Credit Card Approval Prediction

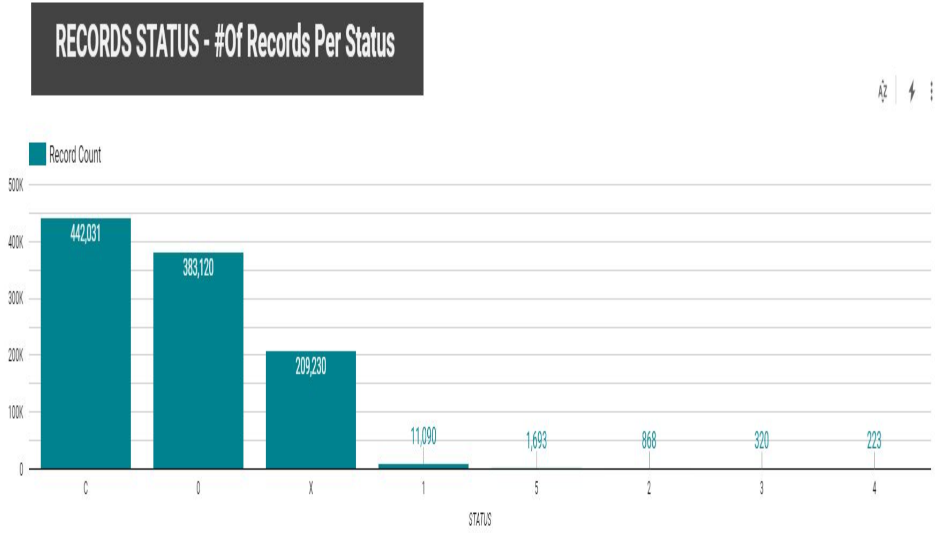
By Oussama Errabia

Sr. Data Scientist | MLOps GCP Developer

What To Expect :

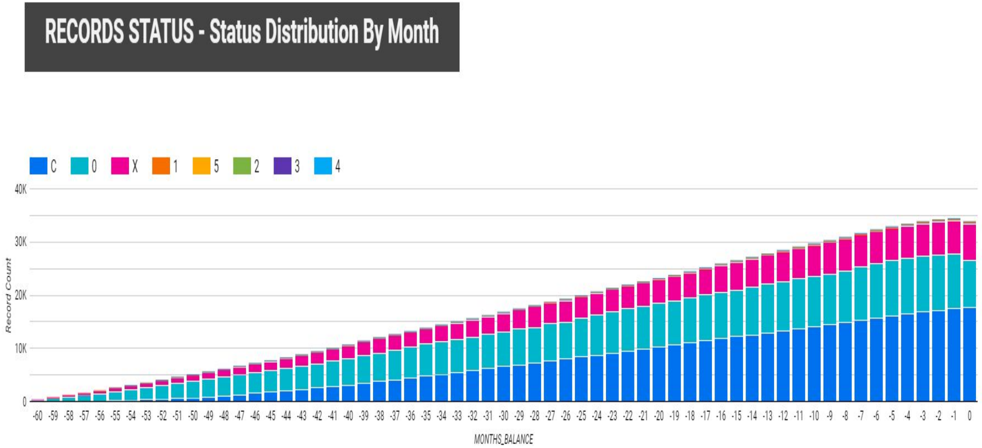
- Data Presentation
  - Univariate analysis
  - Bivariate analysis
  - Duplicated records
- Data Cleaning
  - Missing Values
  - Removing Duplication
- Data Preparation
  - Target definition
  - Features transformation (encoding)
  - Train/Val/Test split
  - Handling Imbalanced Data
- Data Validation - (Tensorflow Data Validation)
  - Train schema
  - Compare train/test/val distributions
- ML - logistic regression :
  - Training
  - Testing
  - Metrics - Precision/Recall/F1-Score/AUC
- LightGbm :
  - Training
  - Testing
  - Metrics - Precision/Recall/F1-Score/AUC
  - Model Interpretation (Using SHAP)
  - Model Calibration
  - Responsible AI

