Théo Feder, Ludovic Paul, Lucas Rouleau

Explainable AI for credit risk management

During this semester, we had the opportunity to carry out a project closely mirroring real-world professional situations, the main aim of which was to measure the viability of the implementation of an **XAI** (eXplainable Artificial Intelligence) approach in credit scoring.

According to IBM, we can define XAI as *'a set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms.'*

We understand limiting credit risk is a big challenge for banks, especially with the amount of the credit loss: Indeed, according to S&P Global Ratings report, the global domestic credit loss for 2022 was $667Bln, with a very important part for the Chinese market.
Thus, it is very important to reduce the credit risk and especially to understand the credit scoring model used by the banks.
We understand that the potential implementation of XAI would be beneficial as it would help the banks to understand the decisions of the algorithms and increase the transparency.

The first task of our project was to choose from the various datasets presented to us which one was most suitable for our analysis. There were 3 different datasets:

1. The first one represented a German credit risk dataset, where each person is "classified as good or bad credit risks according to the set of attributes". However, we decided not to choose it because the number of data is limited (1000 observations), but mostly because the target column has disappeared, which means that we must create it.
2. The second one is a credit risk dataset that "contains columns simulating credit bureau data". This dataset was bigger than the first one (+30k observations).
3. The last one is a Chinese dataset about Credit score cards. Here again, the number of observations is important, and the dataset is divided into 2 tables (Application Record and Credit Record)

While we quickly decided not to choose the first dataset, we deliberated extensively between the last two. Eventually, we opted for the final one for the following reasons:

- Although we must construct the target variable to assess whether an applicant is a good or a bad client, we were more interested to work in the domain of credit card approval prediction.
- Second, we discovered that this dataset is a Chinese one, which we found more original and very interesting as China is the country with the highest contribution to the global credit loss according to S&P Global Ratings.
- The last dataset is large with an important number of observations, and despite some outliers, the data appeared quite coherent after an initial analysis of data quality
- A final reason is that at the time of selecting our dataset, many groups had opted for the second one, so we decided to choose the third dataset to distinguish ourselves.

Presentation of the dataset and data cleaning:

The dataset we choose is made of two different tables, that can be merged with the column ID.

The first one is “application_record” and has 18 columns and 438 557 observations. You can see a simple presentation of this first table:

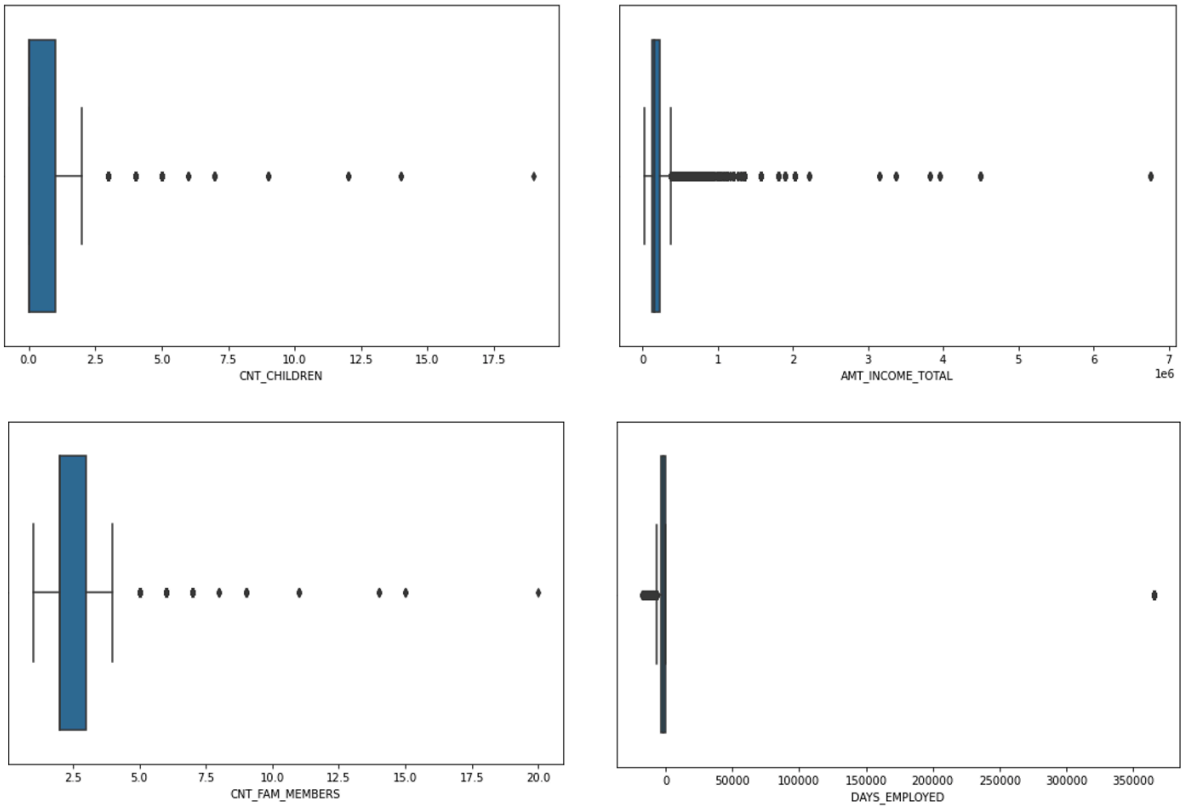
Feature name	Explanation	Remarks	Example of values
ID	Client number		5008804
CODE_GENDER	Gender		M/F
FLAG_OWN_CAR	Is there a car		Y/N
FLAG_OWN_REALTY	Is there a property		Y/N
CNT_CHILDREN	Number of children		0, 3...
AMT_INCOME_TOTAL	Annual income		427500
NAME_INCOME_TYPE	Income category		Working, Pensioner...
NAME_EDUCATION_TYPE	Education level		Higher education, Academic degree...
NAME_FAMILY_STATUS	Marital status		Married, Separated...
NAME_HOUSING_TYPE	Way of living		Rented apartment, House/Apartment
DAYS_BIRTH	Birthday	Count backwards from current day (""), -1 means yesterday	-12005
DAYS_EMPLOYED	Start date of employment	Count backwards from current day (""), If positive, it means unemployed.	-4524
FLAG_MOBIL	Is there a mobile phone		0/1
FLAG_WORK_PHONE	Is there a work phone		0/1
FLAG_PHONE	Is there a phone		0/1
FLAG_EMAIL	Is there an email		0/1
OCCUPATION_TYPE	Occupation		Security staff, Sales staff...
CNT_FAM_MEMBERS	Family size		1, 2...

