Presentation by **Anggraeni kusuma Dewi**

CREDIT SCORE CLASSIFICATION

Dibimbing.id ✦ Data Science ✦ Batch 17

Contents

- Background
- Data Preprocessing
- Baseline Model
- Improvement Model
- Final Evaluation
- Conclusion

Background

- The purpose of this study is to determine the creditworthiness of the customer.

- You are working as a data scientist in a global finance company. Over the years, the company has collected basic bank details and gathered a lot of credit-related information. The management wants to build an intelligent system to segregate the people into credit score brackets to reduce the manual efforts.

Data Preprocessing

Dataset Information :

https://www.kaggle.com/datasets/parisrohan/credit-score-classification

The dataset consist of :

100.000 rows

The features in the dataset :

1. ID: Unique ID of the record
2. Customer_ID: Unique ID of the customer
3. Month: Month of the year
4. Name: The name of the person
5. Age: The age of the person
6. SSN: Social Security Number of the person
7. Occupation: The occupation of the person
8. Annual_Income: The Annual Income of the person
9. Monthly_Inhand_Salary: Monthly in-hand salary of the person
10. Num_Bank_Accounts: The number of bank accounts of the person
11. Num_Credit_Card: Number of credit cards the person is having
12. Interest_Rate: The interest rate on the credit card of the person
13. Num_of_Loan: The number of loans taken by the person from the bank
14. Type_of_Loan: The types of loans taken by the person from the bank
15. Delay_from_due_date: The average number of days delayed by the person from the date of payment
16. Num_of_Delayed_Payment: Number of payments delayed by the person
17. Changed_Credit_Card: The percentage change in the credit card limit of the person
18. Num_Credit_Inquiries: The number of credit card inquiries by the person
19. Credit_Mix: Classification of Credit Mix of the customer
20. Outstanding_Debt: The outstanding balance of the person
21. Credit_Utilization_Ratio: The credit utilization ratio of the credit card of the customer
22. Credit_History_Age: The age of the credit history of the person
23. Payment_of_Min_Amount: Yes if the person paid the minimum amount to be paid only, otherwise no.
24. Total_EMI_per_month: The total EMI per month of the person
25. Amount_invested_monthly: The monthly amount invested by the person
26. Payment_Behaviour: The payment behaviour of the person
27. Monthly_Balance: The monthly balance left in the account of the person
28. Credit_Score: The credit score of the person

Baseline Data Information

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 28 columns):
 #   Column                    Non-Null Count   Dtype  
---  ------                    --------------   -----  
 0   ID                        100000 non-null  object 
 1   Customer_ID               100000 non-null  object 
 2   Month                     100000 non-null  object 
 3   Name                      90015 non-null   object 
 4   Age                       100000 non-null  object 
 5   SSN                       100000 non-null  object 
 6   Occupation                100000 non-null  object 
 7   Annual_Income             100000 non-null  object 
 8   Monthly_Inhand_Salary     84998 non-null   float64
 9   Num_Bank_Accounts         100000 non-null  int64  
 10  Num_Credit_Card           100000 non-null  int64  
 11  Interest_Rate             100000 non-null  int64  
 12  Num_of_Loan               100000 non-null  object 
 13  Type_of_Loan              88592 non-null   object 
 14  Delay_from_due_date       100000 non-null  int64  
 15  Num_of_Delayed_Payment    92998 non-null   object 
 16  Changed_Credit_Limit      100000 non-null  object 
 17  Num_Credit_Inquiries      98035 non-null   float64
 18  Credit_Mix                100000 non-null  object 
 19  Outstanding_Debt          100000 non-null  object 
 20  Credit_Utilization_Ratio  100000 non-null  float64
 21  Credit_History_Age        90970 non-null   object 
 22  Payment_of_Min_Amount     100000 non-null  object 
 23  Total_EMI_per_month       100000 non-null  float64
 24  Amount_invested_monthly   95521 non-null   object 
 25  Payment_Behaviour         100000 non-null  object 
 26  Monthly_Balance           98800 non-null   object 
 27  Credit_Score              100000 non-null  object 
dtypes: float64(4), int64(4), object(20)
memory usage: 21.4+ MB
```

There are so many missing values in this data, so we need cleansing data and change some features data type from object to integer.

Baseline Target Distribution

