Capstone Project - Credit Card Acquisition

Maggie Wang

Objective

Drive credit card pick up **cost effectively** via Cross-sell existing customer

Mean Lead% : 23%

Data source: Kaggle

What I’ve done

1. EDA
2. Data Cleansing
3. Modeling Training
4. Output Prediction
5. Business Case

1 EDA

```
df_train.shape
```

```
(245725, 11)
```

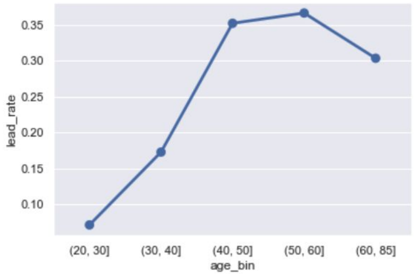
	ID	Gender	Age	Region_Code	Occupation	Channel_Code	Vintage	Credit_Product	Avg_Account_Balance	Is_Active	Is_Lead
0	NNVBBKZB	Female	73	RG268	Other	X3	43	No	1045696	No	0
1	IDD62UNG	Female	30	RG277	Salaried	X1	32	No	581988	No	0
2	HD3DSEMC	Female	56	RG268	Self_Employed	X3	26	No	1484315	Yes	0
3	BF3NC7KV	Male	34	RG270	Salaried	X1	19	No	470454	No	0
4	TEASRWXV	Female	30	RG282	Salaried	X1	33	No	886787	No	0

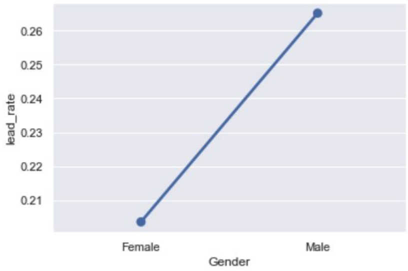
Total 10 Variables

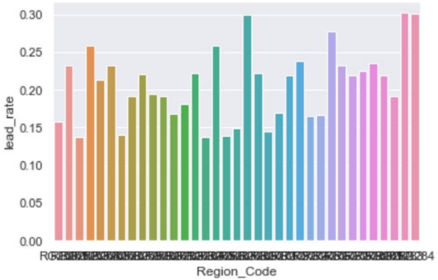
{ What are the variables that'll influence the probability of lead? }

Calculate the lead rate of each variables

Lead Rate per attributes - Demo







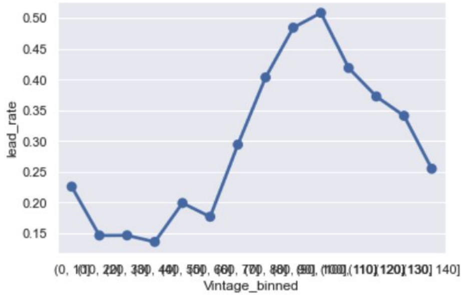
Age: Customer at age 40+ has higher % of lead at 35%+

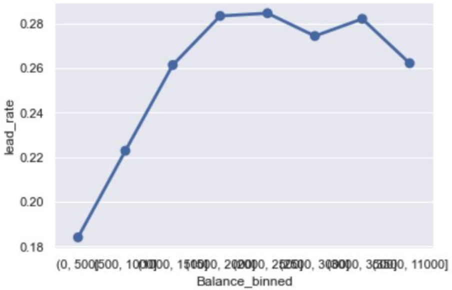
Gender: Male is more likely to want to apply

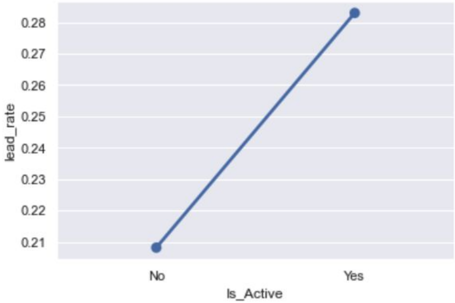
Region: Selective Region has higher % of lead as high as 30% (vs. under 15% for lower quartile)

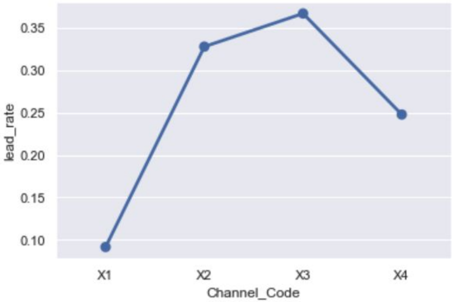
Occupation: Entrepreneur has significantly higher intend (66%) and Self-Employed has 28% lead intention

