Statistics Methods in Finance
Homework 7

0786009 廖家鴻

DUE 2020/12/17 00:00

Outline (HW7 questions)

1.(30%) Logistic regression for Y=0 (no damage) and Y>0 (some damage)

2.(40%) Multinomial logistic regression for the damage score

3.(30%) Ordinary logistic regression for the damage score

1. Logistic regression for Y=0 and Y>0

Before answering to the 3 questions, I split the dataset into training-data and test-data. (training : test = 80 : 20)

```
df2 = df.copy()
X2 = df2.drop(['Damage'], axis=1)
y2 = df2['Damage']
X2_train, X2_test, y2_train, y2_test = sklearn.model_selection.train_test_split(X2, y2, test_size = 0.20)
```

```
'''Method2: Use statsmodels'''
import statsmodels.api as sm
LR1_sm = sm.Logit(y1_train, X1_train).fit()
LR1_sm.summary()
yhat = LR1_sm.predict(X1_test)
pred_LR1_sm = list(map(round, yhat))
print(pred_LR1_sm)
```

Logit Regression Results

Dep. Variable:	Damage	No. Observations:	128
Model:	Logit	Df Residuals:	124
Method:	MLE	Df Model:	3
Date:	Tue, 15 Dec 2020	Pseudo R-squ.:	0.1057
Time:	20:29:22	Log-Likelihood:	-60.137
converged:	True	LL-Null:	-67.241
Covariance Type:	nonrobust	LLR p-value:	0.002634

	coef	std err	z	P>|z|	[0.025	0.975]
Date	-0.1153	0.594	-0.194	0.846	-1.279	1.049
Temp	-0.0209	0.053	-0.395	0.693	-0.125	0.083
Conct	0.0021	0.003	0.658	0.511	-0.004	0.009
Count	0.3630	0.109	3.335	0.001	0.150	0.576

From the left result table, we can find that only the coefficient of "Count" is significant.

The prediction results of test data are shown below:

```
In [227]: confusion_matrix(y1_test, pred_LR1_sm)
Out[227]:
array([[ 0,  4],
       [ 0, 28]], dtype=int64)

In [228]: accuracy_score(y1_test, pred_LR1_sm)
Out[228]: 0.875
```

2. Multinomial logistic regression for the damage score

MNLogit Regression Results

Dep. Variable:	Damage	No. Observations:	128
Model:	MNLogit	Df Residuals:	108
Method:	MLE	Df Model:	16
Date:	Tue, 15 Dec 2020	Pseudo R-squ.:	0.05582
Time:	22:56:08	Log-Likelihood:	-194.04
converged:	True	LL-Null:	-205.51
Covariance Type:	nonrobust	LLR p-value:	0.1153

Damage=1	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.8410	1.195	-1.540	0.124	-4.184	0.502
Date	-0.1975	0.805	-0.245	0.806	-1.775	1.380
Temp	-0.0440	0.074	-0.593	0.553	-0.190	0.101
Conct	0.0027	0.005	0.582	0.560	-0.006	0.012
Count	0.3651	0.122	3.001	0.003	0.127	0.604

Damage=2	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.6004	1.160	-0.517	0.605	-2.875	1.674
Date	-0.0003	0.797	-0.000	1.000	-1.563	1.562
Temp	0.0494	0.077	0.641	0.522	-0.102	0.200
Conct	0.0019	0.005	0.394	0.693	-0.007	0.011
Count	0.4053	0.123	3.282	0.001	0.163	0.647

Damage=3	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.2498	1.055	-0.237	0.813	-2.317	1.818
Date	-0.3604	0.748	-0.482	0.630	-1.826	1.105
Temp	-0.0209	0.069	-0.303	0.762	-0.156	0.114
Conct	-0.0004	0.004	-0.088	0.930	-0.009	0.008
Count	0.2361	0.124	1.911	0.056	-0.006	0.478

Damage=4	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.2319	1.107	-0.210	0.834	-2.402	1.938
Date	-0.4078	0.776	-0.525	0.599	-1.929	1.113
Temp	0.0193	0.074	0.260	0.795	-0.126	0.164
Conct	0.0011	0.005	0.242	0.809	-0.008	0.010
Count	0.3906	0.122	3.206	0.001	0.152	0.629

```
'''Method2: Use statsmodels'''
LR2_sm=sm.MNLogit(y2_train,sm.add_constant(X2_train))
result=LR2_sm.fit()
stats1=result.summary()
```

From the left result table, we can find that only the coefficients of "Count" are relatively significant.

The prediction results of test data are shown below:

```
In [223]: confusion_matrix(y2_test, pred_LR2_sm)
Out[223]:
array([[3, 0, 0, 0, 1],
       [1, 0, 2, 2, 4],
       [3, 2, 0, 3, 1],
       [2, 0, 0, 2, 0],
       [1, 2, 0, 2, 1]], dtype=int64)

In [224]: accuracy_score(y2_test, pred_LR2_sm)
Out[224]: 0.1875
```

3. Ordinary logistic regression for the damage score

```
'''method 1-A'''
from statsmodels.miscmodels.ordinal_model import OrderedModel
LR3_sm = OrderedModel.from_formula("Damage ~ 0 + Date + Temp + Conct + Count", df3_train, distr='logit')
LR3_sm_ord = LR3_sm.fit(method='bfgs')
LR3_sm_ord.summary()
pred_LR3_sm_ord = LR3_sm_ord.model.predict(LR3_sm_ord.params, exog=X2_test[['Date', 'Temp', 'Conct', 'Count']])
# print(pred_LR2_sm_ord)
pred_LR3_sm_ord = pred_LR3_sm_ord.argmax(1)
print(pred_LR3_sm_ord)
```

OrderedModel Results

Dep. Variable:	Damage	Log-Likelihood:	-202.60
Model:	OrderedModel	AIC:	421.2
Method:	Maximum Likelihood	BIC:	444.0
Date:	Tue, 15 Dec 2020		
Time:	23:04:54		
No. Observations:	128		
Df Residuals:	120		
Df Model:	8		

	coef	std err	z	P>|z|	[0.025	0.975]
Date	-0.2424	0.426	-0.570	0.569	-1.077	0.592
Temp	0.0124	0.040	0.310	0.757	-0.066	0.091
Conct	-0.0004	0.003	-0.150	0.881	-0.005	0.005
Count	0.1090	0.049	2.234	0.025	0.013	0.205
0.0/1.0	-1.5467	0.653	-2.369	0.018	-2.827	-0.267
1.0/2.0	-0.1083	0.195	-0.557	0.578	-0.490	0.273
2.0/3.0	-0.3074	0.193	-1.589	0.112	-0.687	0.072
3.0/4.0	0.0616	0.175	0.352	0.725	-0.282	0.405

Again, from the left result table, we can find that only the coefficient of "Count" is relatively significant.

The prediction results of test data are shown below:

```
In [233]: confusion_matrix(y3_test, pred_choice_prob)
Out[233]:
array([[3, 0, 0, 0, 1],
       [1, 0, 0, 1, 7],
       [4, 0, 0, 3, 2],
       [2, 0, 0, 2, 0],
       [2, 0, 0, 2, 2]], dtype=int64)

In [234]: accuracy_score(y3_test, pred_choice_prob)
Out[234]: 0.21875
```

The accuracy here is a little bit better than the multinomial logistic regression