```
train_df['Credit_Score'].value_counts(dropna = False)
```

```
Standard    53174
Poor        28998
Good        17828
Name: Credit_Score, dtype: int64
```

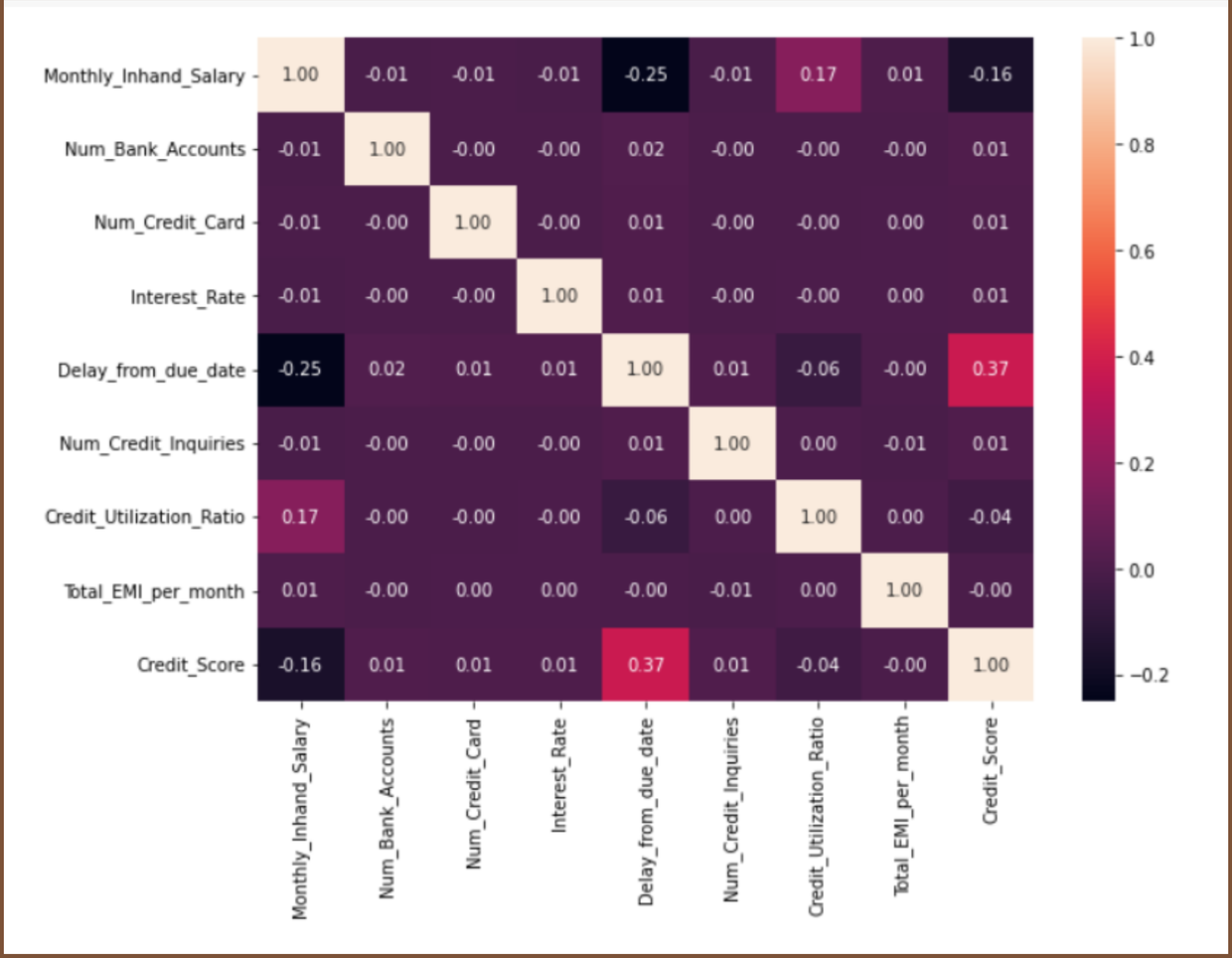
```
train_df['Credit_Score'].value_counts(normalize = True)*100
```