Lead Rate per attributes - Relationship with Banks









Missing variables - fillna

2 Data Cleansing - Include attributes that'll influence the probability of lead
Get Dummies

```
df_train_n
```

	Is_Lead	Gender_Female	Gender_Male	age_bin_(20, 30]	age_bin_(30, 40]	age_bin_(40, 50]	age_bin_(50, 60]	age_bin_(60, 85]	Region_Code_(0, 10]	Region_Code_(10, 20]	...	Balance_binned_(0, 500]	Balance_binned_(500, 1000]	Balance_binned_
0	0	1	0	0	0	0	0	1	0	0	...	0	0	
1	0	1	0	1	0	0	0	0	0	0	...	0	1	
2	0	1	0	0	0	0	1	0	0	0	...	0	0	
3	0	0	1	0	1	0	0	0	0	1	...	1	0	
4	0	1	0	1	0	0	0	0	0	0	...	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
245720	0	0	1	0	0	0	1	0	0	0	...	0	0	
245721	0	0	1	1	0	0	0	0	0	1	...	0	1	
245722	0	1	0	1	0	0	0	0	0	1	...	0	1	
245723	0	1	0	1	0	0	0	0	0	0	...	1	0	
245724	0	0	1	1	0	0	0	0	0	0	...	0	0	

245725 rows × 77 columns

3 Model Training

Logistic Regression

	precision	recall	f1-score	support
0	0.87	0.96	0.91	37594
1	0.81	0.52	0.64	11551
accuracy			0.86	49145
macro avg	0.84	0.74	0.77	49145
weighted avg	0.85	0.86	0.85	49145

Gaussian Naive Bayers

	precision	recall	f1-score	support
0	0.90	0.72	0.80	37594
1	0.45	0.74	0.56	11551
accuracy			0.72	49145
macro avg	0.67	0.73	0.68	49145
weighted avg	0.79	0.72	0.74	49145

Random Forest Tree

	precision	recall	f1-score	support
0	0.88	0.93	0.90	37594
1	0.71	0.58	0.63	11551
accuracy			0.84	49145
macro avg	0.79	0.75	0.77	49145
weighted avg	0.84	0.84	0.84	49145

4 Output Prediction

Prepare the dataset of test file and calculate the Propensity % per User based on Logistic Regression

```python
prospects['propensity'] = model.predict_proba(prospects)[:, 1]
print(prospects.head(10))
```

```
   Gender_Female  Gender_Male  age_bin_(20, 30]  age_bin_(30, 40]  \
0              0            1                 1                 0
1              0            1                 0                 0
2              0            1                 0                 1
3              0            1                 1                 0
4              1            0                 1                 0
5              0            1                 0                 0
6              1            0                 0                 0
7              1            0                 1                 0
8              0            1                 0                 0
9              1            0                 0                 0

   age_bin_(40, 50]  age_bin_(50, 60]  age_bin_(60, 85]  Region_Code_(0, 10]  \
0                 0                 0                 0                    0
1                 1                 0                 0                    0
2                 0                 0                 0                    0
3                 0                 0                 0                    0
4                 0                 0                 0                    0
5                 0                 1                 0                    0
6                 0                 0                 1                    0
7                 0                 0                 0                    0
8                 1                 0                 0                    0
9                 0                 1                 0                    0

   Region_Code_(10, 20]  Region_Code_(20, 30]  ...  \
0                     0                     1  ...
1                     0                     0  ...
2                     1                     0  ...
3                     0                     0  ...
4                     1                     0  ...
5                     0                     0  ...
6                     0                     0  ...
7                     0                     0  ...
8                     0                     0  ...
9                     0                     0  ...

   Balance_binned_(500, 1000]  Balance_binned_(1000, 1500]  \
0                           1                            0
1                           1                            0
2                           0                            0
3                           1                            0
4                           1                            0
5                           0                            0
6                           0                            1
7                           1                            0
8                           0                            1
9                           0                            0

   Balance_binned_(1500, 2000]  Balance_binned_(2000, 2500]  \
0                            0                            0
1                            0                            0
2                            0                            0
3                            0                            0
4                            0                            0
5                            0                            0
6                            0                            0
7                            0                            0
8                            0                            0
9                            1                            0

   Balance_binned_(2500, 3000]  Balance_binned_(3000, 3500]  \
0                            0                            0
1                            0                            0
2                            0                            0
3                            0                            0
4                            0                            0
5                            0                            0
6                            0                            0
7                            0                            0
8                            0                            0
9                            0                            0

   Balance_binned_(3500, 11000]  Is_Active_No  Is_Active_Yes  propensity
0                             0             1              0    0.015758
1                             0             1              0    0.072589
2                             0             1              0    0.079244
3                             0             1              0    0.014817
4                             0             1              0    0.015613
5                             1             1              0    0.100100
6                             0             1              0    0.785400
7                             0             1              0    0.045848
8                             0             0              1    0.478397
9                             0             1              0    0.266197

[10 rows x 78 columns]
```