Univariate Analysis



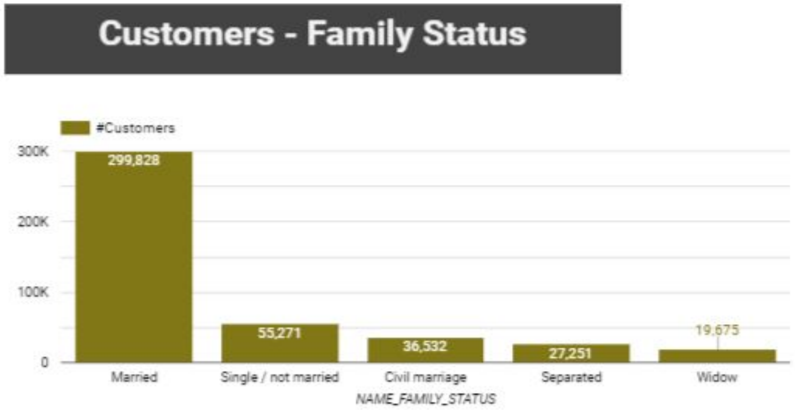
- we visualise the dominating status to be the C,0 and the X

Univariate Analysis



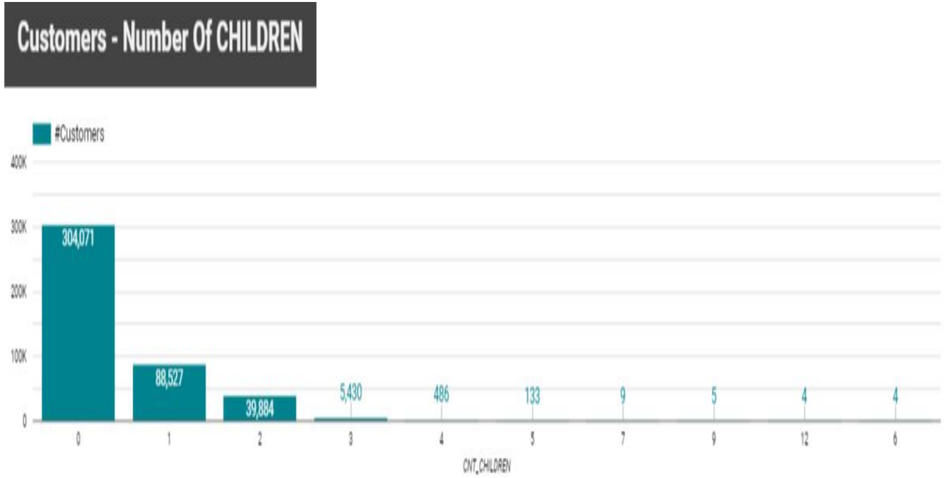
- we visualise a steady cumulative Month to Month.
- It seems the distribution across status almost remains the same from Month to Month.

Univariate Analysis



- we visualise most customers are Married.

Univariate Analysis

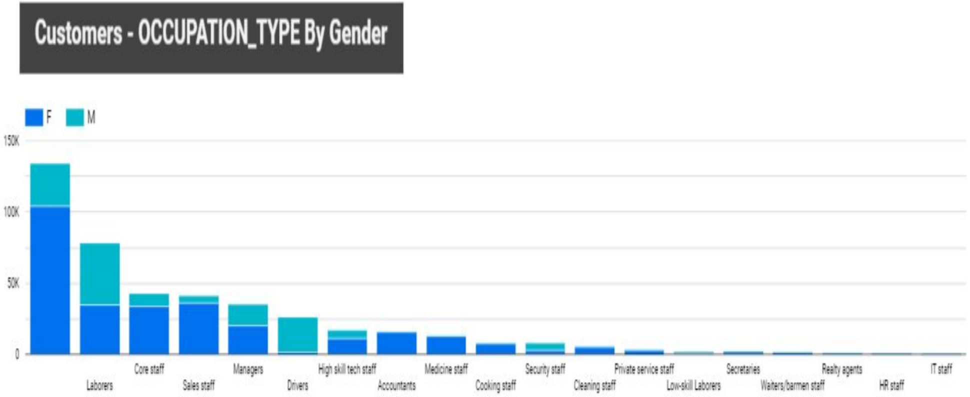


- we visualise most Customers have no Children

Univariate Analysis

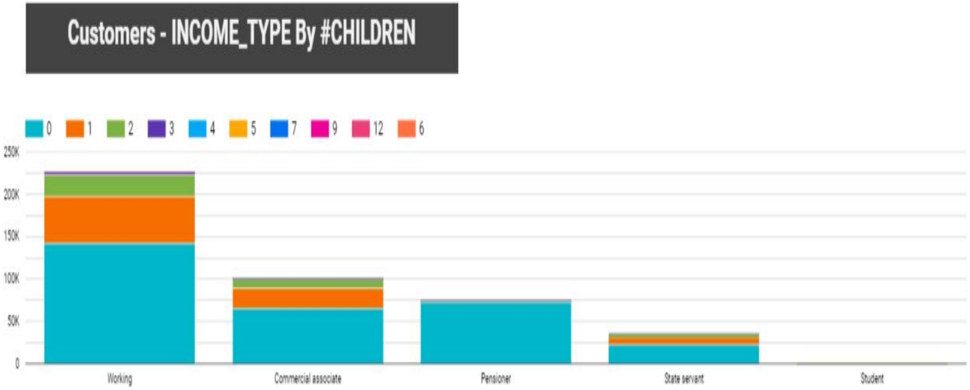
For more infos on the other features, please check the pdf report.

