Predictive Underwriting in Commercial Lines

Executive Summary

The aim of the project is to replicate the pricing decisions made by underwriters for two AXA products. A comprehensive insurance package tailored to safeguard SME business interests and a bond for construction companies. The purpose of the project is to decrease the time spent by underwriters who manually assess the risk of every company requesting for one of these types of coverage and to leverage machine learning tools to streamline the client quotation process.

The solution developed is a system divided into two sub engines. A pricing engine developed by the actuarial team which returns a premium depending on the needs of a given company for all the standard cases and a machine learning engine developed by the data science team which returns a suggested premium and a confidence level for all non-standard (referral) cases. The machine learning engine is triggered every time the pricing engine fails to return a premium. A key part of this initiative is to collect a sufficient number of non-standard cases to assess the machine learning engine performance in real conditions.

AXA has assessed that the machine learning is estimated to reduce the underwriting process significantly for customers and save valuable man hours of underwriters to deal with more complex risks.

AXA Insurance Pte Ltd (Company Reg. No.: 199903512M)
8 Shenton Way #24-01 AXA Tower Singapore 068811
AXA Customer Centre #B1-01
Telephone: +65 6880 4888 – axa.com.sg

Developing a Machine Learning Engine

Infrastructure

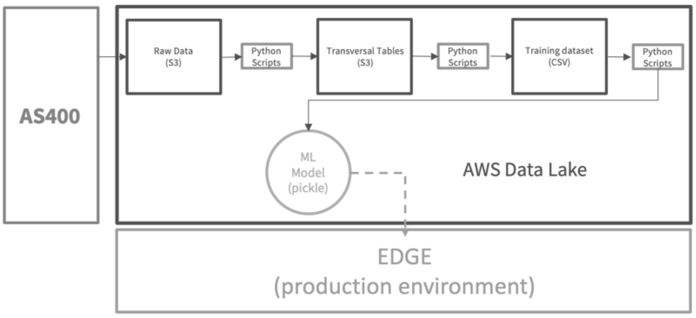

Figure 1: Overview of the infrastructure

The machine learning engine developed on the AXA AWS platform where Python scripts are used to transform the raw data from the core systems into transversal tables which are used to create the training set needed to develop a machine learning model. The model is built using Python Machine Learning libraries (Pandas, Scikit-Learn, LightGBM…). Once the model created, it is saved as a pickle file and sent to the production environment (EDGE).

Data

Python scripts transform the raw data into meaningful data where many business rules are applied in order to obtain a proper training set. As the machine learning engine needs to suggest the premiums for two different products, two machine learning models are created (one for each product) and thus two different training set are created. The data sources remain the same for the two products, but the data used by the machine learning models are different. The data includes information about the policy (duration, coverage type, risks insured, sum insured for the different risks, contract type, number of risks insured, etc.), the company (age of the company, activity sector, number of officers, etc.) and the claims history of the company if any (occurrence date, sum claimed). The list of all variables used by the different models are shown in the appendix.

The useful data/features to predict a correct premium are selected using several methods. The feature selection methodology consists mainly in eliminating features having a single unique value, collinear features, zero importance features and low importance features. At the end of this process 9 features are selected for the Bond product and around 42 features for the SME product. Policies issued before 1st Jan 2019 were used in the training set to create the machine learning models and policies issued after 1st Jan 2019 were used by the validation set to assess the performance of the models (back testing).

Modeling Approach

Our modeling approach consists of using a supervised learning algorithm. We have selected the open source library LightGBM developed by Microsoft. LightGBM is a gradient boosting framework that uses tree-based learning algorithms. Its advantages are faster training speed and higher efficiency, lower memory usage and better accuracy. Additionally, LightGBM supports the SHAP (SHapley Additive exPlanations) library which enables to explain each prediction at the company level. This functionality is a great advantage to explain the behavior of the machine learning models to the underwriters.

For each product, the hyper-parameters optimization is done using the Hyperopt Python library which is a Bayesian optimization technique. The hyper-parameters are found using cross-validation on the entire original training dataset.

Once the hyper-parameters found, 20 LGBMRegressor instances are trained on 20 different subsets of the original training dataset using the Bootstrap Method. By using this bagging method of 20 LGBMRegressor instances per product, we are able to define a confidence level for each prediction made by the machine learning engine (see next section).

Confidence Level Definition

An additional request from the underwriting team was to be able to provide a "confidence level" for each suggested premium predicted by the machine learning models. The confidence level is different from the confidence interval which is already well defined in the literature. In this paper, the confidence level is defined as the "certainty index" for every prediction made by the ML models. In other words, how certain the machine learning model is in predicting that the premium for a given company will be X S$? To our knowledge, only few research articles have mentioned this type of problem known as Regression Conformal Prediction. As most solution proposed from the academic world were either too complex or not adapted to our business problem, we have decided to create our own definition of confidence level.