```
0   71.002000
1   28.998000
Name: Credit_Score, dtype: float64
```

The target is 'Credit_Score' convert to numeric which 0 for Standard and Good and 1 for Poor

Baseline

Feature vs Target Correlation



For the first hypothesis, we can see that 'Delay_from_due_date' (0.37) have high correlation to 'Credit_Score' .

Baseline
Detail Evalution

	Method	F1 Score	Classification Report
0	LogisticRegression	0.422973	precision recall f1-score ...
1	DecisionTreeClassifier	0.671919	precision recall f1-score ...
2	RandomForestClassifier	0.741064	precision recall f1-score ...
3	XGBClassifier	0.700241	precision recall f1-score ...

From Baseline model, the bigger F1 score is by using Random Forest Classifier 0.74

Improvement Model
Scaling All Data Numeric

	Method	F1 Score	Classification Report	status
0	DecisionTreeClassifier'>	0.673666	precision recall f1-score ...	scaling
1	RandomForestClassifier'>	0.742807	precision recall f1-score ...	scaling
2	XGBClassifier'>	0.704944	precision recall f1-score ...	scaling

In improvement model, we do the same data preprocessing then scaling the numeric data, the result slightly increase on Randon Forest Classifier (0.743) before (0.741)

Improvement Model Scaling With Data Continue

	Method	F1 Score	Classification Report	status
0	DecisionTreeClassifier'>	0.670364	precision recall f1-score ...	scaling with data continue
1	RandomForestClassifier'>	0.741996	precision recall f1-score ...	scaling with data continue
2	XGBClassifier'>	0.704944	precision recall f1-score ...	scaling with data continue