Bivariate Analysis



- we visualise that Sales Staff occupation is dominated by womens
- We visualise that Labores occupation is dominated by Men

Bivariate Analysis



- The interesting thing about this graph is that almost all Pensioner have No Children, which a bit surprising.

Bivariate Analysis

For more infos on the other Interactions, please check the pdf report.

Duplicated Records

Customers - Duplicated Records

	CODE_GEN...	FLA...	FLAG...	CNT_...	AMT_I...	NAME_...	NA...	NA...	NAM...	DAYS...	DAYS_...	FLAG_...	FLAG_...	FLAG_P...	FLAG_E...	OCCUPAT...	CNT_FAM_...	Record Count
1.	F	false	false	1	81000	State ser...	High...	Married	With p...	-11707	-2317	1	0	0	0	Core staff	3	115
2.	F	false	false	0	90000	Pensioner	High...	Widow	House...	-22791	365243	1	0	1	0	null	1	61
3.	F	false	true	0	157500	Commer...	Seco...	Single...	House...	-14055	-1986	1	0	0	0	Cooking staff	1	60
4.	M	false	true	0	135000	Commer...	Seco...	Civil ...	House...	-9975	-85	1	0	0	0	Laborers	2	59
5.	F	false	true	0	90000	Commer...	Seco...	Civil ...	House...	-11916	-1361	1	0	0	0	Laborers	2	59
6.	F	false	false	0	51750	Working	Seco...	Married	House...	-17166	-2514	1	0	0	0	Cleaning st...	2	56
7.	F	false	true	0	297000	Commer...	Seco...	Single...	Rente...	-15519	-3234	1	0	0	0	Laborers	1	55
8.	M	false	true	2	112500	Commer...	Seco...	Married	House...	-12715	-1085	1	0	0	0	Laborers	4	53
9.	M	true	true	1	153000	Working	Seco...	Married	House...	-8591	-343	1	0	0	1	Low-skill La...	3	53
10.	M	true	true	0	225000	Pensioner	Seco...	Married	House...	-17742	365243	1	0	0	0	null	2	53
11.	F	false	true	0	112500	Working	High...	Single...	House...	-10998	-2721	1	1	0	0	Core staff	1	51
12.	M	false	false	1	180000	Working	High...	Single...	House...	-15323	-3306	1	0	0	0	Managers	2	47
13.	M	true	true	1	292500	Commer...	Seco...	Married	House...	-13482	-717	1	0	0	0	Drivers	3	47
14.	F	false	true	0	225000	Working	High...	Married	House...	-10863	-955	1	0	1	1	Accountants	2	44
15.	F	false	true	0	180000	Pensioner	Seco...	Single...	House...	-20107	365243	1	0	1	0	null	1	42
16.	F	false	true	0	135000	Commer...	Seco...	Married	House...	-17532	-9642	1	1	1	0	High skill te...	2	41
17.	M	true	true	3	112500	Working	Seco...	Married	House...	-13341	-5276	1	0	0	0	Laborers	5	40
18.	F	false	true	0	112500	Working	Seco...	Married	House...	-13637	-3578	1	0	1	0	Laborers	2	39
19.	F	false	true	1	157500	State ser...	Seco...	Single...	House...	-12676	-1350	1	0	0	0	Waiters/bar...	2	37
20.	M	false	false	0	99000	Working	Seco...	Married	House...	-16291	-5069	1	0	0	0	Laborers	2	37
21.	F	false	true	0	382500	Working	Seco...	Separ...	House...	-14431	-2523	1	0	0	0	Laborers	1	37

1 - 100 / 90085

- By removing the ID from the Application dataset, we can clearly see that there are a lot of duplicated records, like the top one above is duplicated 115 times.
- There are only ~90k unique records (from an original ~440k which is 20%)

Data Cleaning :

Missing Values

- The only feature that had missing values was the Occupation Type feature (you can notice this from the pdf report)
- It was handled by replacing the missing value by NotDefined.

Data Cleaning :

Removing Duplicate

- Since all the training features (the X) will be generated from the application data, then it is a must to drop the duplicated values, and so we did.
- The credit_record dataset was used to generate the target (the Y) only, so not to introduce any data leakage.