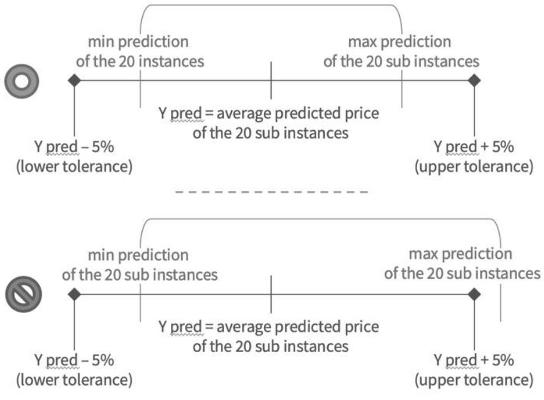

Figure 2: Definition of the confidence level

The confidence level is defined as follow:

- Let CL be a confidence level between [0;1]
- Let Ymin be the minimum of all premiums predicted by the 20 LGBMRegressor instances
- Let Ymax be the maximum of all premiums predicted by the 20 LGBMRegressor instances
- Let Ypred be the mean of all the premiums predicted by the 20 LGBMRegressor instances
- Let alpha be a tolerance level between [0;1]

If any predicted premium from the 20 instances falls outside the interval: [*Ypred*(1-alpha); Ypred*(1+alpha)*] then *CL* = 0
If none of the predicted premiums falls outside the same interval, then $CL = 1 - \frac{(Ymax - Ymin)}{(Ypred*(1-alpha); Ypred*(1+alpha))}$

Performance of the Models

Performance of the models have been back tested during the development of phase.

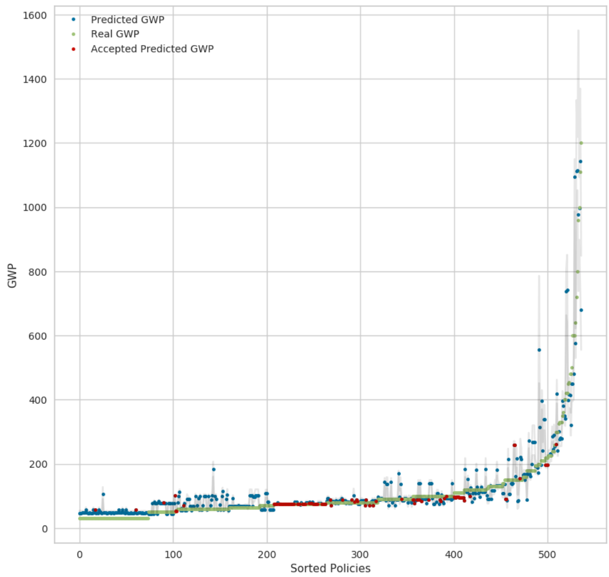

Figure 3: Results of the predicted premiums for Immigration Bonds on the validation set

Figure 3 shows the results the machine learning model for the Bond product. The red dots are the accepted premiums predicted by the model after checking the confidence level. Blue dots are rejected predictions and green dots are the real premiums of the Bond policies. Confidence level allow us to reduce the risk to predict a wrong price.

After applying the confidence level with a threshold at 0, the error on predicted premiums decreases around:

- -1.7S$ on average (vs 7.7S$ without confidence level) for Bonds
- 7.6S$ on average (vs 25.36S$ without confidence level) for SME Product

Future State

The immediate next step is to monitor the behaviour of the deployed machine learning models to check if predicted premiums are coherent with the decision of underwriters. In parallel, collecting data generated by the production environment is a must in order to increase the performance of the models.

The premiums predicted by the machine learning models for non-standard cases need to be reviewed by the underwriters to define what is the acceptable confidence level where the machine learning engine can automatically suggest a premium without any human intervention.

Finally, patterns discovered by the machine learning models with the help of the Shapley values can be integrated back to the pricing engine to decrease the number of non-standard cases.

References

Regression conformal prediction with random forests: Johansson, U., Boström, H., Löfström, T. et al. Mach Learn (2014) 97: 155. https://doi.org/10.1007/s10994-014-5453-0

Regression Conformal Prediction with Nearest Neighbours: Harris Papadopoulos, Vladimir Vovk, Alex Gammerman

Uncertainty Quantification in Deep Learning: https://www.inovex.de/blog/uncertainty-quantification-deep-learning/

lightGBM: https://github.com/microsoft/LightGBM

Hyperopt: https://github.com/hyperopt/hyperopt

SHAP: https://github.com/slundberg/shap

Appendix

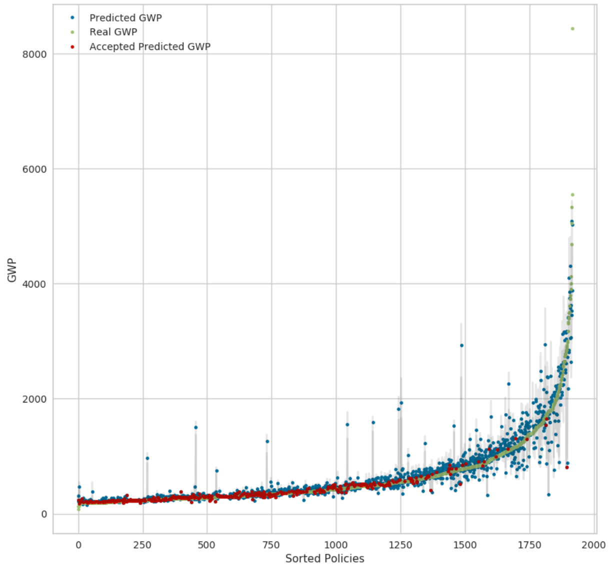

Figure 4: Results of the predicted premiums for SME Product on the validation set

