Assignment – 3 (**243000561**- *MD Fahim Shahriar Chowdhury*)

I am using **loandata_cleaned.csv** as my chosen dataset to answer the below questions.

Answer to the Question No-1

1. A brief summary of the data - so that I understand what the data is in general

Answer: I am using the dataset **loan data** and python to analyze the data to get a general overview of the dataset. Here is below my findings:

Dataset Overview:

1. **Shape of the Dataset**:
   - **Total Records**: 6,912 entries
   - **Features**: 8 columns
     1. **loan_status**: Binary target variable indicating loan default (1) or non-default (0).
     2. **loan_amnt**: Loan amount taken by the borrower.
     3. **int_rate**: Interest rate applied to the loan.
     4. **grade**: Loan grade based on creditworthiness (A to G).
     5. **emp_length**: Length of employment in years.
     6. **home_ownership**: Type of home ownership (RENT, MORTGAGE, OWN).
     7. **annual_inc**: Annual income of the borrower.
     8. **age**: Age of the borrower.
   - No duplicate rows present
   - **Missing Values**:
     - **Interest Rate**: 657 missing values
     - **Employment Length**: 326 missing values

2. **Preview of the Data**:

	loan_status	loan_amnt	int_rate	grade	emp_length	home_ownership	annual_inc	age
0	0	8000	6.62	A	6	MORTGAGE	38400	23
1	0	5600	10.75	B	NULL	RENT	20000	21
2	0	4000	15.23	C	16	RENT	56000	23
3	0	10000	5.79	A	2	RENT	51996	32
4	0	15000	8.49	A	3	MORTGAGE	56000	24

3. **Target Variable (Loan Status):**
   - Binary classification problem:
     - 70% of loans are non-default (0)
     - 30% of loans are default (1)

4. **Feature Analysis:**

Numerical Features:

- **Loan Amount:**
    - Range: $1,000 to $35,000
    - Mean: $9,294 Median: $7,800
- **Interest Rate:**
    - Range: 5.42% to 22.48%
    - Mean: 10.97%
- **Annual Income**:
    - Range: $4,080 to $1,900,000
    - Median: $54,000
- **Age**:
    - Range: 20 to 73 years
    - Mean: 27.6 years

Categorical Features:

- **Grade: 7 categories (A to G)**
    - Majority are A and B grades Very few F and G grades
- **Home Ownership: 3 categories**
    - **RENT**: Most common
    - **MORTGAGE**: Second most common
    - **OWN**: Least common

5. **Key Correlations and Trends:**
    - Positive correlation between loan amount and annual income (0.36)
    - Interest rate has positive correlation with loan default (0.22)
    - Negative correlation between annual income and loan default (-0.10)

	loan_status	loan_amnt	int_rate	emp_length	annual_inc	age
loan_status	1	-0.07	0.22	0	-0.1	-0.02
loan_amnt	-0.07	1	0.1	0.1	0.36	0.05
int_rate	0.22	0.1	1	-0.03	0.03	0.02
emp_length	0	0.1	-0.03	1	0.12	0.02
annual_inc	-0.1	0.36	0.03	0.12	1	0.15
age	-0.02	0.05	0.02	0.02	0.15	1

6. **Data Quality Considerations**
    - Missing values in interest rate (9.5% missing) and employment length (4.7% missing) need to be addressed
    - No duplicate records
    - Good variety in numerical ranges
    - Well-balanced categorical variables
    - Age distribution suggests a younger borrower population