Data Preparation :

Target Definition

- Based on the Vintage Analysis provided with the description of the dataset, we defined Bad Customers as customers with past due more than 60 days as it has the most adequate percentage (1.4%).

Data Preparation :

Features Transformations

In ML, features should be numeric and with meaningful magnitude, so :

- For Binary features : we coded them into 0 and 1s
- For multi-category features : we applied one hot encoding.
- We scaled the Continuous features (For Logistic Regression)

Data Preparation :

Train/Val/Test Split

We Split the data into 3 :

Train : it will be used to train the model

Val : it will be used for hyperparameters tuning

Test : it will be used to test and validate the the find model.

Data Preparation :

Handling Imbalanced Data

We managed to handle the extreme imbalance data using models parameters for both the logistic_regression and the lightgbm :

Logistic_regression : class_weight='balanced'

Lightgbm : scale_pos_weight.

Although google recommend downsampling and scale the positive weight in order to keep the models calibrated,however, given the size of the data, i chose to balance the classes using model params and then calibrate the output model

Data Validation :

Tensorflow Data Validation

1 - Motivation

Why Data validation ?

1 - We want to make sure the train/test/val have the same distribution across all features

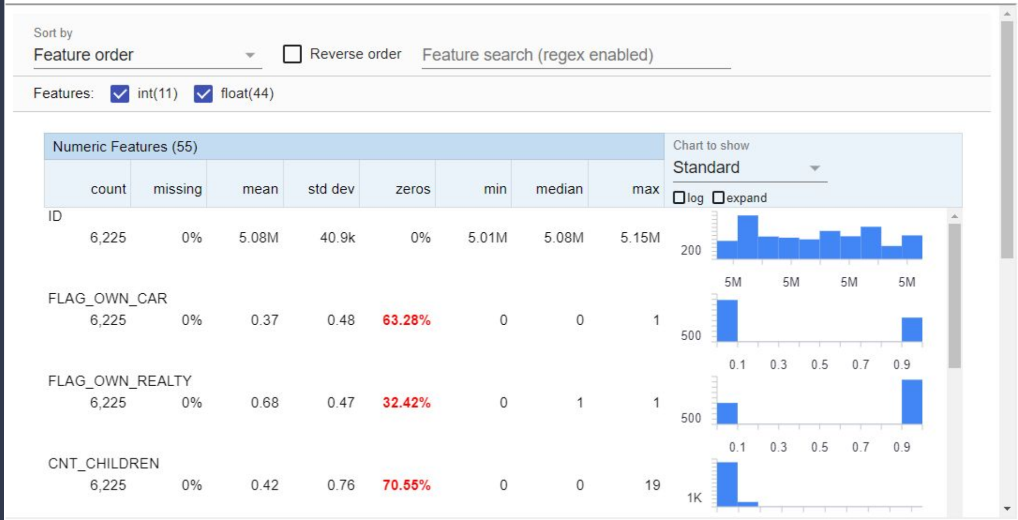
2 - we want to detect if there is any outliers

3 - we want to generate train schema to be used in production in our pipelines.

Data Validation :

Tensorflow Data Validation

1 - train stats

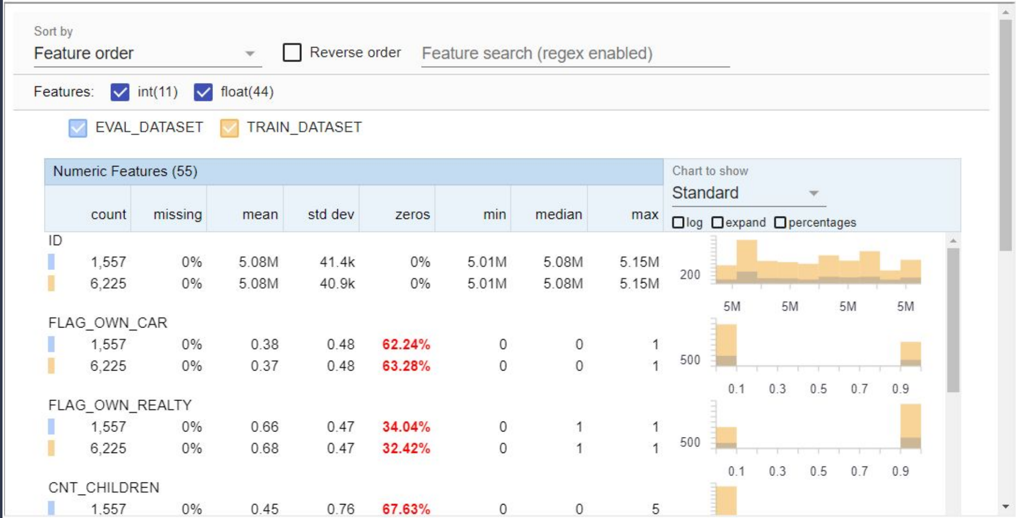


For more information, please check the data validation notebook

Data Validation :