next improvement scaling with data continue, where the results Random Forest Classifier (0.742) tend to decrease than before (0.743)

Improvement Model
Age Convert to Numeric

	Method	F1 Score	Classification Report	status
0	DecisionTreeClassifier'>	0.682517	precision recall f1-score ...	scaling + age data type
1	RandomForestClassifier'>	0.751547	precision recall f1-score ...	scaling + age data type
2	XGBClassifier'>	0.702058	precision recall f1-score ...	scaling + age data type

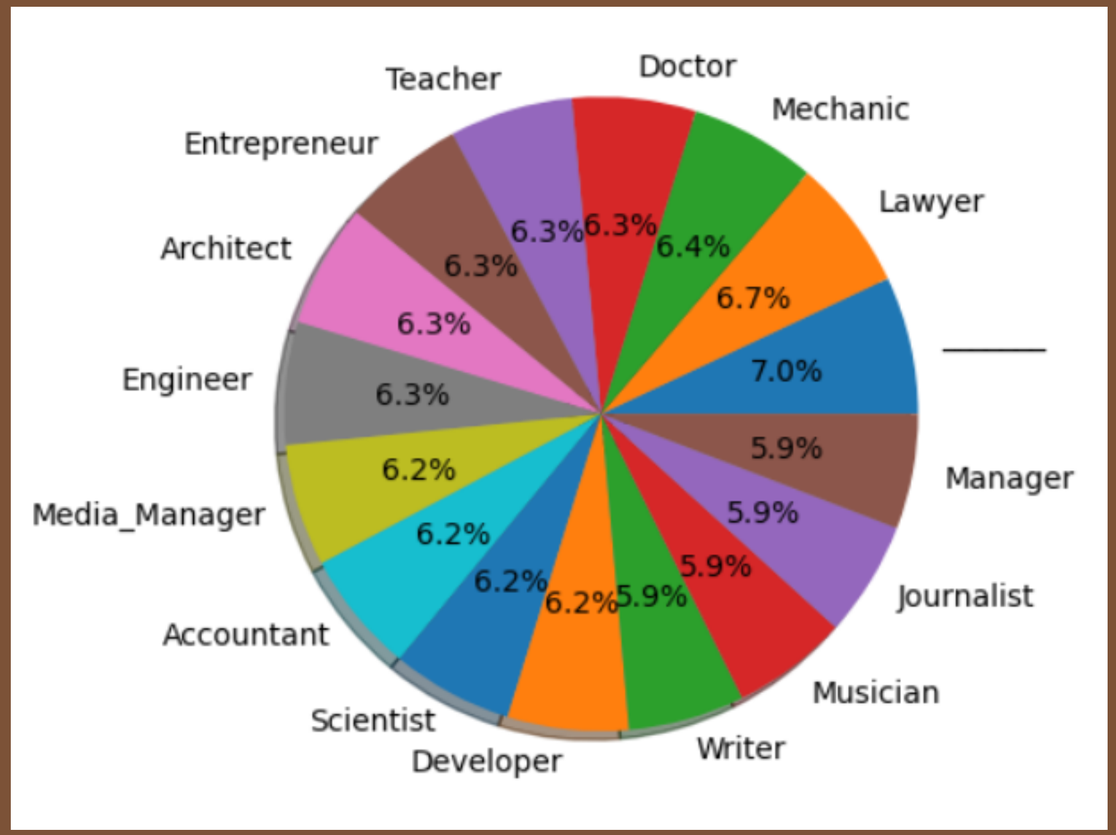
next improvement age convert to numeric, where the results Random Forest Classifier (0.752) increase than before (0.742)

Improvement Model Age Normalize

	Method	F1 Score	Classification Report	status
0	DecisionTreeClassifier'>	0.681556	precision recall f1-score ...	scaling + age normalize