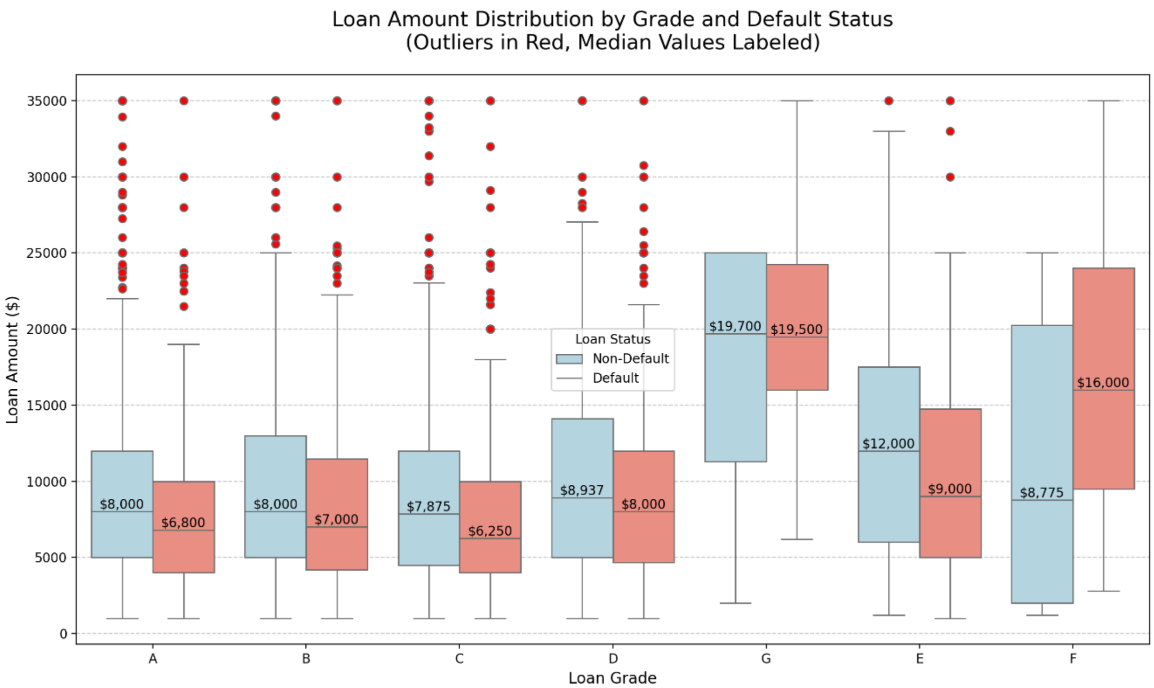
Answer to the Question No-2

2. 2/3 charts which visualize the data nicely. You are free to use whatever software for this. Just take a screenshot of the chart and put it