Tensorflow Data Validation

1 - train/ val distribution

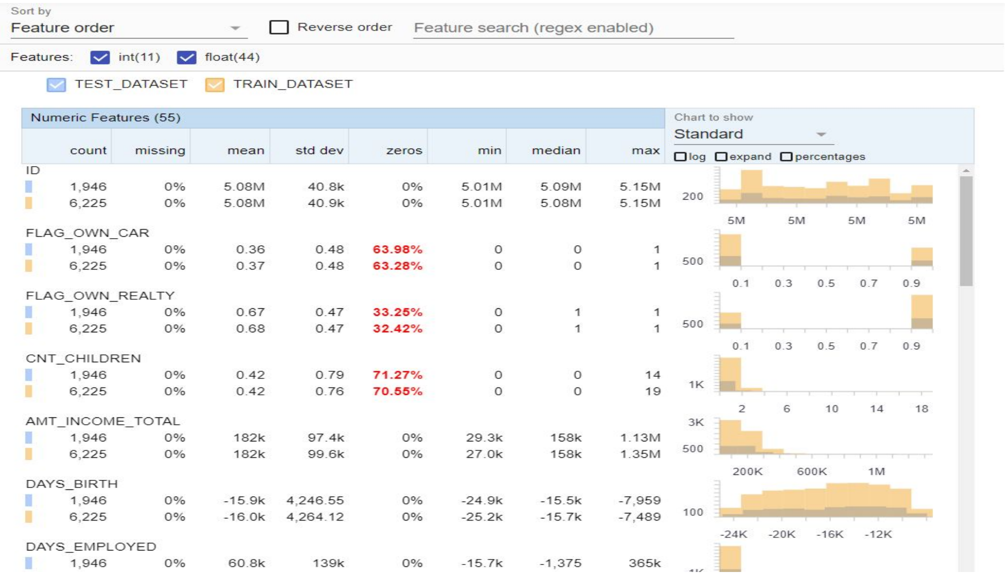


- We visualize that the train and val data have identical distribution across all features.
- For more information, please visit the notebook

Data Validation :

Tensorflow Data Validation

1 - train/ test distribution



	count	missing	mean	std dev	zeros	min	median	max
ID								
	1,946	0%	5.08M	40.8k	0%	5.01M	5.09M	5.15M
	6,225	0%	5.08M	40.9k	0%	5.01M	5.08M	5.15M
FLAG_OWN_CAR								
	1,946	0%	0.36	0.48	63.98%	0	0	1
	6,225	0%	0.37	0.48	63.28%	0	0	1
FLAG_OWN_REALTY								
	1,946	0%	0.67	0.47	33.25%	0	1	1
	6,225	0%	0.68	0.47	32.42%	0	1	1
CNT_CHILDREN								
	1,946	0%	0.42	0.79	71.27%	0	0	14
	6,225	0%	0.42	0.76	70.55%	0	0	19
AMT_INCOME_TOTAL								
	1,946	0%	182k	97.4k	0%	29.3k	158k	1.13M
	6,225	0%	182k	99.6k	0%	27.0k	158k	1.35M
DAYS_BIRTH								
	1,946	0%	-15.9k	4,246.55	0%	-24.9k	-15.5k	-7,959
	6,225	0%	-16.0k	4,264.12	0%	-25.2k	-15.7k	-7,489
DAYS_EMPLOYED								
	1,946	0%	60.8k	139k	0%	-15.7k	-1,375	365k

- We visualize that the train and test data have identical distribution across all features.
- For more information, please visit the notebook

Data Validation :

Tensorflow Data Validation

1 - Outliers Detection

```
anomalies = tfdv.validate_statistics(statistics=eval_stats, schema=schema)
tfdv.display_anomalies(anomalies)
```

No anomalies found.

- We also used the TFDV to detect outliers, which None are detected
- For more information, please visit the notebook

Logistic Regression

Train

We Trained 3 Logistic Regression Models :

1 - trained on all the transformed features generated from the application data.

2 - we used L1 regularization to detect non-useful features (which will have a coef of 0) and then we dropped those features to train the second model

3 - we used Correlation analysis to detect collinearity, as it hurts logistic regression model, so we removed highly correlated feature and we used the remaining ones to train the third model.