Feature Name	Source	Description	Example
'FT_UEN_AGE'	ACRA	Age of the company	10.4
'FT_REGISTRATION_AGE'	ACRA	Age since registration	10.6
'FT_NO_OF_OFFICERS'	ACRA	Number of officers	5
'FT_POLICY_DURATION'	Data Lake	Policy cover duration	2.2
'FT_CLAIM_INC_TOT'	Data Lake	Total claim amount	0
'FT_PRIMARY_SSIC_CODE'	Data Lake	1st SSIC code in ACRA	0
'FT_SECONDARY_SSIC_CODE'	Data Lake	2nd SSIC code in ACRA	17
'FT_RISK_DURATION_LBH'	Data Lake	Cover period for risk LBH	791
'FT_TOTSIL_LBH'	Data Lake	Total Sum Insured for risk LBH	5000
TG_GWP (Target Variable)	Data Lake	Total GWP of policy	500

Table 1: List of features used for the Bond Product Machine Learning Model

Feature Name	**Source**	**Description**	**Example**
FT_UEN_AGE	ACRA	Age of the company	0.9
FT_REGISTRATION_AGE	ACRA	Age since registration	0.9
FT_NO_OF_OFFICERS	ACRA	Number of officers	4
FT_POLICY_DURATION	Data Lake	Policy cover duration	1
FT_RSKNO	Data Lake	Total Number of Risks	6
FT_CLAIM_AMT_TOT	Data Lake	Total Amount Paid of Claims	0
FT_CLAIM_INC_TOT	Data Lake	Total Amount of Claims	0
FT_CNTTYPE	Data Lake	Contract Type	2
FT_TRANTYPE	Data Lake	Transaction Type (Renewal or NB)	1
FT_OCCUP	Data Lake	Occupancy of company	2
FT_PRIMARY_SSIC_CODE	ACRA	1st SSIC code in ACRA	5
FT_SECONDARY_SSIC_CODE	ACRA	2nd SSIC code in ACRA	0
FT_CLAIM_AMT_LWC	Data Lake	Total claim amount paid for risk LWC	0
FT_CLAIM_INC_LPX	Data Lake	Total claim amount for risk LPX	0
FT_CLAIM_INC_LWC	Data Lake	Total claim amount for risk LWC	0
FT_CLAIM_NO_LWC	Data Lake	Total number of claims for risk LWC	0
FT_RISK_DURATION_APG	Data Lake	Cover period for risk APG	364
FT_RISK_DURATION_LMG	Data Lake	Cover period for risk LMG	0
FT_RISK_DURATION_LPX	Data Lake	Cover period for risk LPX	364
FT_RISK_DURATION_LWC	Data Lake	Cover period for risk LWC	364
FT_RISK_DURATION_PAA	Data Lake	Cover period for risk PAA	364
FT_RISK_DURATION_PBM	Data Lake	Cover period for risk PBM	364

Feature Name	Source	Description	Example
FT_RISK_DURATION_PCC	Data Lake	Cover period for risk PCC	364
FT_RISK_DURATION_PCI	Data Lake	Cover period for risk PCI	0
FT_RISK_DURATION_PFC	Data Lake	Cover period for risk PFC	0
FT_TOTSIL_APG	Data Lake	Total Sum Insured for risk APG	50000
FT_TOTSIL_LMG	Data Lake	Total Sum Insured for risk LMG	0
FT_TOTSIL_LPX	Data Lake	Total Sum Insured for risk LPX	500000
FT_TOTSIL_LWC	Data Lake	Total Sum Insured for risk LWC	72000
FT_TOTSIL_PAA	Data Lake	Total Sum Insured for risk PAA	100000
FT_TOTSIL_PBM	Data Lake	Total Sum Insured for risk PBM	3000
FT_TOTSIL_PCC	Data Lake	Total Sum Insured for risk PCC	25000
FT_TOTSIL_PCI	Data Lake	Total Sum Insured for risk PCI	0
FT_TOTSIL_PFC	Data Lake	Total Sum Insured for risk PFC	0
FT_RSKNO_APG	Data Lake	Number of risks of type APG	6
FT_RSKNO_LMG	Data Lake	Number of risks of type LMG	0
FT_RSKNO_LPX	Data Lake	Number of risks of type LPX	5
FT_RSKNO_LWC	Data Lake	Number of risks of type LWC	4
FT_RSKNO_PBM	Data Lake	Number of risks of type PBM	3
FT_RSKNO_PCC	Data Lake	Number of risks of type PCC	2
FT_RSKNO_PCI	Data Lake	Number of risks of type PCI	0
FT_RSKNO_PFC	Data Lake	Number of risks of type PFC	0
TG_GWP (Target Variable)	Data Lake	Total GWP of policy	500

Table 2: List of features used for the SME Product Machine Learning Model