in the document and explain what you found out.

Answer:

a) Box Plot- Loan Amount Distribution by Grade and Default Status

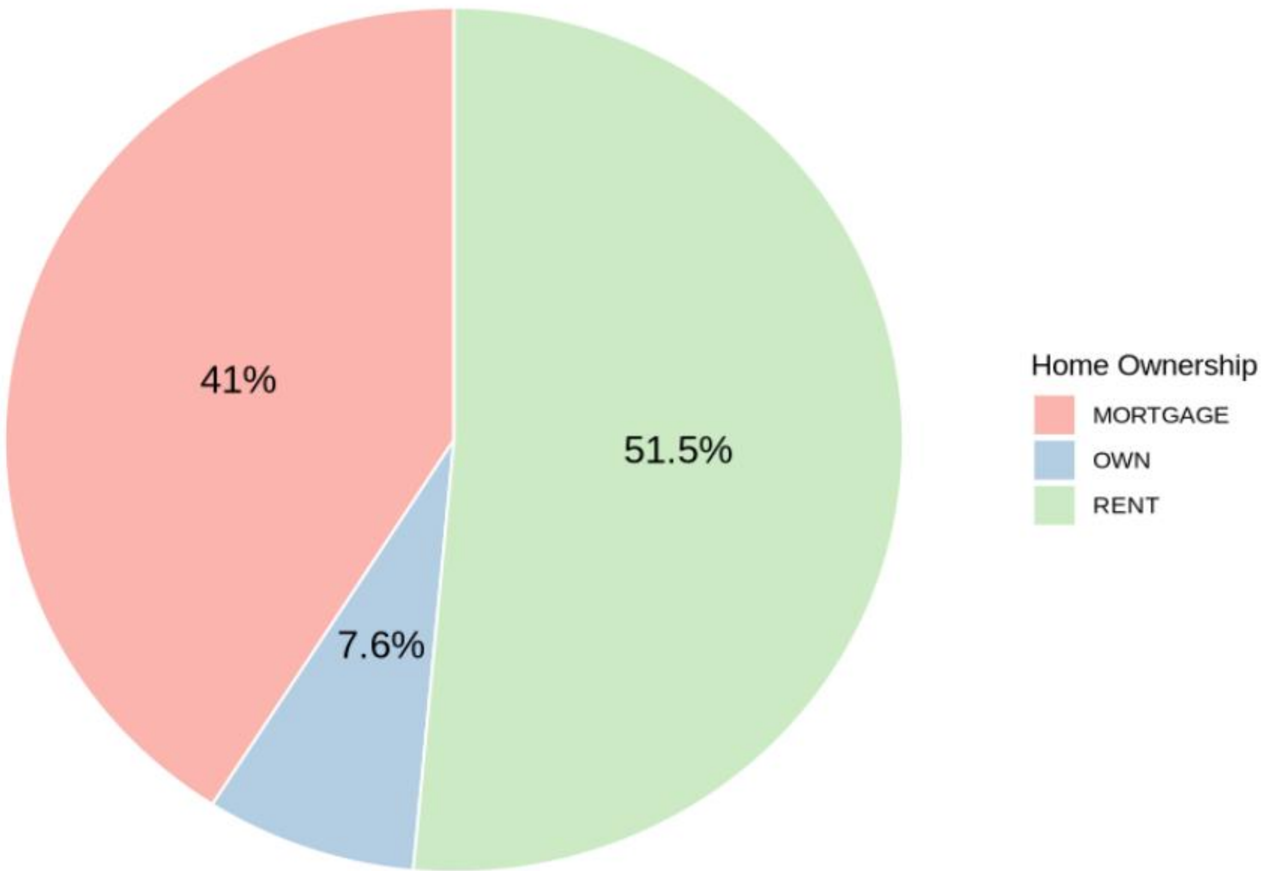
First here is my outcome from the dataset.