The second table, “credit_record”, has 3 columns and 1 048 575 observations:

Feature name	Explanation	Remarks
ID	Client number	
MONTHS_BALANCE	Record month	The month of the extracted data is the starting point, backwards, 0 is the current month, -1 is the previous month, and so on
STATUS	Status	0: 1-29 days past due 1: 30-59 days past due 2: 60-89 days overdue 3: 90-119 days overdue 4: 120-149 days overdue 5: Overdue or bad debts, write-offs for more than 150 days C: paid off that month X: No loan for the month

The first operation was to describe and visualize the data with histograms to detect possible outliers that would disturb our analysis.
In addition, we noticed that some variables (like DAYS_BIRTH) were not very clear, prompting us to make some transformations for these variables.

We continue the data visualization with box plots in order to have a better understanding and a better visualization for outliers:



For example, we detected an error for the column “Days_employed”, representing the number of days where the clients worked. The value of this column is supposed to be negative as it counts backward from the current day. We noticed that over 70k observations had the value 365 243 for this column. We decided to delete them.

Moreover, we decided to delete extreme values for different categories:

- ‘AMT_INCOME_TOTAL’: We keep individuals whose annual income is lower than 400k RMB (approximately 5% of the observations). We did it because we noticed that after this threshold, the values increased exponentially, which could disrupt our future analysis.
- ‘CNT_CHILDREN’ and ‘CNT_FAM_MEMBERS’, which represent the number of children and the family size of a client. We drop 8 outliers that have more than 10 children and a family size exceeding 12 members.
- We created a column ‘FLAG_SINGLE’ that is equal to ‘CNT_FAM_MEMBERS’ minus ‘CNT_CHILDREN’ and that represents the number of family members without the children. We drop the 5 outliers for which ‘FLAG_SINGLE’ was inferior to 1.

After dropping outliers, we decided to create columns to have a clearer view:

- YEARS_EMPLOYED' which gives the positive number of years a person have worked
- ‘AGE', which give the age of the person with a positive figure and not a negative number of days like the column ‘DAYS_BIRTH’
- AMT_INCOME_USD’ which gives the annual income in USD and not in Yuan, as the data is from China. To do so we used the spot exchange rate at the time that was 1$ for 7.2235 Yuan.

For the table ‘Application_record’ we had in total 22 columns with the ones we created. We decided to keep 16, we deleted the following categories:

- CODE_GENDER: We decided not to consider this for our scoring because of discriminatory purposes. Indeed, it is neither ethical nor legal to consider gender to decide to grant a credit or not.
- AMT_INCOME_TOTAL as it is replaced by AMT_INCOME_USD
- DAYS_BIRTH as it is replaced by AGE
- DAYS_EMPLOYED as it is replaced by YEARS_EMPLOYED
- CNT_FAM_MEMBERS as it is replaced by FLAG_SINGLE
- FLAG_MOBIL, equal to 1 if there is a mobile phone, is redundant with FLAG_PHONE, equal to 1 if there is a phone.