1	RandomForestClassifier'>	0.751856	precision recall f1-score ...	scaling + age normalize
2	XGBClassifier'>	0.702058	precision recall f1-score ...	scaling + age normalize

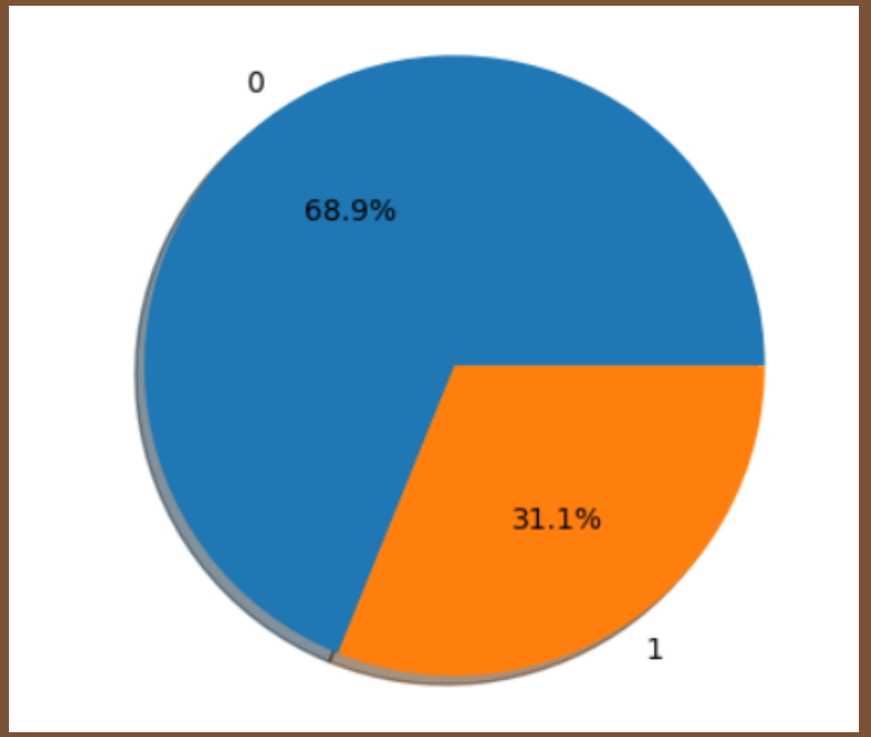
next improvement age normalize, where the results Random Forest Classifier (0.752) tend to be the same than before (0.752)

Occupation

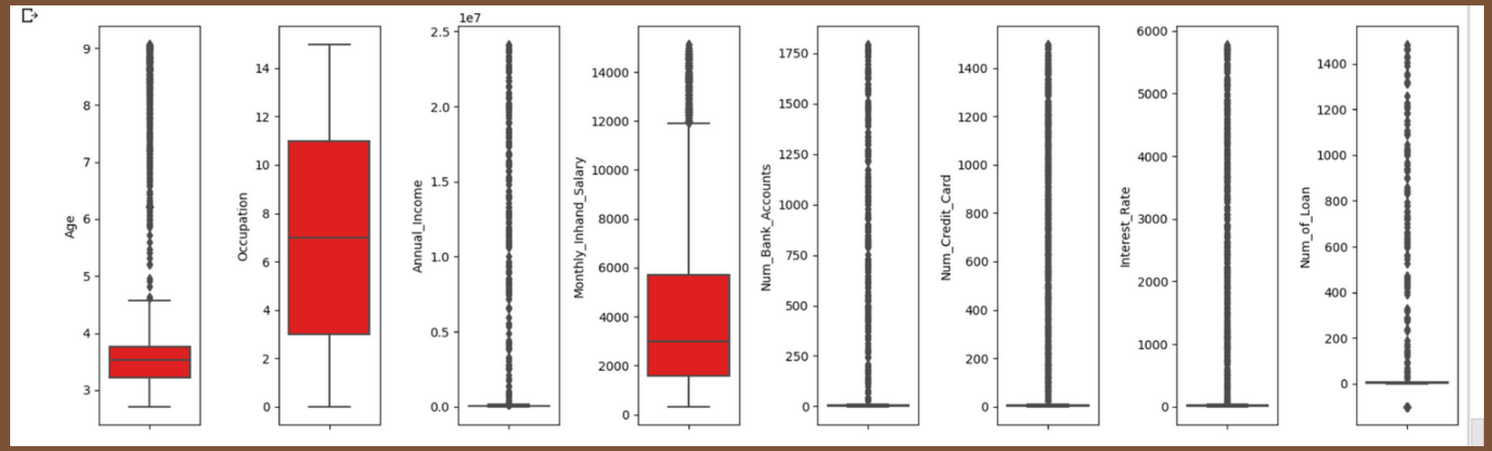


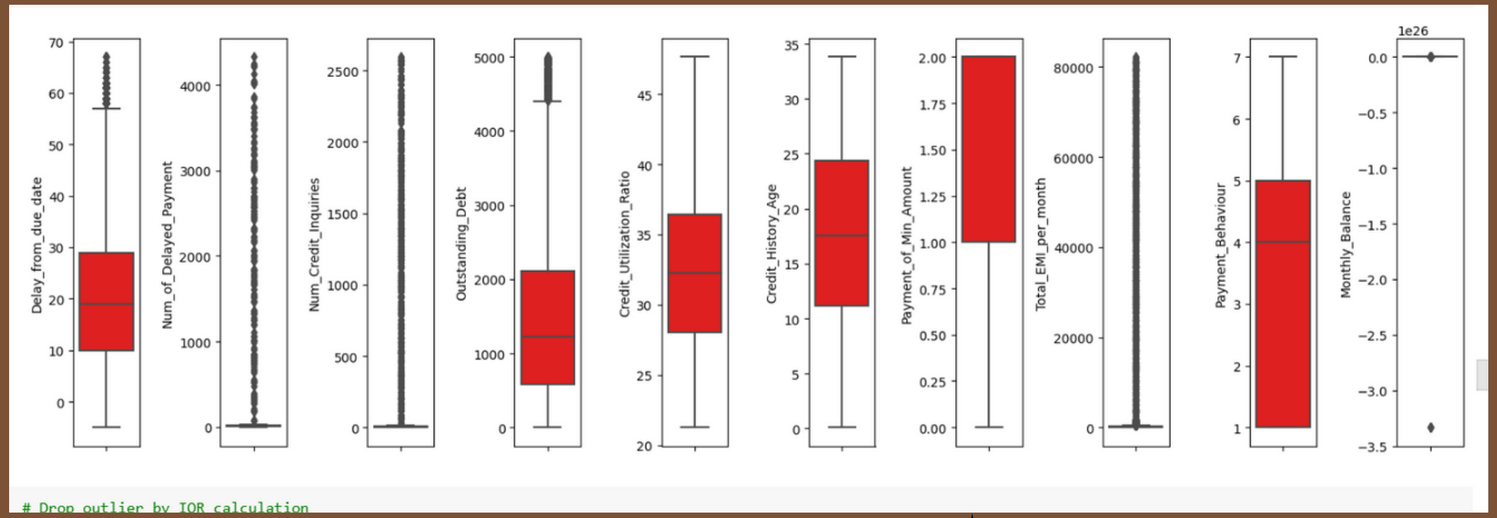
We can see that the Occupation feature is almost equally distributed, without drop the unique value as others

Explore Data Balance

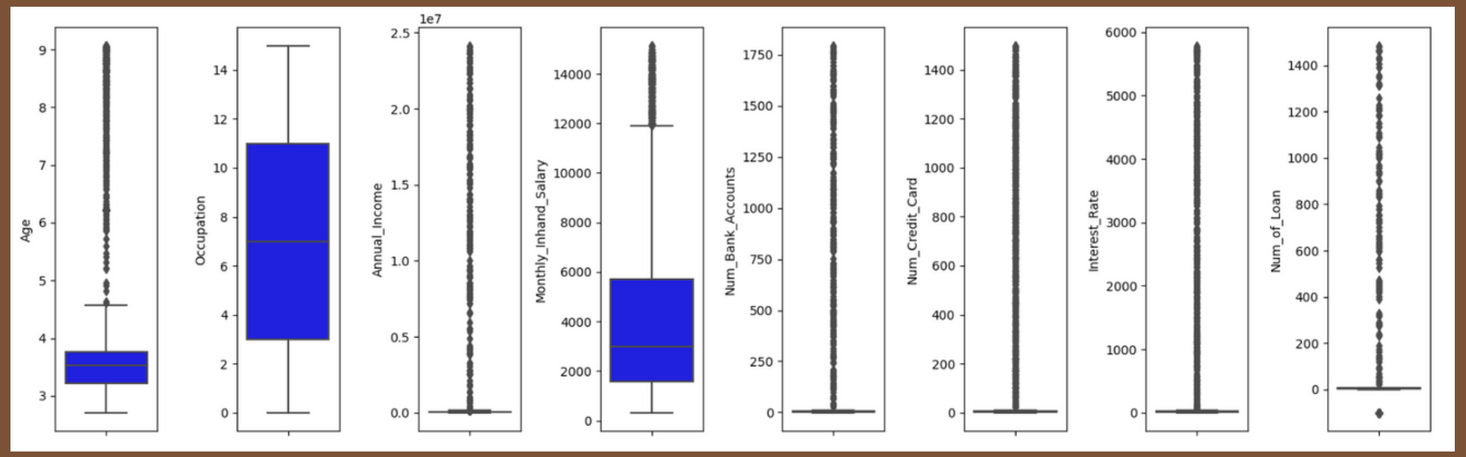


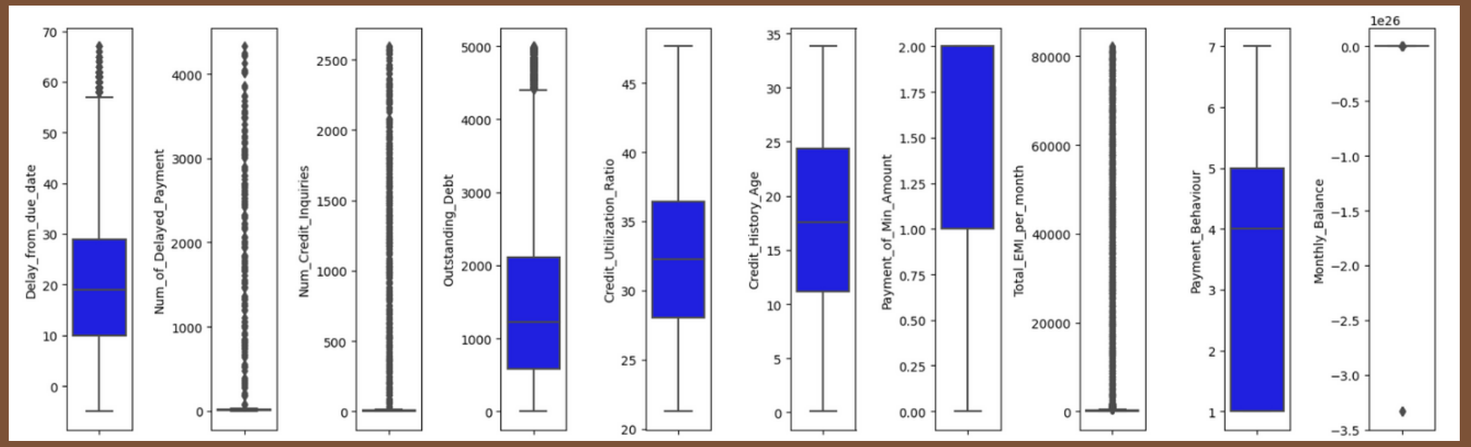
0 for Standard n Good
1 for Poor



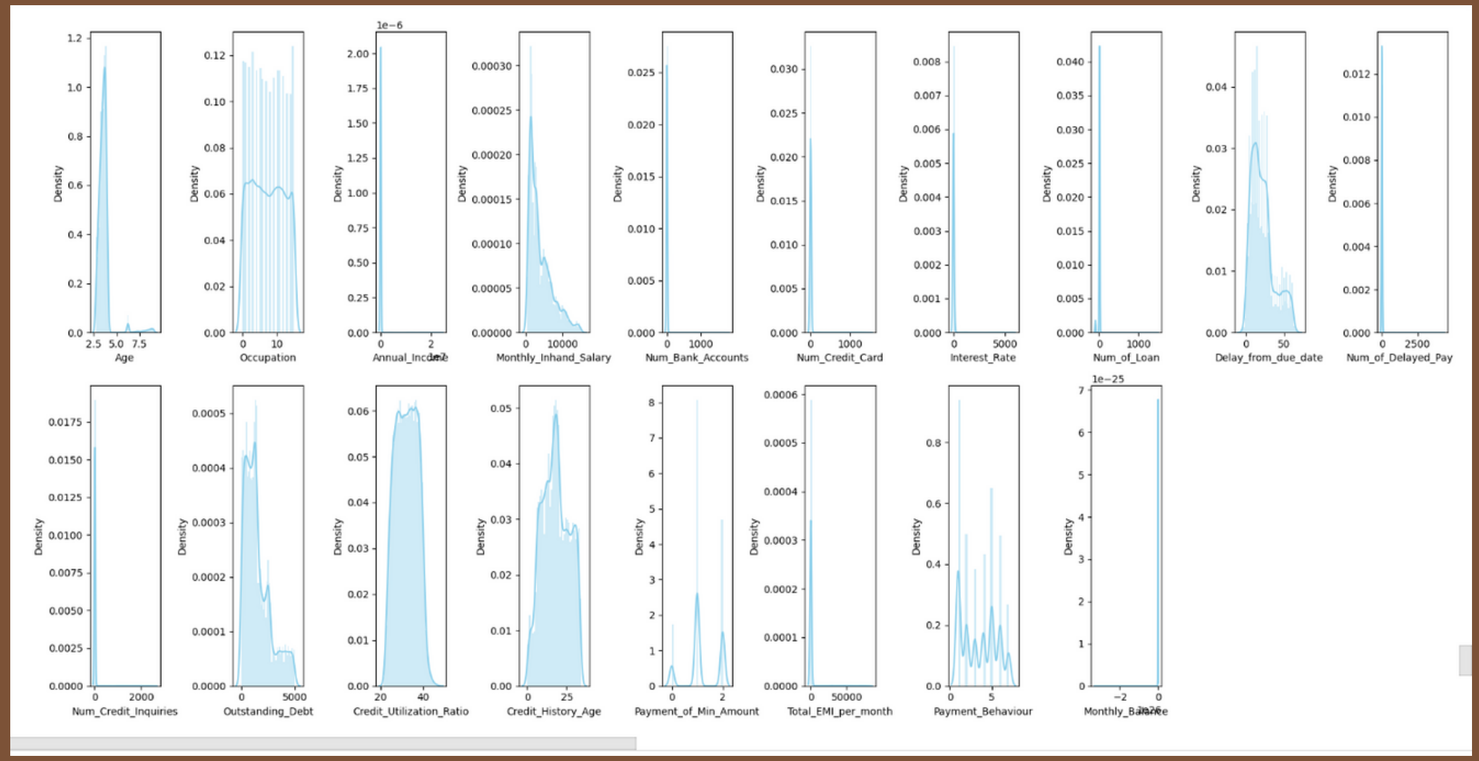


Boxplot all numerical features, addition occupation and payment behavior change to be numeric