Key Features and Observations:

1. **Color Coding**:
   - Light blue boxes: Non-default loans
   - Salmon boxes: Default loans
   - Red dots: Outliers
2. **Data Labels**:
   - Median values are labeled for each grade and loan status combination
   - Values are shown in dollars for easy interpretation
3. **Outlier Analysis**: (showing analysis of outlier of three grade as sample)
   - **For Grade A:**
     1. Non-default: 77 outliers, maximum value $35,000.
     2. Default: 12 outliers, maximum value $35,000.
   - **For Grade B:**
     1. Non-default: 28 outliers, maximum value $35,000.
     2. Default: 27 outliers, maximum value $35,000.
   - **For Grade C:**
     1. Non-default: 39 outliers, maximum value $35,000.
     2. Default: 29 outliers, maximum value $35,000.

b) Pie Chart - Home Ownership Distribution



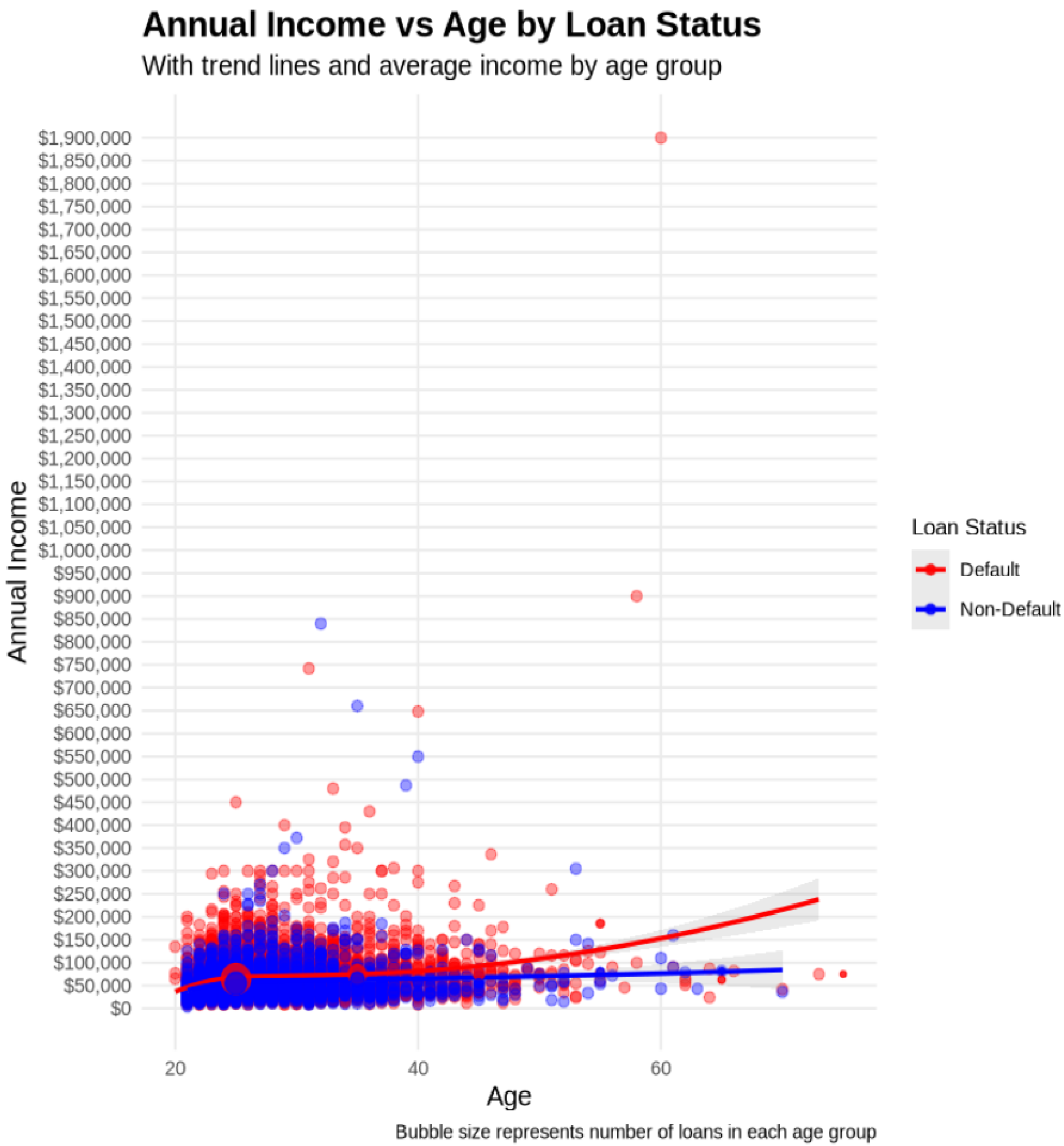
Key Observations:

- Colors: Pastel palette for better visibility
- RENT is the dominant category (~51.5%)
- MORTGAGE is second most common (~41%)
- OWN is the smallest category (~7.5%)
- Clear visualization of the proportional distribution

c) Scatter Plot - Annual Income vs Age with Loan Status

The scatter plot shows the relationship between Annual Income and Age, categorized by Loan Status (Default vs. Non-Default).
Key features include:

1. **Trend Lines**: Smoothed trend lines indicate the general relationship between age and income for each loan status group.
2. **Bubble Points**: Larger bubbles represent the average income for specific age groups, with bubble size proportional to the number of loans in that group.
3. **Color Coding**: Red represents defaults, while blue represents non-defaults, making it easy to distinguish patterns.



Summary Statistics:

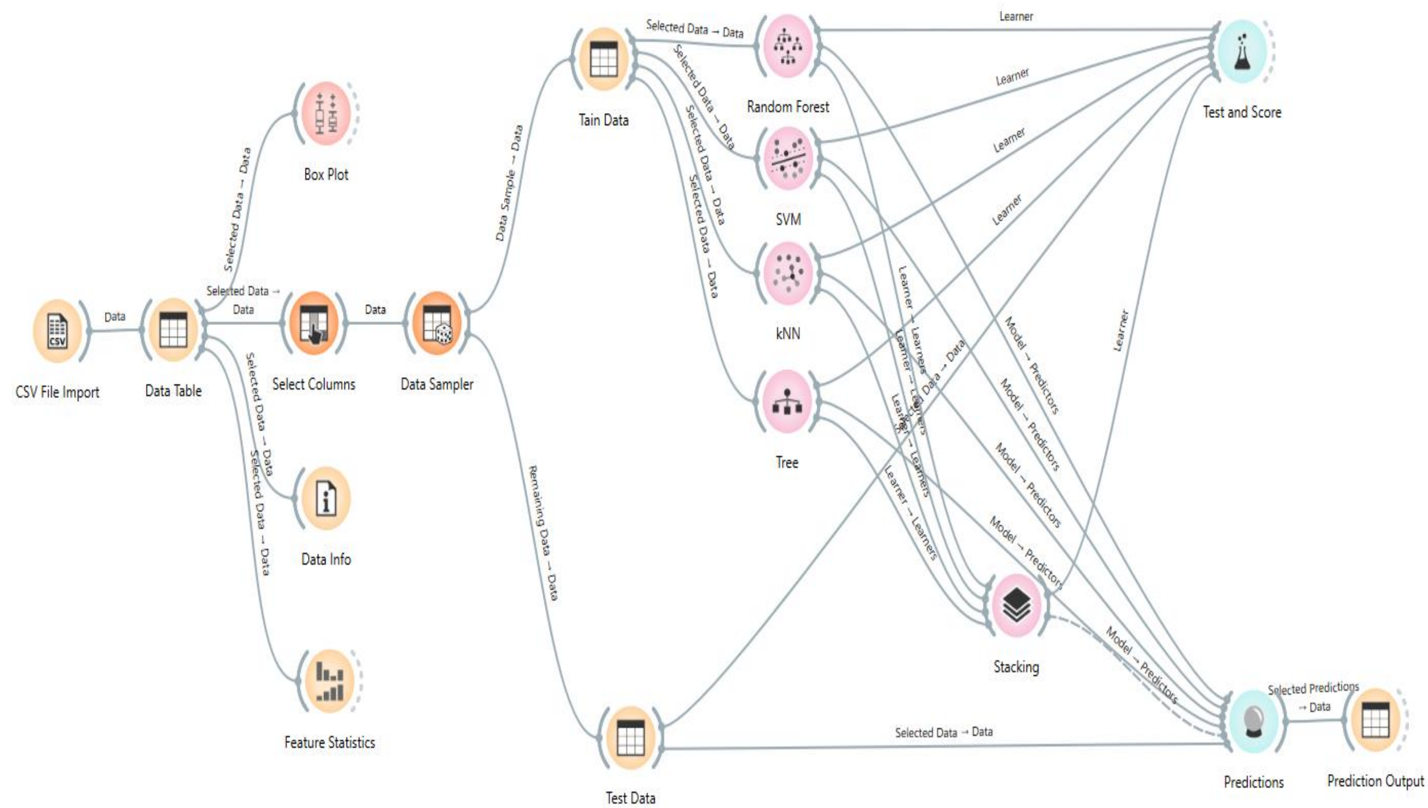
- **Average Age**: Borrowers who default tend to have a slightly lower average age.
- **Average Income**: Non-default borrowers have a higher average income
- **Median Income**: The median income for non-defaults ($57,500) is also higher than for defaults ($48,000).
- **Loan Count**: There are significantly more non-default loans (4,841) compared to defaults (2,071).

Answer to the Question No-3

3. Build a model in OrangeML, and show me the screenshot of the model and the Test and Evaluate result - showing me that you have tried out at least 3 models, tested their accuracy and then tell me which is the best model for this data.

Answer:

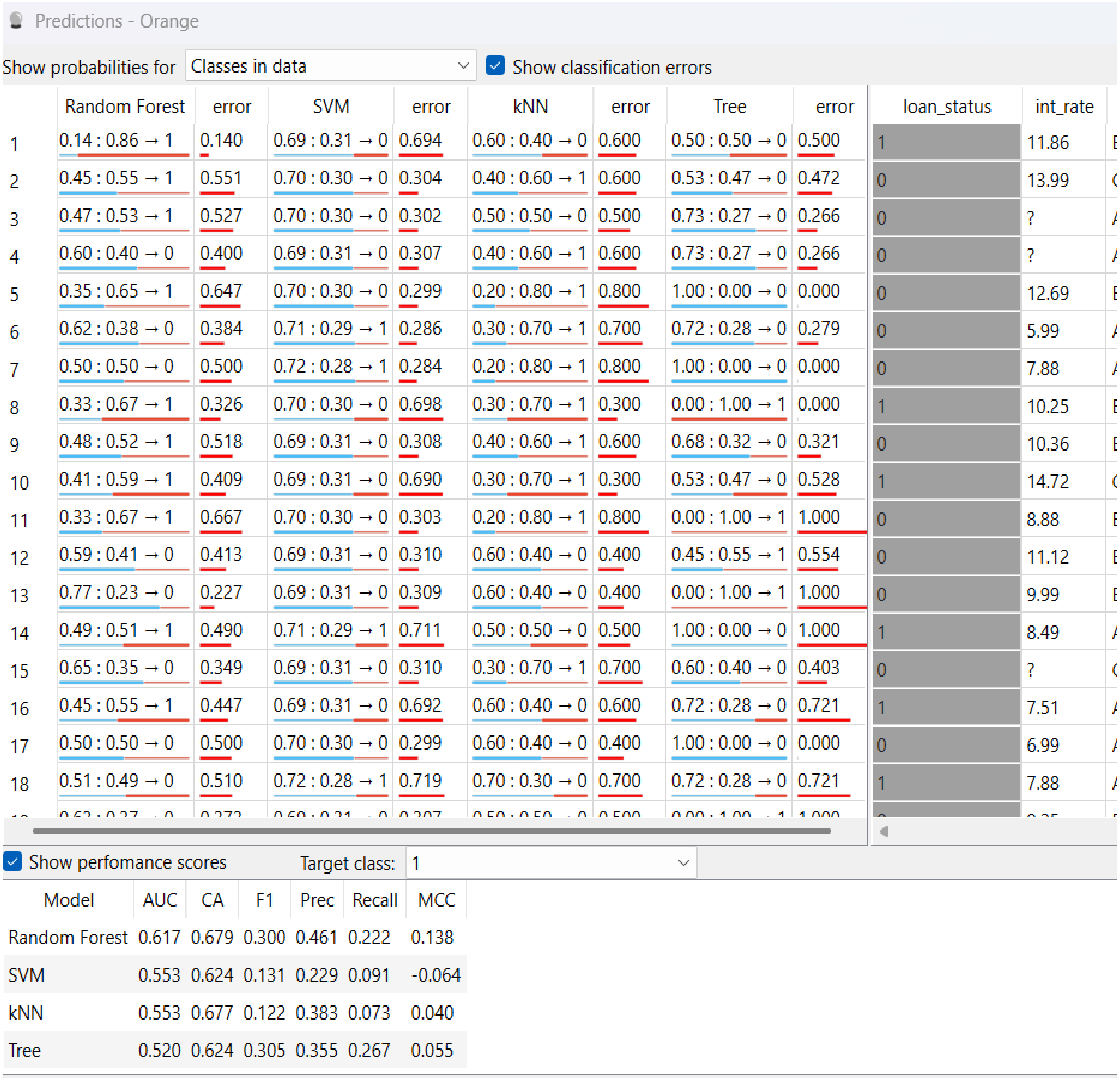
Here is my Orange ML diagram Snapshot:



Model Performance Analysis:

1. Random Forest:

- **Performance:**
  - AUC: **0.581**, Accuracy: **0.673**, F1: **0.644**
  - Precision: **0.638**, Recall: **0.673**, MCC: **0.143**, LogLoss: **0.944**
- **Analysis:**
  - Random Forest handles non-linear relationships and is robust to outliers, making it suitable for categorical classification.
  - It performed reasonably well, but its Recall and MCC indicate some misclassification issues, especially with minority classes.
  - Its **LogLoss (0.944)** is relatively high, indicating poor probability calibration, and it might struggle with imbalanced data.