Another key step before the use of different models is to implement a mapping to transform categorical variables to numerical values. This operation is summarized with the following tables:

NAME_EDUCATION_TYPE	NAME_INCOME_TYPE	NAME_FAMILY_STATUS	Numeric Value
Lower secondary	Student	Widow	1
Secondary / secondary special	Pensioner	Separated	2
Incomplete higher	State servant	Single / not married	3
Higher education	Working	Civil marriage	4
Academic degree	Commercial associate	Married	5

NAME_HOUSING_TYPE

Housing Type	Numeric Value
With parents	1
Rented apartment	2
Municipal apartment	3
Co-op apartment	4
Office apartment	5
House / apartment	6

OCCUPATION_TYPE

Occupation Types	Numeric Value
Low-skill Laborers, Cleaning staff, Cooking staff, Waiters/barmen staff	1
Security staff, Sales staff, Laborers, Drivers	2
Medicine staff, Secretaries, HR staff, Accountants, Core staff, Realty agents	3
Private service staff, High skill tech staff, Managers, IT staff	4

In addition, we transform variables that can have Y/N outcomes into numerical ones with 1/0 as possible outcomes. Moreover, we perform one hot encoding for other categorical variables.

IMPLEMENTATION OF THE SCORE

Concerning the second table, *credit_record*, we also perform several operations before using different models.

First, for the column '*Status*', we remove the possible outcome *'X'*, which corresponds to a client that doesn't have a loan for the month. Indeed, we are not interested in this status to construct our score because it doesn't bring any information for our credit scoring. For the other possible outcomes of Status, we use another mapping that enables us to create a column '*Points*' that will be helpful to create our credit score. This operation is summarized in the following table:

STATUS	Meaning	Points
C	Paid off that month	9
0	1-29 days past due	-1
1	30-59 days past due	-9
2	60-89 days overdue	-11
3	90-119 days overdue	-13
4	120-149 days overdue	-16
5	Overdue or bad debts, write-offs for more than 150 days	-26

This mapping allows to reward the clients that have reimbursed their credit during the month and penalize gradually the clients that have a past due.

In addition, we create a column '*Weight*'. This variable is equal to:

$$Weight_i = e^{(0.01 * MonthsBalance_i)}$$

This formula helps to consider the **temporality**: Indeed, the weight is bigger for a client with a recent overdue. On the other hand, the weight decreases exponentially as you go back in time and that the overdue occurred a long time ago.

This enables us to create the variable '*Weighted_Points'* which is equal to the product of the variable *'Points' and 'Weights'*.

After that, we create a variable '*Line_Count'*, which represents the number of records a given client has in the bank. Because we want to train our models for clients that have sufficient historical data, we only keep clients that have at least 12 records (which corresponds to 1 year).

Finally, we group the dataframe by ID and for each client we calculate its score, which is equal to the sum of the Weighted points divided by the sum of the weights for each client i:

$$Score_i = \frac{\sum_{j=1}^{n_i} WeightedPoints_{ij}}{\sum_{j=1}^{n_i} Weights_{ij}}$$

Where:

- *i represents a given client (ID)*
- N_i *is the number of records for a client*
- $WeightedPoints_{ij}$ *are the weighted points for the j-the record of client i*
- $Weights_{ij}$ *is the weight of the j-th record for a given client.*

We merge the tables to have the features and the score of all the different clients.
By keeping the clients whose IDs are present in the 2 original tables, we end up with a final dataset of 13269 clients.

This score is standardized with a range between 0 and 100 and not between -26 and 9 for better interpretability.

SCORING AND IMPLEMENTATION OF THE MODELS

Because of the original mapping of the points, we notice that our score distribution is not very intuitive, with most of the distribution being between -5 and 10. In order to be clearer, we decided to rescale our distribution in order to have a score between 0 and 100. It enables us to have a better classification:

- Between 0 and 25: Very high risk
- Between 25 and 50: High risk
- Between 50 and 75: Medium risk
- Between 75 and 100: Low risk.

We noticed that the dataset is unbalanced, which is an important issue when we want to predict less frequent categories (for example clients that represent a very high risk).
To handle this issue we resample the data by oversampling the minority classes.

We then train several models in order to select the best one:

- Random Forest
- Gradient Boosting
- XGBoost
- LightGBM

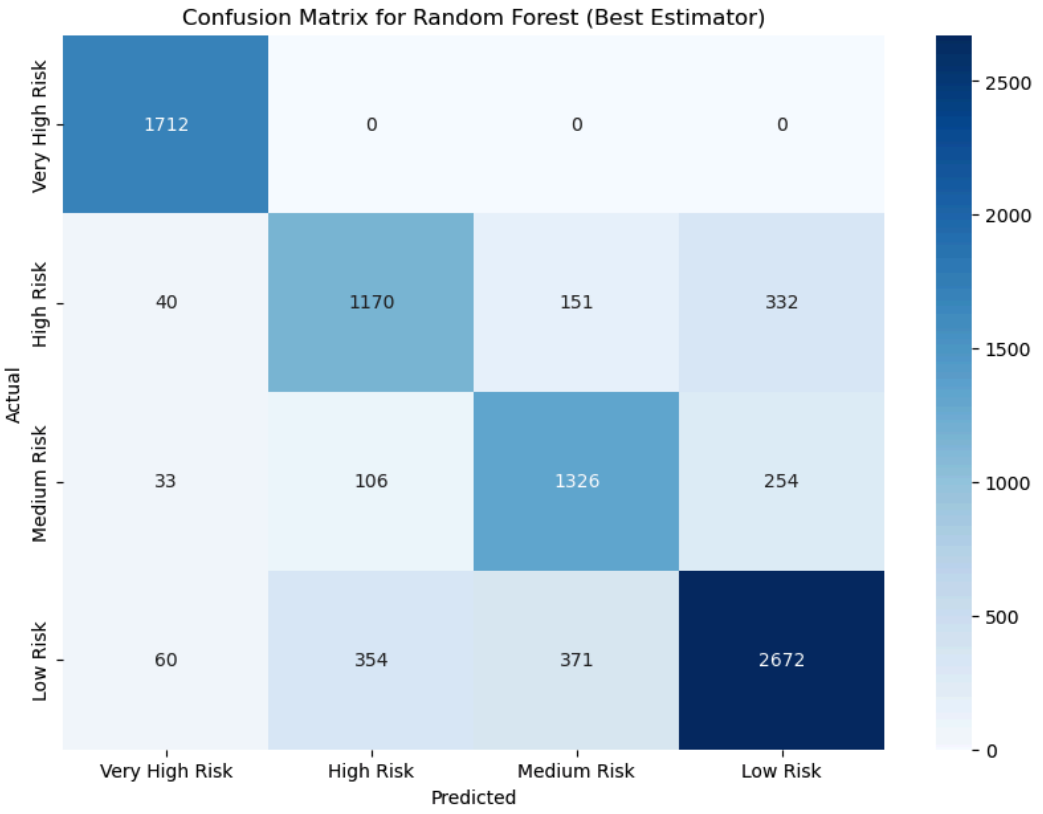
We evaluate the models and plot confusion matrices with cross-validation. We completed this analysis with an evaluation of the models with VIF (Variance Inflation Factor) and PCA(Principal Component Analysis) but it didn't improve the score.
In addition, we also evaluate other models: a Neural network and a Multinomial logistic regression. We obtain the following results:

Model	Test Accuracy	Precision (macro avg)	Recall (macro avg)	F1-Score (macro avg)
Multinomial Logistic Regression	0.4178	0.32	0.29	0.22
Neural Network (Categorical Crossentropy)	0.5949	0.57	0.50	0.42
RandomForest	**0.7998**	**0.79**	**0.80**	**0.80**
GradientBoosting	0.5062	0.64	0.39	0.47
XGBoost	**0.7322**	**0.75**	**0.67**	**0.70**
LightGBM	**0.7054**	**0.76**	**0.66**	**0.66**

We notice with this table that the three best models are the Random Forest, the XGBoost and the LightGBM.

The final step of this part involves tuning hyperparameters for these 3 machine learning models using RandomizedSearchCV. It helps to find the best set of hyperparameters (learning rate, number of trees….)
For each model, the best parameters and cross-validation are printed with a confusion matrix to visualize the performance of the best model in classifying the different risk levels we defined earlier.

After looking at the results and the confusion matrix, we conclude that the best model for our analysis is the Random Forest.
Indeed, this model has excellent precision, especially for the Very High-Risk category. It is even more important from a business point of view as, for a Chinese Bank that suffers a lot from credit loss, our priority is to minimize the risks.
We will use this model for the implementation of the XAI models.



IMPLEMENTATION OF EXPLAINABLE ARTIFICIAL INTELLIGENCE

The final part of our analysis concerns the implementation of XAI.
As a reminder, XAI represents the methods that make the decision-making of processes of AI models understandable for humans, ensuring transparency and interpretability.
In the context of credit scoring, an efficient implementation of XAI would enable banks to explain and justify credit decisions to customers and regulators, but also ensures fairness by detecting and addressing biases.
In our analysis we implemented 3 different XAI models:

- Permutation importance
- SHAP
- LIME

1. **Permutation importance:**

The permutation importance model is a model which measures the importance of features in a predictive model, in order to understand their contribution to the model's predictions.

Feature	Value	Uncertainty
YEARS_EMPLOYED	0.172	± 0.003
AGE	0.116	± 0.003
AMT_INCOME_USD	0.093	± 0.002
FLAG_PHONE	0.035	± 0.002
OCCUPATION_TYPE_High skill tech staff	0.026	± 0.001
CNT_CHILDREN	0.025	± 0.001
FLAG_WORK_PHONE	0.024	± 0.001
FLAG_OWN_CAR	0.023	± 0.002
OCCUPATION_TYPE_Laborers	0.021	± 0.001
FLAG_OWN_REALTY	0.021	± 0.001

With this table we understand the most important features for the prediction. The first value indicates how much the model's performance decreases when the feature is randomly shuffled. The second value indicates the standard deviation of the permutation importance scores.
We see with these results that features with the most importances are pretty common and not very surprising (the number of years employed, the age, the yearly income) as they are important characteristics that a bank checks when a client is applying for a credit.