We also noticed a slight increase in the metrics we observe as we move from a model to the other.

Logistic Regression

Testing & Metrics

```
evaluate(lr_model,test_df,test_labels, 'TEST STATS')
---  TEST STATS ----
```

	Accuracy	Precision	Recall	ROC_auc	PR_auc
0	0.64851	0.065598	0.511364	0.604435	0.078663

```
evaluate(lr_model_updated,X_test,test_labels,'TEST STATS')
---  TEST STATS ----
```

	Accuracy	Precision	Recall	ROC_auc	PR_auc
0	0.64851	0.065598	0.511364	0.604435	0.078663

```
evaluate(lr_model_updated_2, X_test_updated, test_labels, "TEST STATS")
---  TEST STATS ----
```

	Accuracy	Precision	Recall	ROC_auc	PR_auc
0	0.651593	0.066176	0.511364	0.604007	0.078651

- The above 3 tables shows the metrics of the 3 models from 1 to 3 respectively.
- All 3 models have a recall in the range of 0.5x and a precision of 0.0x, which definitely a clear indicator that the models are not doing great which can be explained by many reasons the obvious on is the data is too noisy.

LightGbm

Train

We trained light gbm model on the same data used to train the logistic regression model

The model also suffered from the noise in the data which can be noticed from how easy the model can overfit with almost perfect train score with very bad test scores.

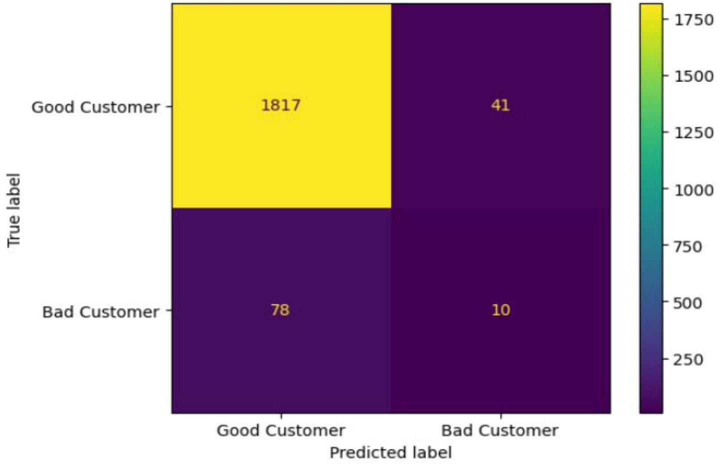
LightGbm

Testing & Metrics

1 - Confusion Matrix

```
In [31]: plot_confusion_matrix(test_labels,test_df,"---TEST --")
```

```
Best Threshold=0.7684093727512775, F-Score : 0.14388489208633096
```



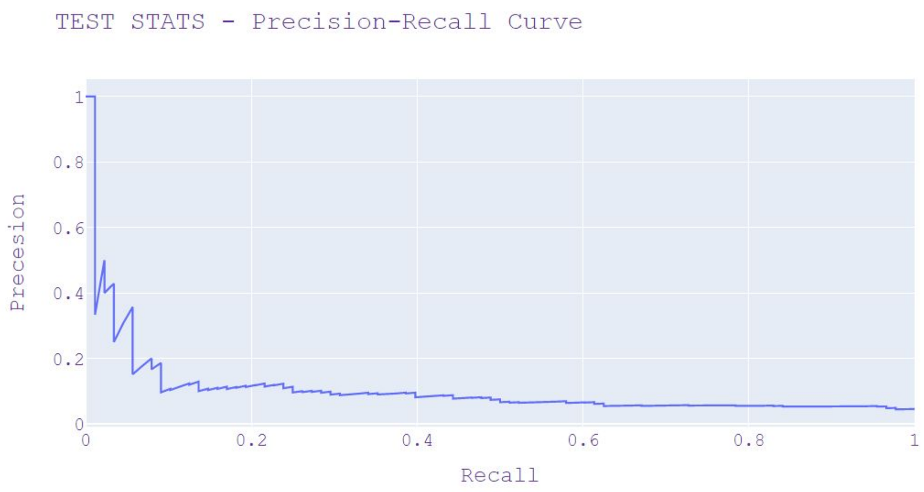
As we mentioned earlier, the model is doing no so great on the test data, even low in performance compared to logistic regression, this due to the high amount of noise in the data that the model easily learns from.

All of this is confirmed by the above confusion matrix.

LightGbm

Testing & Metrics

1 - Precision-Recall Curve



- We usually use the precision-recall cruve to find the best tradeoff between the precision and recall.
- We observe from the above curve that the model is not doing so great on the test data.