Predictions - Orange

Show probabilities for: Classes in data — Show classification errors

	Random Forest	error	SVM	error	kNN	error	Tree	error	loan_status	int_rate
1	0.14 : 0.86 → 1	0.140	0.69 : 0.31 → 0	0.694	0.60 : 0.40 → 0	0.600	0.50 : 0.50 → 0	0.500	1	11.86
2	0.45 : 0.55 → 1	0.551	0.70 : 0.30 → 0	0.304	0.40 : 0.60 → 1	0.600	0.53 : 0.47 → 0	0.472	0	13.99
3	0.47 : 0.53 → 1	0.527	0.70 : 0.30 → 0	0.302	0.50 : 0.50 → 0	0.500	0.73 : 0.27 → 0	0.266	0	?
4	0.60 : 0.40 → 0	0.400	0.69 : 0.31 → 0	0.307	0.40 : 0.60 → 1	0.600	0.73 : 0.27 → 0	0.266	0	?
5	0.35 : 0.65 → 1	0.647	0.70 : 0.30 → 0	0.299	0.20 : 0.80 → 1	0.800	1.00 : 0.00 → 0	0.000	0	12.69
6	0.62 : 0.38 → 0	0.384	0.71 : 0.29 → 1	0.286	0.30 : 0.70 → 1	0.700	0.72 : 0.28 → 0	0.279	0	5.99
7	0.50 : 0.50 → 0	0.500	0.72 : 0.28 → 1	0.284	0.20 : 0.80 → 1	0.800	1.00 : 0.00 → 0	0.000	0	7.88
8	0.33 : 0.67 → 1	0.326	0.70 : 0.30 → 0	0.698	0.30 : 0.70 → 1	0.300	0.00 : 1.00 → 1	0.000	1	10.25
9	0.48 : 0.52 → 1	0.518	0.69 : 0.31 → 0	0.308	0.40 : 0.60 → 1	0.600	0.68 : 0.32 → 0	0.321	0	10.36
10	0.41 : 0.59 → 1	0.409	0.69 : 0.31 → 0	0.690	0.30 : 0.70 → 1	0.300	0.53 : 0.47 → 0	0.528	1	14.72
11	0.33 : 0.67 → 1	0.667	0.70 : 0.30 → 0	0.303	0.20 : 0.80 → 1	0.800	0.00 : 1.00 → 1	1.000	0	8.88
12	0.59 : 0.41 → 0	0.413	0.69 : 0.31 → 0	0.310	0.60 : 0.40 → 0	0.400	0.45 : 0.55 → 1	0.554	0	11.12
13	0.77 : 0.23 → 0	0.227	0.69 : 0.31 → 0	0.309	0.60 : 0.40 → 0	0.400	0.00 : 1.00 → 1	1.000	0	9.99
14	0.49 : 0.51 → 1	0.490	0.71 : 0.29 → 1	0.711	0.50 : 0.50 → 0	0.500	1.00 : 0.00 → 0	1.000	1	8.49
15	0.65 : 0.35 → 0	0.349	0.69 : 0.31 → 0	0.310	0.30 : 0.70 → 1	0.700	0.60 : 0.40 → 0	0.403	0	?
16	0.45 : 0.55 → 1	0.447	0.69 : 0.31 → 0	0.692	0.60 : 0.40 → 0	0.600	0.72 : 0.28 → 0	0.721	1	7.51
17	0.50 : 0.50 → 0	0.500	0.70 : 0.30 → 0	0.299	0.60 : 0.40 → 0	0.400	1.00 : 0.00 → 0	0.000	0	6.99
18	0.51 : 0.49 → 0	0.510	0.72 : 0.28 → 1	0.719	0.70 : 0.30 → 0	0.700	0.72 : 0.28 → 0	0.721	1	7.88

Show perfomance scores — Target class: 1

Model	AUC	CA	F1	Prec	Recall	MCC
Random Forest	0.617	0.679	0.300	0.461	0.222	0.138
SVM	0.553	0.624	0.131	0.229	0.091	-0.064
kNN	0.553	0.677	0.122	0.383	0.073	0.040
Tree	0.520	0.624	0.305	0.355	0.267	0.055

2. Support Vector Machine (SVM):

- **Performance:**
  - AUC: **0.508**, Accuracy: **0.619**, F1: **0.586**
  - Precision: **0.569**, Recall: **0.619**, MCC: **-0.008**, LogLoss: **0.621**
- **Analysis:**
  - SVM struggles with multi-class classification and imbalanced data without proper kernel tuning.
  - Low AUC and MCC (**-0.008**) indicate poor discrimination ability, and negative MCC highlights potential prediction bias.
  - LogLoss is acceptable but fails to outperform ensemble methods.