2. **SHAP**

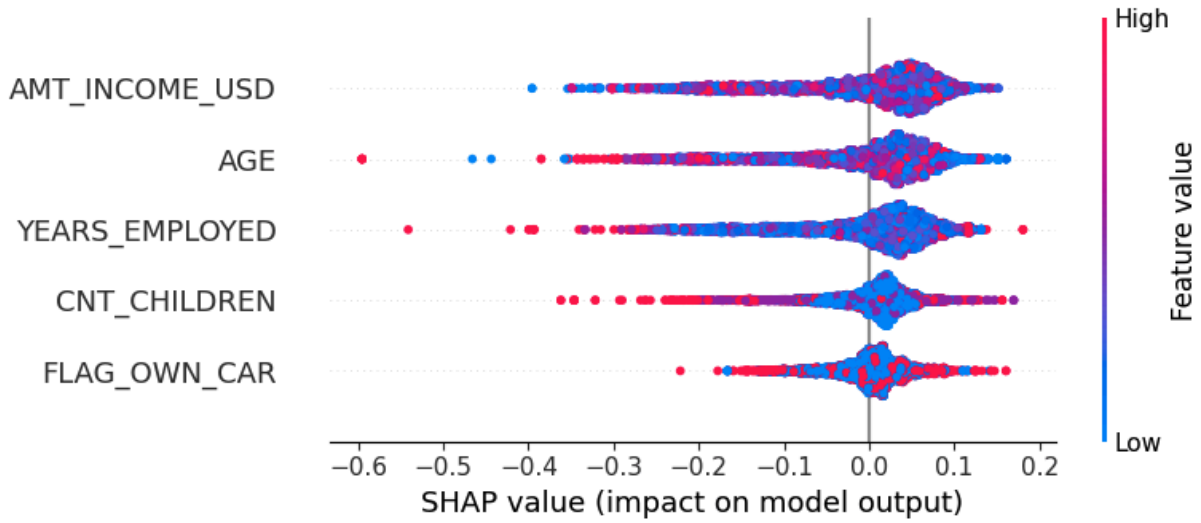
The second XAI model we implement is a SHAP model.
For this implementation, we selected 5 of the most important variables according to the permutation importance: (Age, Years_Employed, AMT_Income_USD, CNT_Children, Flag_Own_Car)

Shap is based on the the concept of Shapley values, and it indicates how each feature influences a prediction:
A positive SHAP value increases the model's prediction, while a negative SHAP value decreases the model's prediction.

In our analysis we plotted 3 different graphs:

A. Summary plots

It displays the overall impact of features on predictions, with the Y-axis representing the features and the X-axis the SHAP values.The dots represent the predictions while the color indicates the feature value (red = high and blue = low)



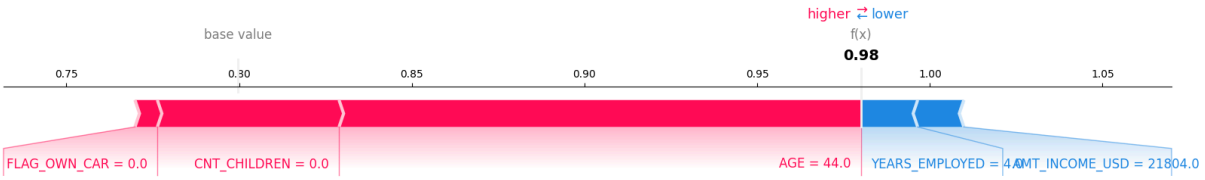
This graph represents the summary plot for the High Risk. We notice for each variable a bulk just above zero.
Moreover, we see that income is the first factor in this plot, which makes sense from a financial point of view.
Finally, we see that for most variables the red dots, representing high values, are in the tail of the distribution, which is logical since these features represent generally good financial health and hence not a risky client.

B. Force plots

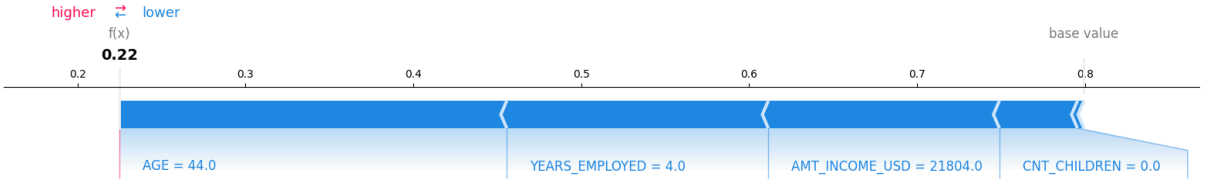
This kind of graph shows the baseline value and how each feature's SHAP value moves the prediction from this baseline to the final prediction:
Actual: High Risk, Predicted: High Risk

In the following example, we see how the variables push or not for the adoption of each class. Here, the prediction is accurate and we can determine why thanks to this method.