LightGbm

Testing & Metrics

1 - Precision / Recall / AUC

--- TEST STATS ----

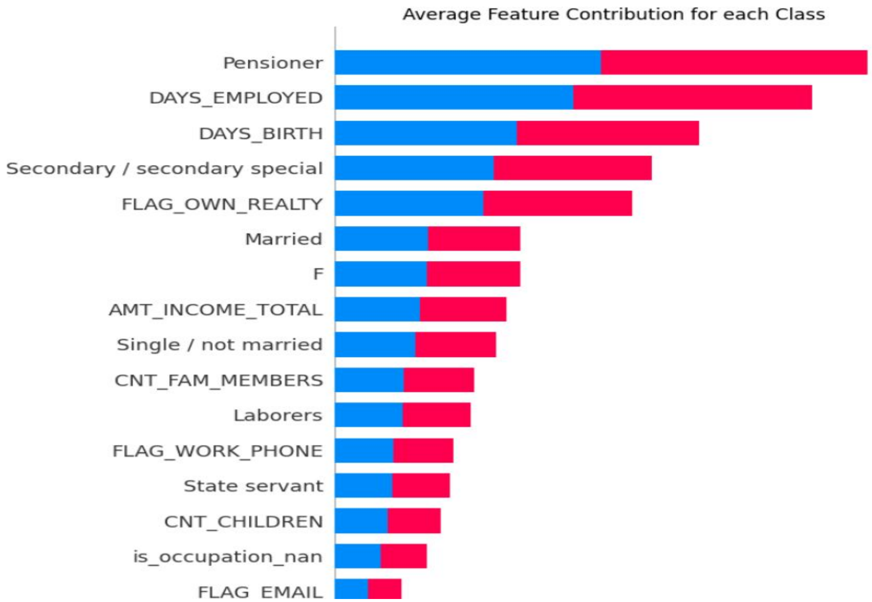
	Accuracy	Precision	Recall	ROC_auc	PR_auc
0	0.888489	0.122807	0.238636	0.640847	0.105659

- We observe an auc of 0.64, which mean the model is ok in ranking from good customers to bad customers, however, in our case, we care more about recall and precision
- The model is performing not so great on both the recall and precision

LightGbm

Model Interpretation

1 - Feature Importance using SHAP

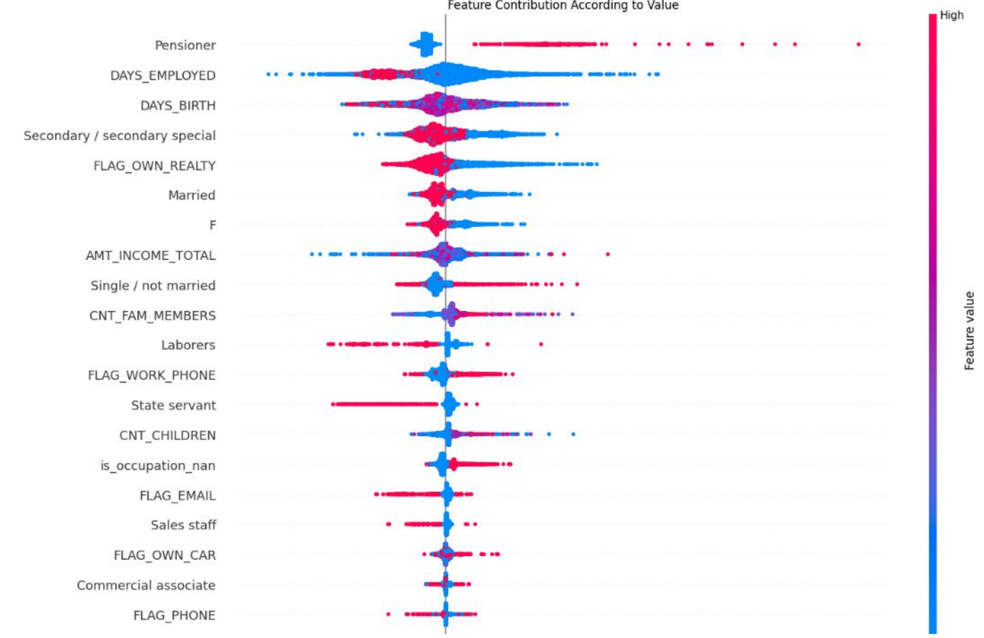


- We used the shap package to observe the features importance given our lightgbm model.
- The above graph shows the top important features.

