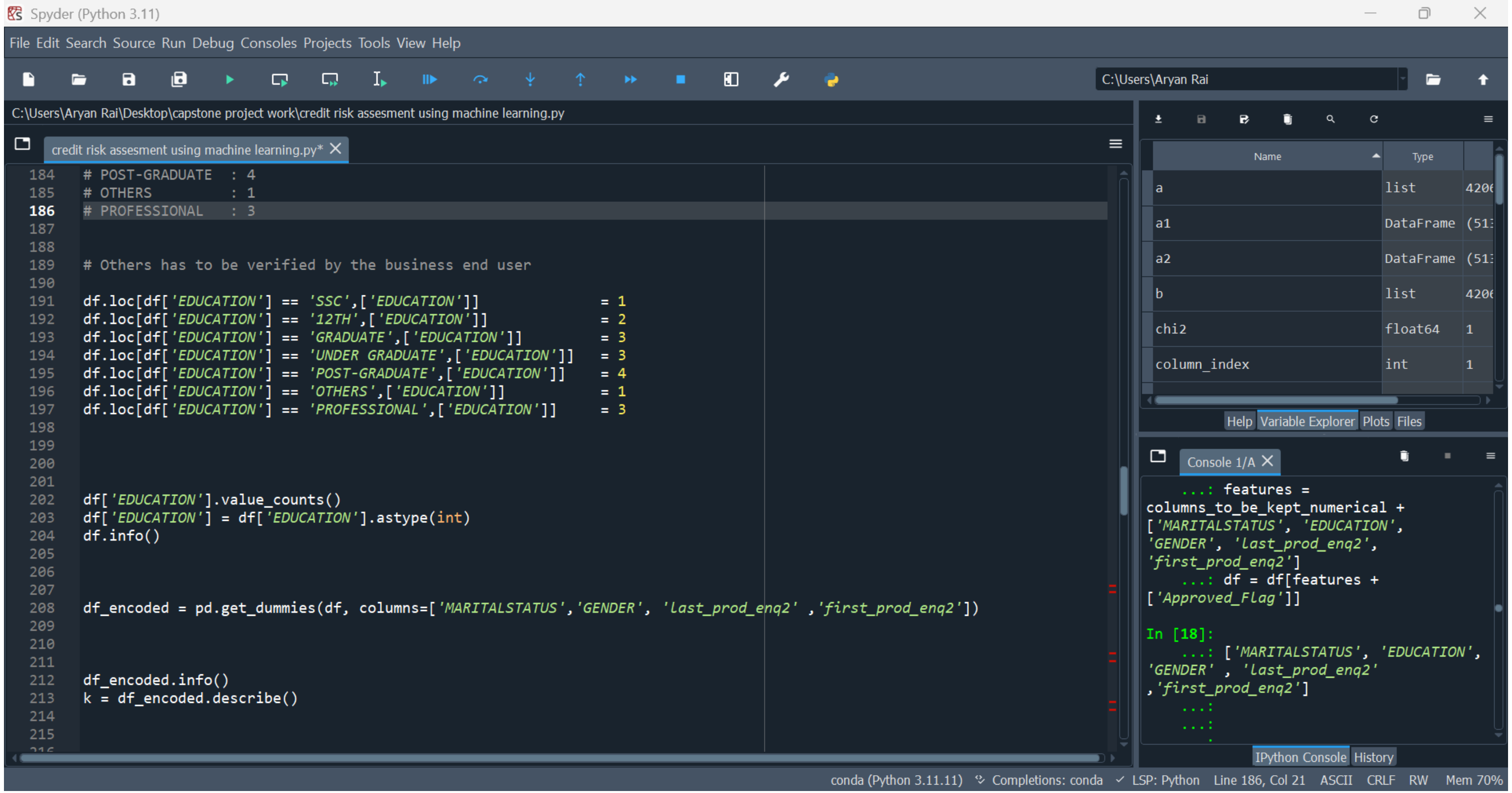
Index	PROSPECTID	Total TL	Tot Closed TL	Tot Active TL	l TL opened	TL closed L	ct tl open L6	t tl closed L6	pct active tl	pct closed tl	TL opened L	TL closed L1	t tl open L12	tl closed L1	t Missed Pm	Auto TL	CC TL	Consumer TL	Gold TL	Home TL	PL TL
0	1	5	4	1	0	0	0	0	0.2	0.8	0	0	0	0	0	0	0	0	1	0	4
1	2	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0
2	3	8	0	8	1	0	0.125	0	1	0	2	0	0.25	0	1	1	0	6	1	0	0
3	5	3	2	1	0	0	0	0	0.333	0.667	0	0	0	0	0	1	0	0	0	0	0
4	6	6	5	1	0	0	0	0	0.167	0.833	0	1	0	0.167	0	4	0	0	2	0	0
5	8	6	4	2	0	0	0	0	0.333	0.667	1	2	0.167	0.333	0	1	0	0	0	0	0
6	9	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
7	11	7	2	5	1	0	0.143	0	0.714	0.286	3	1	0.429	0.143	0	2	1	2	0	0	2
8	13	2	2	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
9	14	2	1	1	1	0	0.5	0	0.5	0.5	1	1	0.5	0.5	0	0	0	2	0	0	0
10	15	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
11	16	8	1	7	7	1	0.875	0.125	0.875	0.125	8	1	1	0.125	2	1	1	5	0	0	0
12	17	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
13	20	7	5	2	1	0	0.143	0	0.286	0.714	1	1	0.143	0.143	0	0	0	0	2	0	0
14	21	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0
15	22	5	2	3	0	1	0	0.2	0.6	0.4	2	1	0.4	0.2	1	3	0	1	0	0	1
16	23	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0
17	25	3	1	2	0	0	0	0	0.667	0.333	1	1	0.333	0.333	0	0	0	0	0	0	0
18	27	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

