Credit_card_default

January 21, 2020

In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [29]:
```
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
```

In [10]:
```
credit_card_default_df = pd.read_excel('https://archive.ics.uci.edu/ml/machine-learning
```

In [11]:
```
credit_card_default_df.head()
```

Out[11]:

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	\
1	20000	2	2	1	24	2	2	-1	-1	
2	120000	2	2	2	26	-1	2	0	0	
3	90000	2	2	2	34	0	0	0	0	
4	50000	2	2	1	37	0	0	0	0	
5	50000	1	2	1	57	-1	0	-1	0	

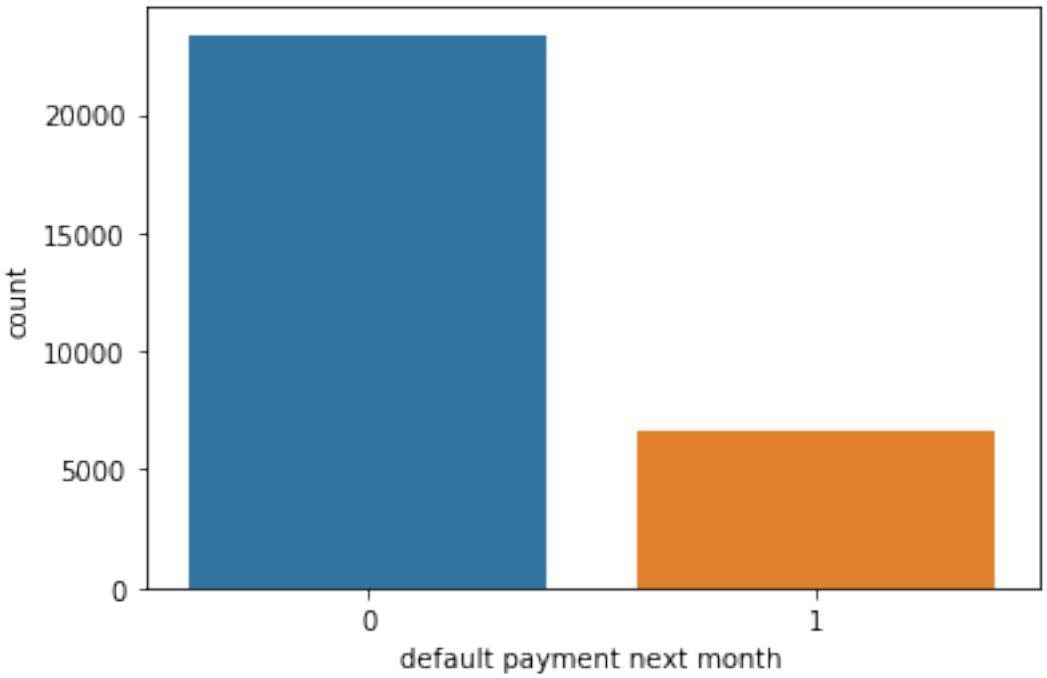
ID	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	\
		...							
1	-2	...	0	0	0	0	689	0	
2	0	...	3272	3455	3261	0	1000	1000	
3	0	...	14331	14948	15549	1518	1500	1000	
4	0	...	28314	28959	29547	2000	2019	1200	
5	0	...	20940	19146	19131	2000	36681	10000	

ID	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
1	0	0	0	1
2	1000	0	2000	1
3	1000	1000	5000	0
4	1100	1069	1000	0
5	9000	689	679	0

[5 rows x 24 columns]

In [34]:
```
sns.countplot(data = credit_card_default_df, x='default payment next month')
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1a2ecaa7f0>




```
In [24]: credit_card_default_df[predict].values.ravel()

Out[24]: array([1, 1, 0, ..., 1, 1, 1])

In [15]: features = ['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE',
                     'PAY_0', 'PAY_2','PAY_3', 'PAY_4', 'PAY_5', 'PAY_6',
                     'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3',
                     'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6',
                     'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
                     'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6']

In [19]: predict = ['default payment next month']

In [26]: from sklearn.feature_selection import SelectFromModel
         X = credit_card_default_df[features]
         y = credit_card_default_df[predict].values.ravel()

In [27]: X.shape

Out[27]: (30000, 23)

In [30]: clf = ExtraTreesClassifier(n_estimators=50)
         clf = clf.fit(X, y)
         clf.feature_importances_
```

```
Out[30]: array([0.06579586, 0.01198064, 0.03371215, 0.0200629 , 0.06713449,
                0.09870915, 0.04143305, 0.03538504, 0.02695245, 0.03767196,
                0.02810789, 0.0507392 , 0.04700449, 0.04531697, 0.04405851,
                0.04333364, 0.04427121, 0.04430054, 0.04229839, 0.0420066 ,
                0.04037105, 0.04315849, 0.04619534])

In [31]: model = SelectFromModel(clf, prefit=True)
         X_new = model.transform(X)
         X_new.shape

Out[31]: (30000, 10)

In [32]: clf = RandomForestClassifier(n_estimators=300, random_state=42)

In [33]: print(cross_val_score(clf, X=X_new, y = credit_card_default_df[predict].values.ravel(),

[0.8064 0.819  0.819 ]


In [25]: clf.fit(X=credit_card_default_df[features], y = credit_card_default_df[predict].values.

Out[25]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                                criterion='gini', max_depth=None, max_features='auto',
                                max_leaf_nodes=None, max_samples=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=300,
                                n_jobs=None, oob_score=False, random_state=42, verbose=0,
                                warm_start=False)

In [ ]:
```