LightGbm

Model Interpretation

2 - Feature Contribution By Value

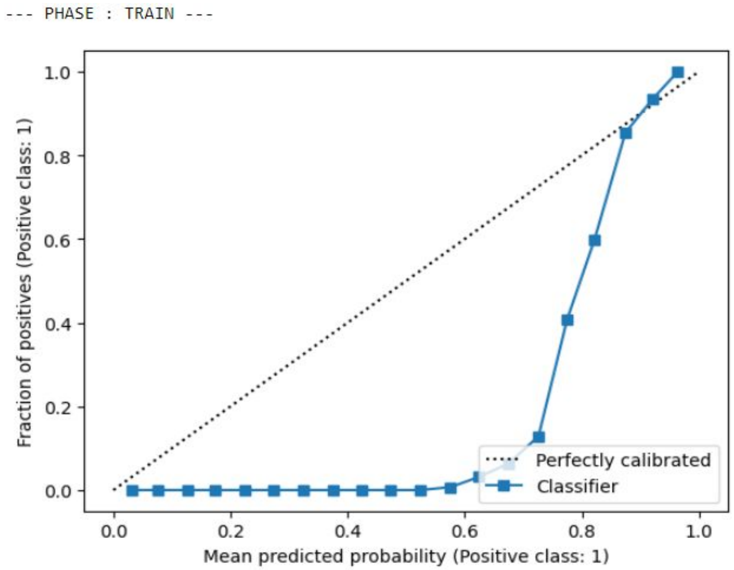


There is a lot to say for analyzing the above graph, and just to mention a few :

- Low values for Days Employed seems to push the customers to the bad side, which is expected, however, what is not expected is to see some low values push the customer to the good side

LightGbm

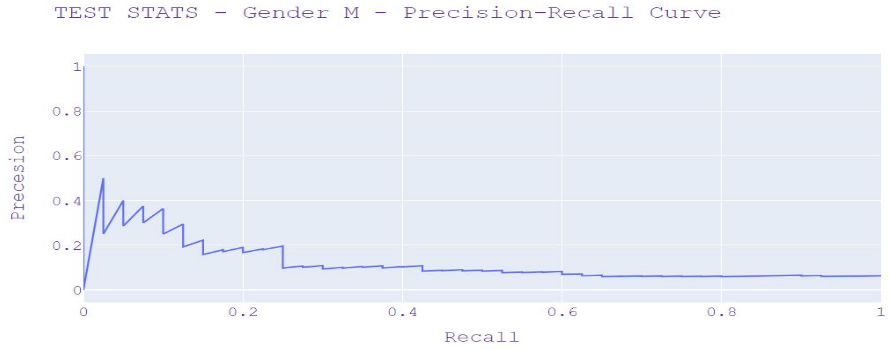
Model Calibration



- We usually care about Calibration in order to interpret the output of our model as probabilities.
- The above graph is the calibration curve of our lightgbm model, it needs extra post processing for calibration, like Istonic Calibration.

LightGbm

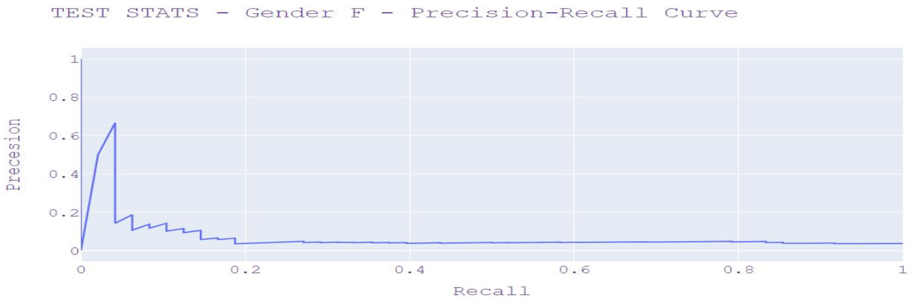
Responsible AI



--- TEST STATS - Gender M ----

Out[16]:

	Accuracy	Precision	Recall	ROC_auc	PR_auc
0	0.93816	0.0	0.0	0.588283	0.131621



--- TEST STATS - Gender F ----

[17]:

	Accuracy	Precision	Recall	ROC_auc	PR_auc
0	0.961808	0.0	0.0	0.56665	0.076706

Given we are using Gender as a feature, it is important to check for bias by computing the models performance by Gender.

Final Notes

- The PDF report was generated using Looker Studio, the data was imported into BigQuery first.
- The trained models seems to report an OK AUC even though they are optimized for F1-score (recall and precision), given that, it can be useful to have a good ranking of customers.
- I did not redo the Vintage Analysis given the fact it is already done and reported in the description of the dataset

Thank You