Format Resize Background color Column min/max Save and Close Close

- Here we have done Encoding Categorical Features:Ordinal encoding is applied to the EDUCATION feature. One-hot encoding is applied to categorical variables like MARITALSTATUS, GENDER, and others.



- Then we have described the data for checking the data type after label encoding and also we have checked that our target variable count after label encoding to check for dataset is balanced or imbalanced



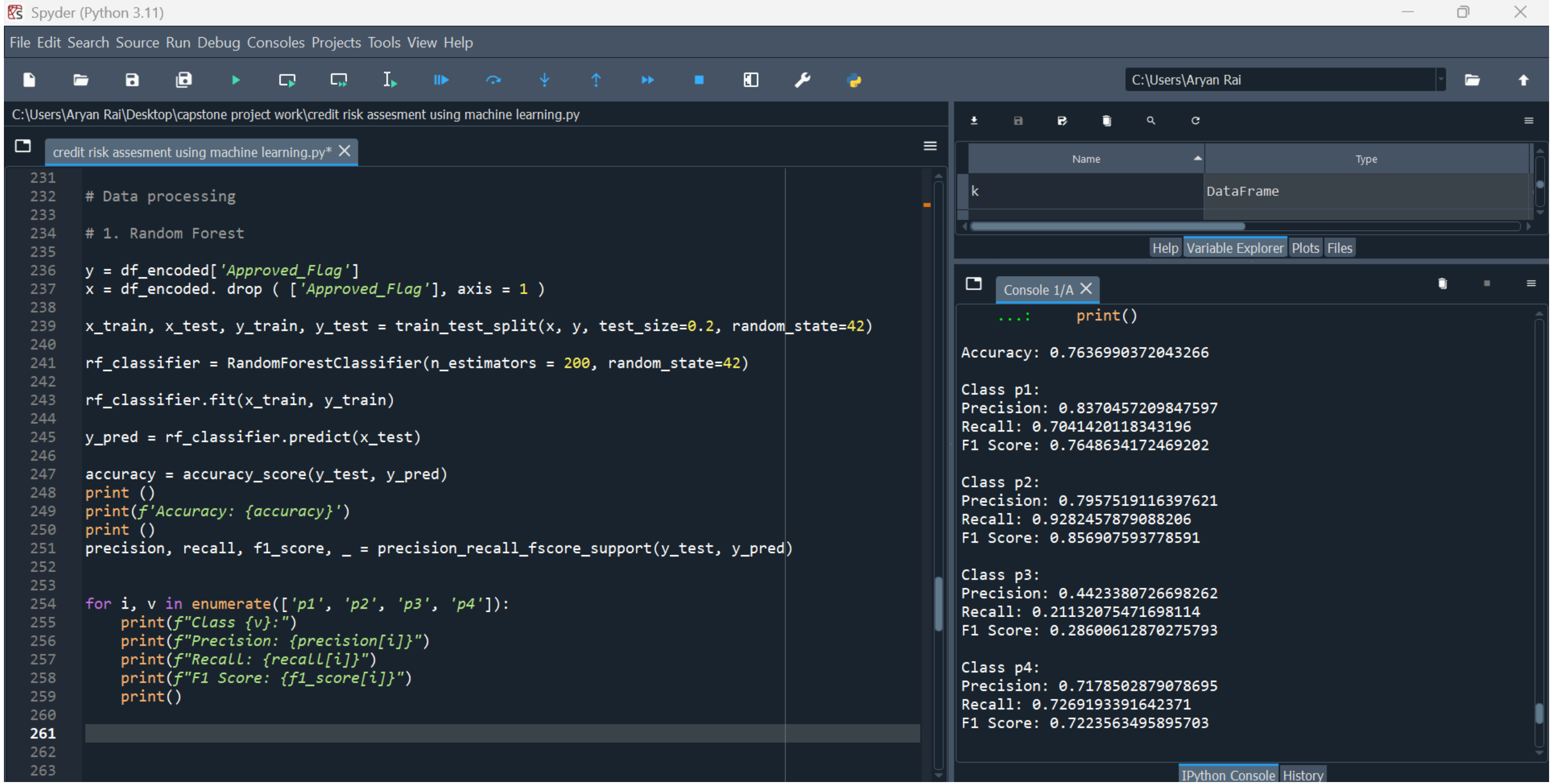
Confusion Matrix

Accuracy, Recall, Precision, F1 score

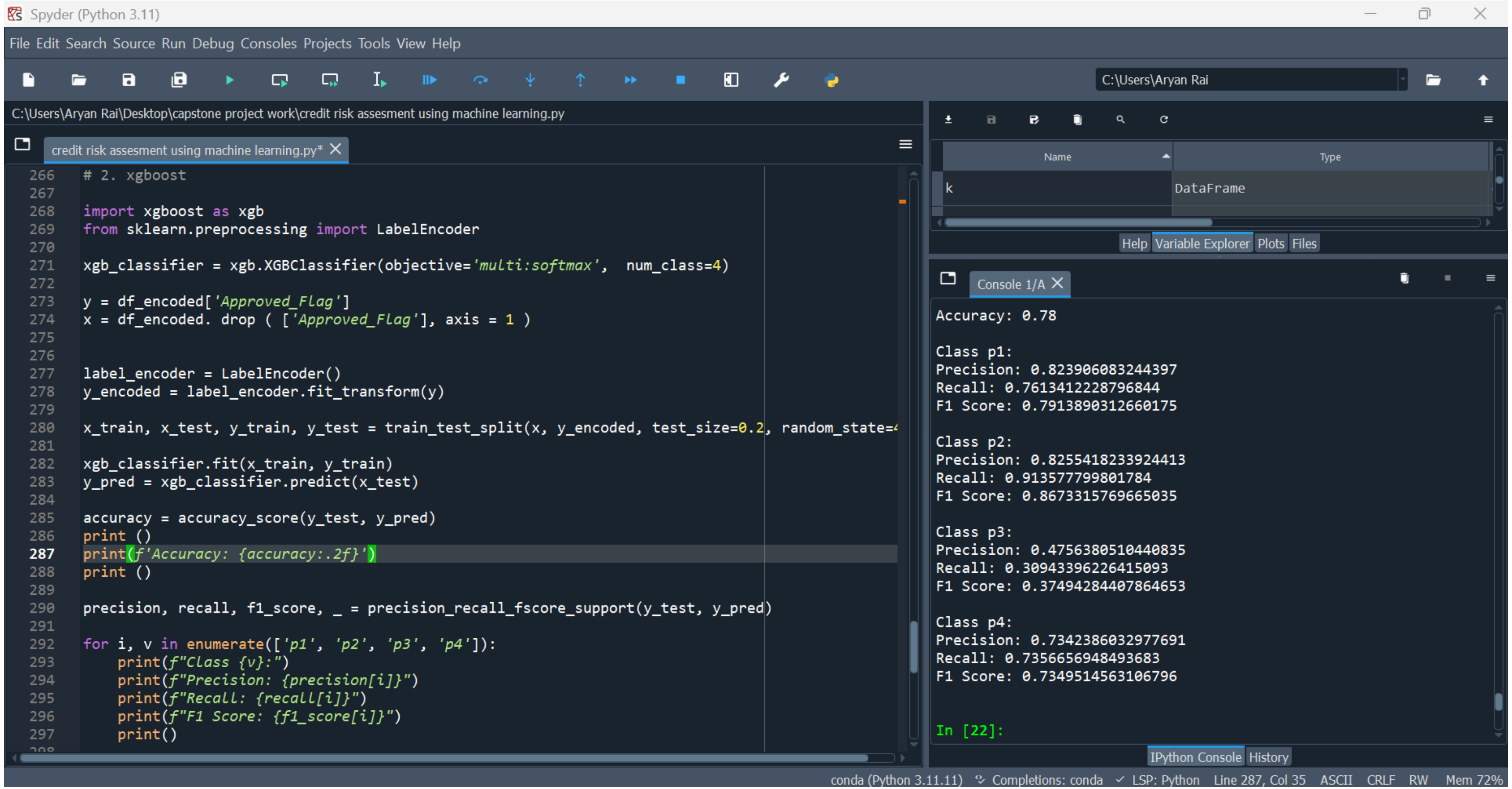
		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
Actual Class	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