4 Output Prediction

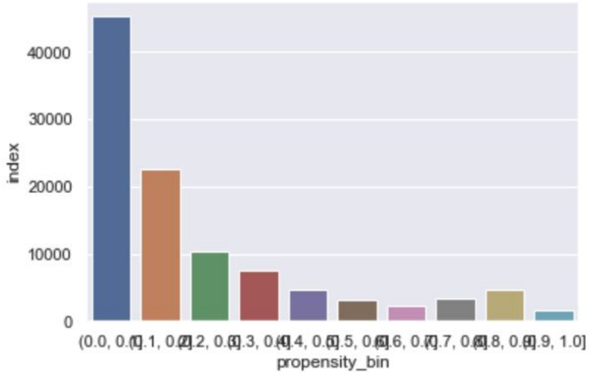
Binning on the propensity → **GroupBy** → **Graph**

```python
bins = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5,0.6,0.7,0.8,0.9,1.0]
Propensity["propensity_bin"] = pd.cut(Propensity['propensity'], bins=bins)
Propensity
```

	index	propensity	propensity_bin
0	0	0.015758	(0.0, 0.1]
1	1	0.072589	(0.0, 0.1]
2	2	0.079244	(0.0, 0.1]
3	3	0.014817	(0.0, 0.1]
4	4	0.015613	(0.0, 0.1]
...	...	...	...
105307	105307	0.455697	(0.4, 0.5]
105308	105308	0.192293	(0.1, 0.2]
105309	105309	0.119519	(0.1, 0.2]
105310	105310	0.579042	(0.5, 0.6]
105311	105311	0.691785	(0.6, 0.7]

105312 rows × 3 columns

	propensity_bin	index	propensity
0	(0.0, 0.1]	45285	0.056655
1	(0.1, 0.2]	22548	0.146921
2	(0.2, 0.3]	10260	0.245678
3	(0.3, 0.4]	7504	0.346164
4	(0.4, 0.5]	4719	0.448373
5	(0.5, 0.6]	3080	0.545001
6	(0.6, 0.7]	2292	0.664439
7	(0.7, 0.8]	3359	0.751456
8	(0.8, 0.9]	4595	0.846699
9	(0.9, 1.0]	1670	0.923325



5 Business Case Application

	propensity_bin	index
0	(0.0, 0.1]	45285
1	(0.1, 0.2]	22548
2	(0.2, 0.3]	10260
3	(0.3, 0.4]	7504
4	(0.4, 0.5]	4719
5	(0.5, 0.6]	3080
6	(0.6, 0.7]	2292
7	(0.7, 0.8]	3359
8	(0.8, 0.9]	4595
9	(0.9, 1.0]	1670

<30% - Low Propensity

30%-70% - Medium Propensity

+70% - High Propensity

5 Business Case Application

Assume it costs $50 to reach each customer….

	propensity_bin	propensity	Total_Customers	Cost
0	(0.0, 0.3]	0.107552	78093	3904650
1	(0.3, 0.7]	0.449843	17595	879750
2	(0.7, 1.0]	0.826753	9624	481200

It will costs total of $5,265,600 to reach full base.

5 Business Case Application

Assume it costs $50 to reach each customer....

	propensity_bin	propensity	Total_Customers	Cost
0	(0.0, 0.3]	0.107552	78093	3904650
1	(0.3, 0.7]	0.449843	17595	879750
2	(0.7, 1.0]	0.826753	9624	481200

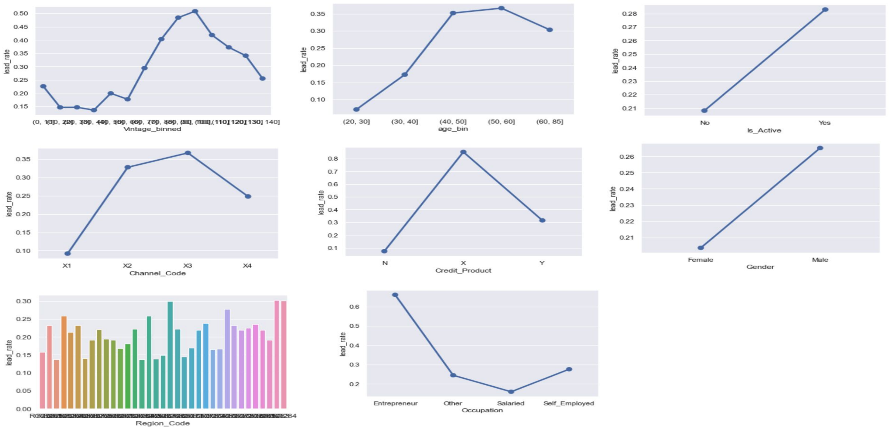
It will costs total of $5,265,600 to acquired 24,271 Credit Cards at CPL$217.

	propensity_bin	propensity	Total_Customers	Cost	Estimated_Leads	CPL
0	(0.0, 0.3]	0.107552	78093	3904650	8399.034990	464.892694
1	(0.3, 0.7]	0.449843	17595	879750	7914.987324	111.149894
2	(0.7, 1.0]	0.826753	9624	481200	7956.673450	60.477535

If only invest on High & Medium Propensity.....

It will costs total of $1,360,950 to acquired 15,872 (-35% vs. invest in Total) Credit Cards at CPL$86 (-60% vs. invest in Total).

Conclusion



Instead of selecting customers based on different attributes, it's more beneficial for business to combine the attributes and apply predictive modelling to identify the most responsive customers.

Thank you.