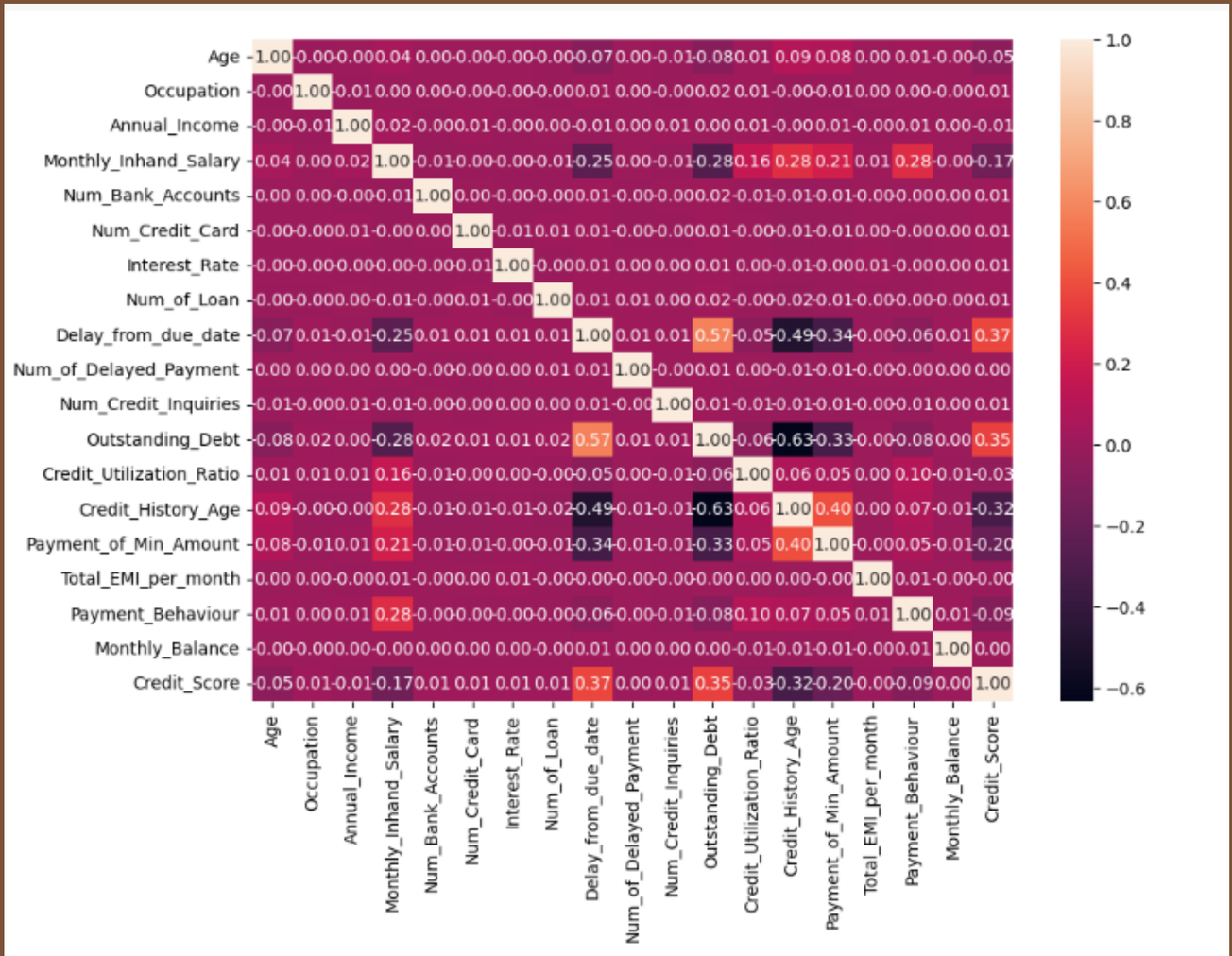


Boxplot all numerical features after remove age outlier



Subplot to see the data distribution, almost the feature has positively skewed

Model Improvement
Feature vs Target Correlation



from this heatmap correlation, we can see that there are 3 features high correlated to 'Credit_Score' there are 'Delay_from_due_date' (0.37), 'Outstanding_Debt' (0.35), 'Credit_History_Age' (0.32)

Improvement Model Remove Outlier Age

	Method	F1 Score	Classification Report	status
0	DecisionTreeClassifier'>	0.680954	precision recall f1-score ...	scaling + remove outlier age
1	RandomForestClassifier'>	0.776788	precision recall f1-score ...	scaling + remove outlier age
2	XGBClassifier'>	0.729125	precision recall f1-score ...	scaling + remove outlier age

next improvement remove outlier age, where the results Random Forest Classifier (0.777) increased than before (0.752)

Cross Validation

```
✓ 1m [110] scores = cross_val_score(model, X, y, cv=5, scoring='f1_weighted')

✓ 1s [111] scores

array([0.79009469, 0.80153983, 0.79554778, 0.79128221, 0.79818571])
```

last step cross validation, we got the biggest f1 weighted scores 0.80

Conclusion

- Classifying customers based on their credit scores helps banks and credit card companies immediately to issue loans to customers with good creditworthiness. A person with a good credit score will get loans from any bank and financial institution.
- We got the biggest f1 scores using Random Forest Classifier (0.78)
- After cross validation using f1 weighted scores (0.80)

Thank You !