$$\text{F1 score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

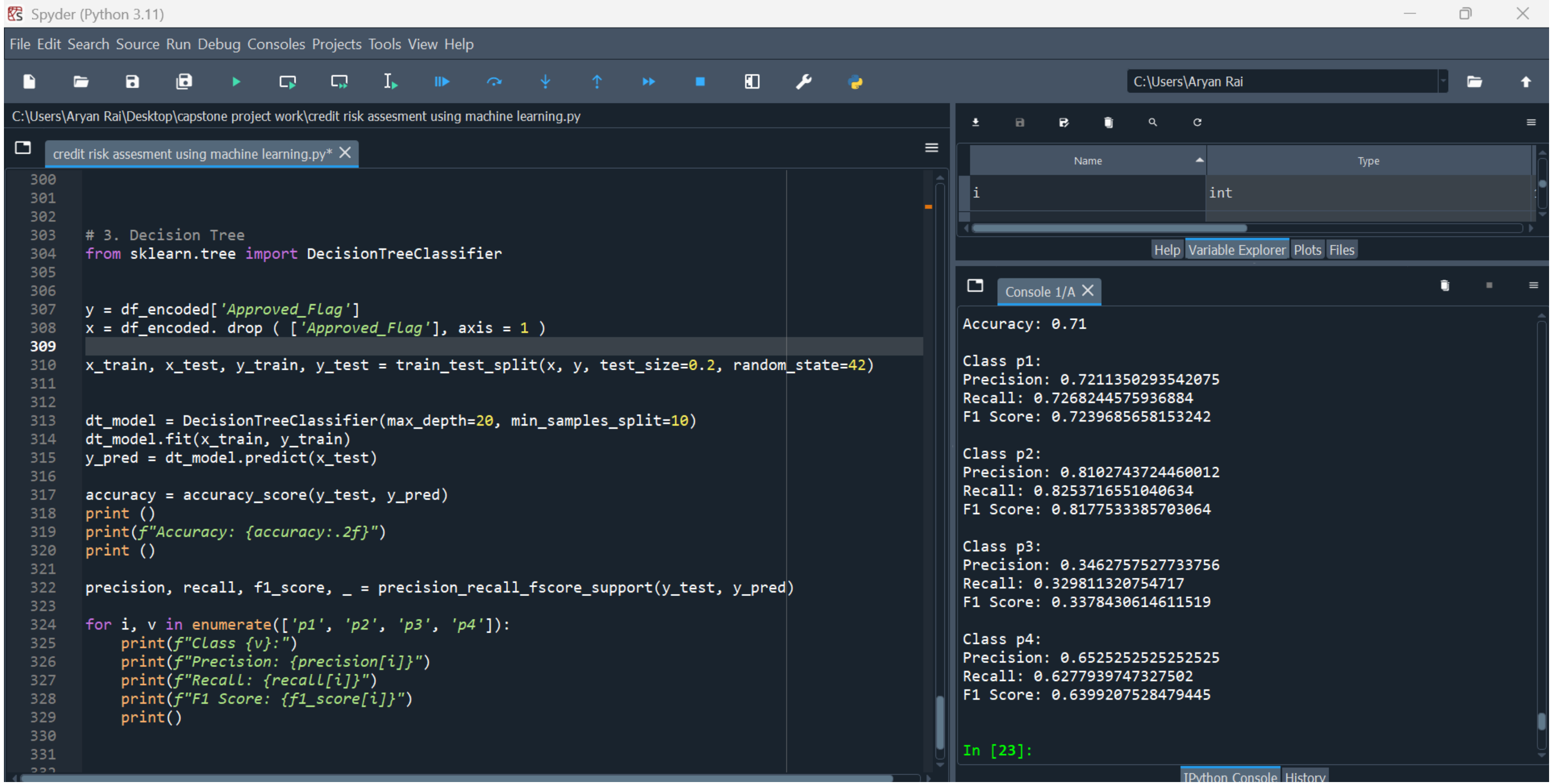
- Applied Random forest Model



- Applied Xgboost model



- Applied Decision Tree



Thank You