MDS6212 Fintech Theory and Practice: Week 1 Assignment

220041006 ***Yihang Li***

Data confidential requirements:

- Do not share datasets with others
- Do not post datasets on the Web site
- Datasets can only be used to work on assignments of MDS6212, M.Sc. in Data Science, CUHK-Shenzhen.

Datasets:

- Week 1 Data.csv
- Week 1 Variable descriptions.xlsx

> Reference, Dai, Lili, Jianlei Han, Jing Shi, and Bohui Zhang, 2020, Digital Footprints as Collateral for Debt Collection, working paper.

Q1

1) Present a table of summary statistics for the key variables including the borrower's age the borrower has a frequent contact, approval dummy, and delinquency dummy

	age	instalments_amount	nominalrates	creditlevelasbuyer	tencentscore	gaodescore
count	5000.000000	5000.000000	4997.000000	4031.000000	5000.000000	5000.000000
mean	27.675400	406201.420000	0.276058	53.119077	58.608168	0.201975
std	8.326146	130623.360240	0.085912	108.629757	14.218112	0.076724
min	18.000000	50000.000000	0.130080	0.000000	9.000000	0.023518
25%	21.000000	320000.000000	0.204560	0.000000	53.888889	0.192094
50%	25.000000	398000.000000	0.204579	14.000000	60.200000	0.192094
75%	32.000000	498000.000000	0.359347	58.000000	65.258929	0.192094
max	56.000000	869000.000000	0.494185	1830.000000	98.000000	0.732120

	gender	highcontact	deal	default
False	4267	2539	2793	1280
True	733	2461	2207	925

Important step: filling missing values of default. (See in *.html)

Q2

2) Perform a logit regression and examine the relation between the delinquency likelihood and credit scores

Using LogisticRegression from sklearn

In [32]:
```
q2_X = Key_Data[['creditlevelasbuyer', 'tencentscore', 'gaodescore']]
q2_y = Key_Data['default']
q2_X = scaler.fit(q2_X).transform(q2_X)
```

In [33]:
```
q2_logit = LogisticRegression().fit(q2_X, q2_y.astype(bool))
```

score:float

- Mean accuracy of self.predict(X) wrt. y.

In [34]:
```
q2_logit.score(q2_X, q2_y.astype(bool))
```

Out[34]:
```
0.7217671878878134
```

In [35]:
```
q2_logit.coef_
```

Out[35]:
```
array([[0.10413267, 0.31336157, 0.13033775]])
```

Using Logit from statsmodels

In [36]:
```
import statsmodels.api as sm
```

In [37]:
```
q2_X = Key_Data[['creditlevelasbuyer', 'tencentscore', 'gaodescore']]
q2_y = Key_Data['default'].map({True:1, False:0})
q2_X = scaler.fit(q2_X).transform(q2_X)
q2_X = sm.add_constant(q2_X)
```

In [38]:
```
q2_model = sm.Logit(q2_y, q2_X)
q2_result = q2_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.580266
         Iterations 5
```

In [39]:
```
q2_result.summary()
```

Out[39]:

Logit Regression Results

Dep. Variable:	default	**No. Observations:**	4029
Model:	Logit	**Df Residuals:**	4025
Method:	MLE	**Df Model:**	3
Date:	Sat, 26 Sep 2020	**Pseudo R-squ.:**	0.01901
Time:	20:28:39	**Log-Likelihood:**	-2337.9
converged:	True	**LL-Null:**	-2383.2
Covariance Type:	nonrobust	**LLR p-value:**	1.605e-19

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.9781	0.036	-27.192	0.000	-1.049	-0.908
x1	0.1044	0.035	2.979	0.003	0.036	0.173
x2	0.3138	0.038	8.161	0.000	0.238	0.389
x3	0.1305	0.035	3.765	0.000	0.063	0.198

Q3

3) Perform a logit regression and examine the relation between the loan approval likelihood and credit scores

Using LogisticRegression from sklearn

In [40]:
```
q3_X = Key_Data[['creditlevelasbuyer', 'tencentscore', 'gaodescore']]
q3_y = Key_Data['deal']
q3_X = scaler.fit(q3_X).transform(q3_X)
```

In [41]:
```
q3_logit = LogisticRegression().fit(q3_X, q3_y.astype(bool))
```

In [42]:
```
q3_logit.score(q3_X, q3_y.astype(bool))
```

Out[42]:
```
0.6443286175229586
```

In [43]:
```
q3_logit.coef_
```

Out[43]:
```
array([[ 0.20166668, -0.39714435, -0.15879942]])
```

Using Logit from statsmodels

In [44]:
```
q3_X = Key_Data[['creditlevelasbuyer', 'tencentscore', 'gaodescore']]
q3_y = Key_Data['deal']
q3_X = scaler.fit(q3_X).transform(q3_X)
q3_X = sm.add_constant(q3_X)
```

In [45]:
```
q3_model = sm.Logit(q3_y, q3_X)
q3_result = q3_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.658019
         Iterations 5
```

In [46]:
```
q3_result.summary()
```

Out[46]:

Logit Regression Results

Dep. Variable:	deal	**No. Observations:**	4029
Model:	Logit	**Df Residuals:**	4025
Method:	MLE	**Df Model:**	3
Date:	Sat, 26 Sep 2020	**Pseudo R-squ.:**	0.04411
Time:	20:28:39	**Log-Likelihood:**	-2651.2
converged:	True	**LL-Null:**	-2773.5
Covariance Type:	nonrobust	**LLR p-value:**	9.324e-53

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.1993	0.033	-6.098	0.000	-0.263	-0.135
x1	0.2019	0.039	5.182	0.000	0.126	0.278
x2	-0.3976	0.035	-11.445	0.000	-0.466	-0.329
x3	-0.1589	0.035	-4.580	0.000	-0.227	-0.091

Q4

4) Perform a logit regression and examine the relation between the loan approval likelihood and the dummy whether the borrower has a frequent contact

Using LogisticRegression from sklearn

In [47]:
```
q4_X = Key_Data['highcontact']
q4_y = Key_Data['deal']
q4_X = scaler.fit(q4_X.values.reshape(-1, 1)).transform(q4_X.values.reshape(-1, 1))
```

In [48]:
```
q4_logit = LogisticRegression().fit(q4_X, q4_y.astype(bool))
```

In [49]:
```
q4_logit.score(q4_X, q4_y.astype(bool))
```

Out[49]:
```
0.5487714072970961
```

In [50]:
```
q4_logit.coef_
```

Out[50]:
```
array([[0.08461788]])
```

Using Logit from statsmodels

In [51]:
```
q4_X = Key_Data['highcontact']
q4_y = Key_Data['deal']
q4_X = scaler.fit(q4_X.values.reshape(-1, 1)).transform(q4_X.values.reshape(-1, 1))
q4_X = sm.add_constant(q4_X)
```

In [52]:
```
q4_model = sm.Logit(q4_y, q4_X)
q4_result = q4_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.687495
         Iterations 4
```

In [53]:
```
q4_result.summary()
```

Out[53]:

Logit Regression Results

Dep. Variable:	deal	**No. Observations:**	4029
Model:	Logit	**Df Residuals:**	4027
Method:	MLE	**Df Model:**	1
Date:	Sat, 26 Sep 2020	**Pseudo R-squ.:**	0.001289
Time:	20:28:39	**Log-Likelihood:**	-2769.9
converged:	True	**LL-Null:**	-2773.5
Covariance Type:	nonrobust	**LLR p-value:**	0.007503

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.1961	0.032	-6.187	0.000	-0.258	-0.134
x1	0.0847	0.032	2.672	0.008	0.023	0.147