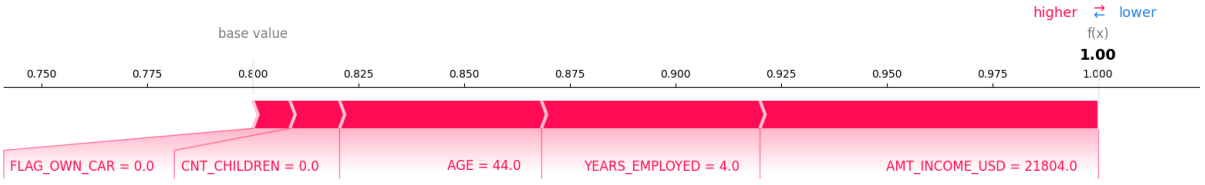
```
SHAP Force Plot: Random Forest (Class Very High Risk)
```



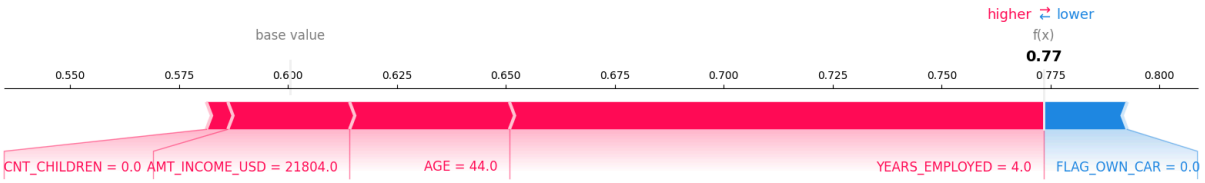
SHAP Force Plot: Random Forest (Class High Risk)



SHAP Force Plot: Random Forest (Class Medium Risk)



SHAP Force Plot: Random Forest (Class Low Risk)

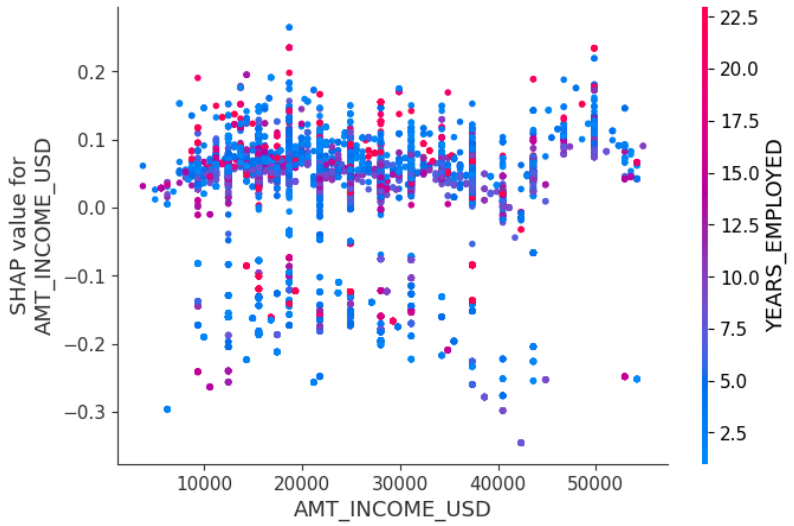


C. Dependance plots

Finally, the dependance plots, which show the effect of a single feature on predictions by plotting SHAP values against actual feature values

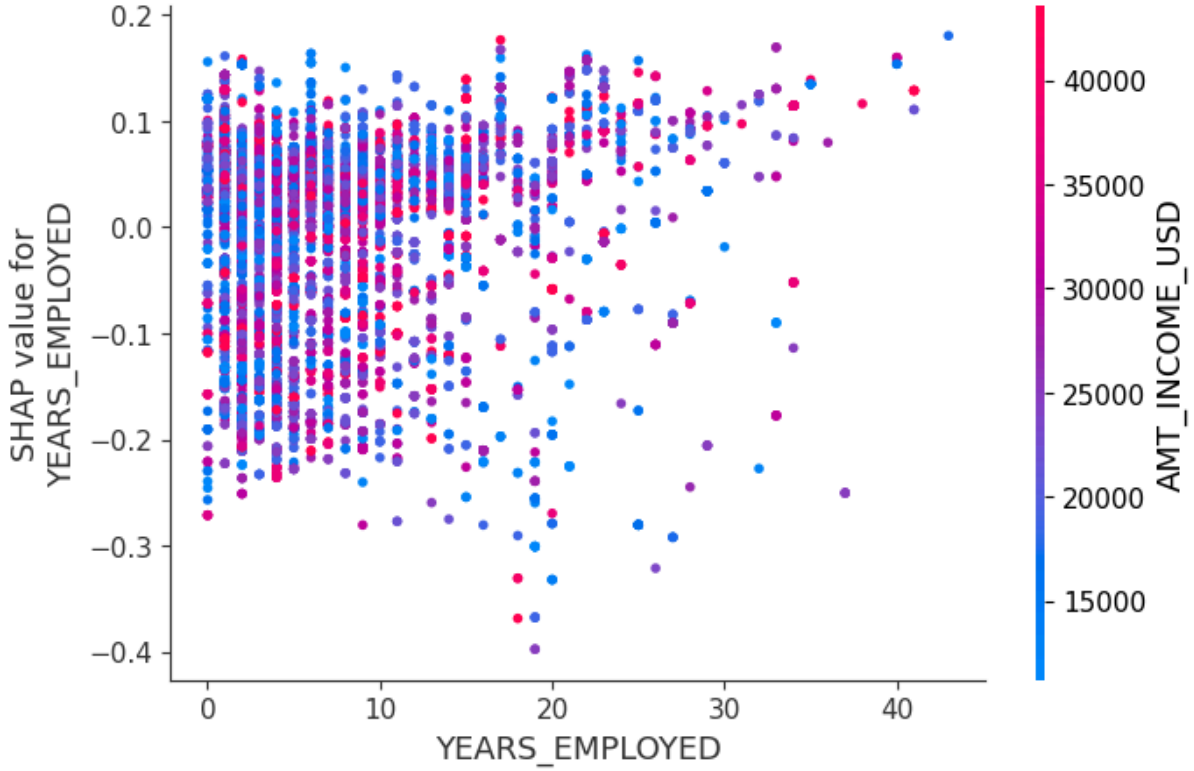
We can extract multiple types of information from them. For instance:

SHAP Dependence Plot: AMT_INCOME_USD & YEARS_EMPLOYED (Class Very High Risk)



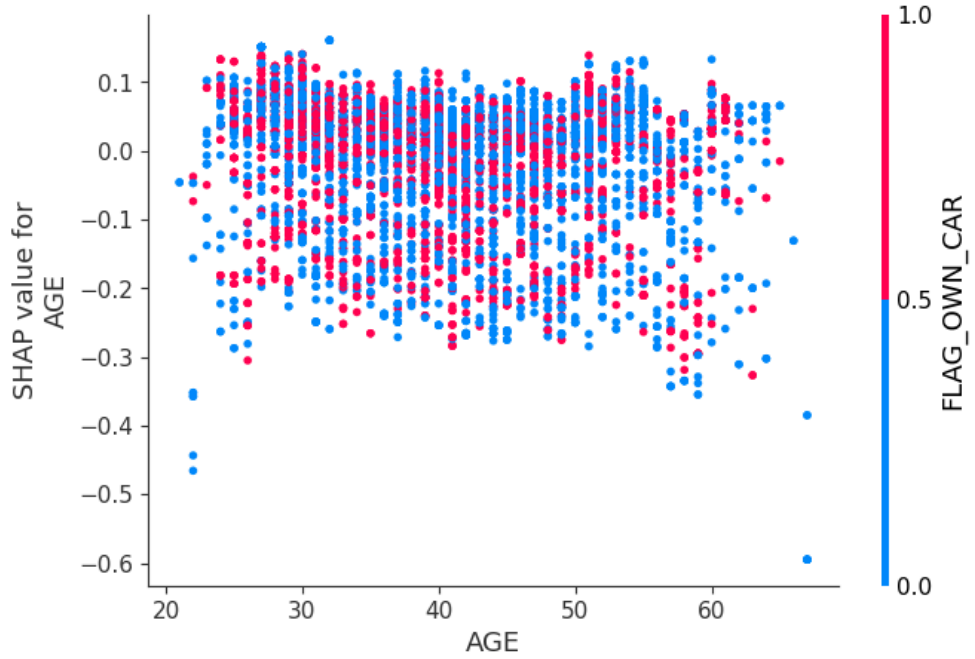
Here we can see that the income generally participates positively to being very high risk.

```
SHAP Dependence Plot: YEARS_EMPLOYED & AMT_INCOME_USD (Class Medium
Risk)
```



Here we can see that the more experience a person has, the more likely she is to be in medium risk.

```
SHAP Dependence Plot: AGE & FLAG_OWN_CAR (Class High Risk)
```