3. k-Nearest Neighbors (kNN):

- **Performance:**
  - AUC: **0.489**, Accuracy: **0.668**, F1: **0.571**
  - Precision: **0.550**, Recall: **0.668**, MCC: **-0.031**, LogLoss: **1.017**
- **Analysis:**
  - kNN relies on distance metrics, making it sensitive to scale and high-dimensional data.
  - Low MCC (**-0.031**) and high LogLoss indicate misclassification issues and unreliable probability estimates.
  - Its recall (**0.668**) is moderate, but precision is low, suggesting misclassification of negative cases.

4. Decision Tree (Tree):

- **Performance:**
  - AUC: **0.544**, Accuracy: **0.632**, F1: **0.614**
  - Precision: **0.604**, Recall: **0.632**, MCC: **0.071**, LogLoss: **9.248**
- **Analysis:**
  - Decision Trees tend to overfit, especially when the data is noisy or lacks sufficient samples per class.
  - It has moderate accuracy and recall but a **very high LogLoss (9.248)**, indicating poor probability calibration.
  - Simplicity in interpretability is its main advantage, but it does not generalize well for this dataset.
- **Why Not Chosen?**
  - Extremely high LogLoss indicates poor probabilistic predictions.
  - Overfitting and instability make it unreliable for production use.

5. Stacking Model (Stack):

- **Performance:**
  - AUC: **0.592**, Accuracy: **0.687**, F1: **0.573**
  - Precision: **0.590**, Recall: **0.687**, MCC: **0.015**, LogLoss: **0.609**
- **Analysis:**
  - Stacking combines predictions from multiple models, leveraging their strengths while minimizing weaknesses.
  - It has the **highest AUC (0.592)** and **best Recall (0.687)**, crucial for identifying positive cases in classification tasks.
  - Low LogLoss (**0.609**) reflects better probability calibration compared to others.

Evaluation results for target (None, show average over classes)

Model	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
Random Forest	0.581	0.673	0.644	0.638	0.673	0.143	0.446	0.944
SVM	0.508	0.619	0.586	0.569	0.619	-0.008	0.374	0.621
kNN	0.489	0.668	0.571	0.550	0.668	-0.031	0.318	1.017
Tree	0.544	0.632	0.614	0.604	0.632	0.071	0.433	9.248
Stack	0.592	0.687	0.573	0.590	0.687	0.015	0.317	0.609

Why Not Use MAE, R^2, or MAPE?
These metrics are suitable for **regression tasks**, not **classification tasks**. Here's why:

1. **MAE (Mean Absolute Error):**
   - Measures the average magnitude of prediction errors.
   - Cannot evaluate categorical outcomes, as it assumes continuous numeric values.
2. **R^2 (R-Squared):**
   - Measures variance explained by the model.
   - Inapplicable for categorical variables since it assumes linear relationships between continuous features.
3. **MAPE (Mean Absolute Percentage Error):**
   - Evaluates percentage errors in predictions.
   - Makes no sense for categorical predictions, which are about class labels rather than magnitudes.

For categorical classification, **metrics like AUC, F1 Score, Precision, Recall, MCC, and LogLoss** are preferred because they assess the model's ability to predict discrete class labels accurately, handle imbalances, and calibrate probabilities.

Why Use the Stack Model?
The **Stack Model** outperforms others because it combines multiple models' strengths, improving prediction accuracy and generalization. It has:

- **Highest AUC (0.592)** – Better class separation.
- **Best Recall (0.687)** – Captures more positive cases.
- **Lowest LogLoss (0.609)** – Better probability calibration.

Why Not Other Models that I have tested?

- **Random Forest:** Moderate performance but higher LogLoss (0.944) and risk of overfitting.
- **SVM:** Low AUC (0.508) and MCC (-0.008), indicating poor discrimination and imbalance handling.
- **kNN:** High LogLoss (1.017) and sensitivity to scaling, leading to unreliable predictions.
- **Decision Tree:** High LogLoss (9.248) and overfitting issues.

Conclusion:
The **Stack Model** is more robust, reduces overfitting, and delivers the most balanced results for this **categorical classification task**.

Final Recommendation:

Based on performance metrics, the **Stacking Model (Stack)** should be selected for this dataset due to its better generalization and ability to handle classification problems effectively.