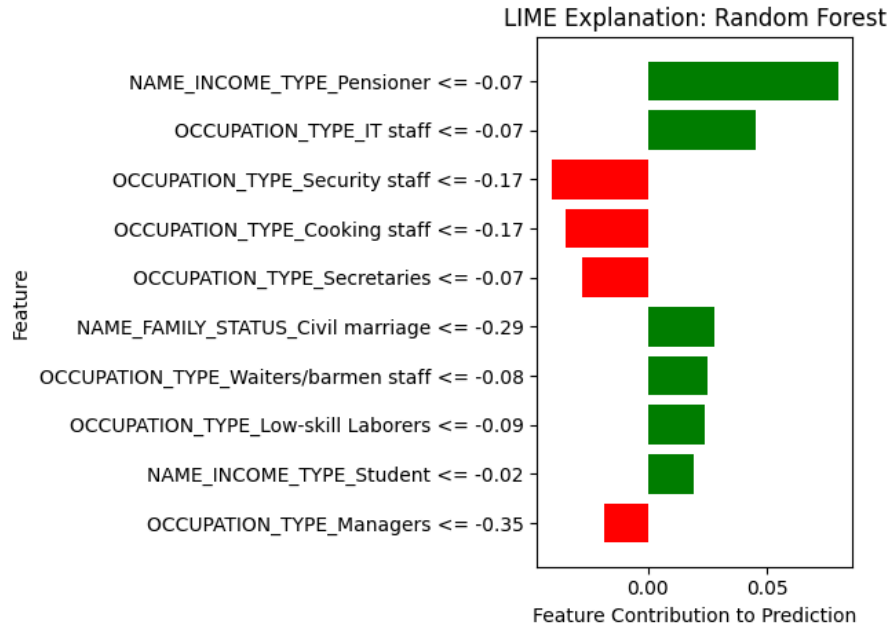


But with these graphs, the interpretation is not always so straight forward as shown in this last plot.

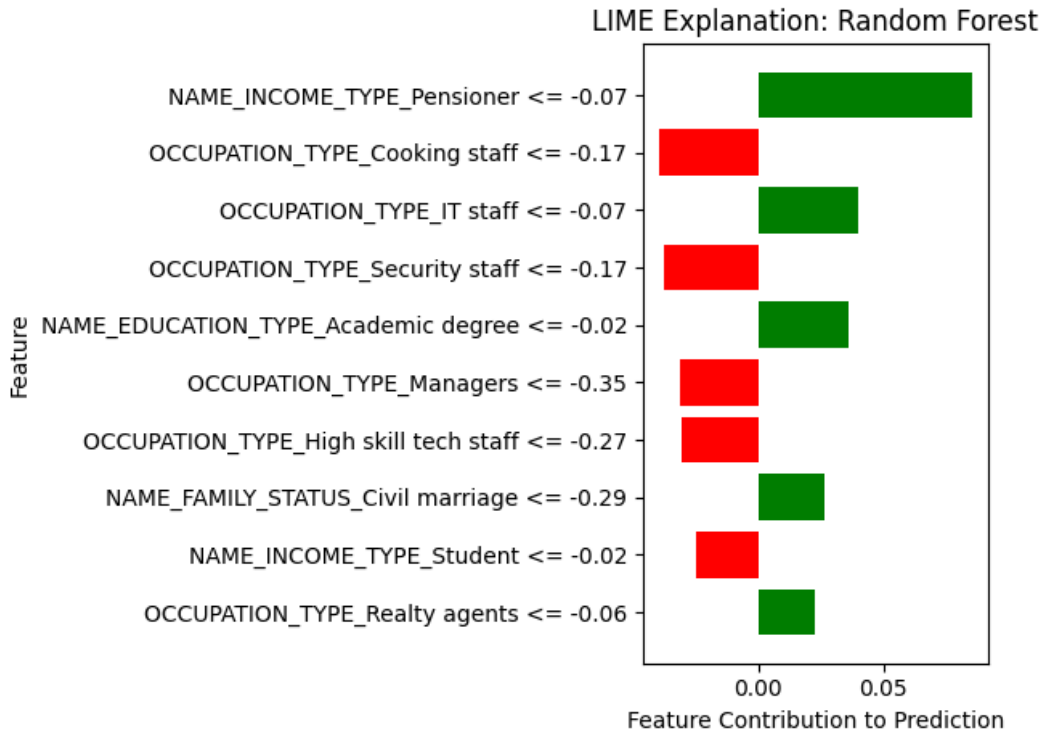
3. LIME

The final XAI model is LIME
In this XAI, taking random examples, our results are very different from what SHAP and permutation importance had shown us. Here, our one hot encoded categorical variables are the most important, contrary to all evidence. We know that LIME is not as reliable as shap, heaven though it is faster , and here may be an example.

```
Instance index: 4104
```



```
Instance index: 4760
```



In addition of these models, we would have liked to make other implementations in the XAI field:

1. Implementation of Quantus
2. Combined approach with OmniXAI
    - Initial Global Insight with PDP (Partial Dependance Plot)
    - Detailed Local Insight with SHAP
    - Accurate Local effects with ALE (Accumulated Local Effects)

Conclusion:

In this analysis, we tried to implement XAI in order to enhance a classical analysis for credit scoring using machine learning models.

We clearly understand the impact that a successful implementation of XAI can have from a business point of view, especially using our case where Chinese banks would benefit from a reduction of the credit risk.
Indeed, XAI can enhance the understanding of credit scoring models and the explanation of the contribution of different variables to predictions.

However, in our analysis, the results on the implementation of XAI are mixed: XAI gives us a few precisions but overall the expected improvements were not fully realized. It is indeed possible that a better implementation might have yielded improved results.

Overall, while XAI has a big potential, especially in credit scoring where it could offer significant benefits, the results of our analysis indicate mixed outcomes and challenges that need to be addressed.