Loan2

March 29, 2019

1 Hitel elemzés

1.1 Feladat

Hitelezéshez elrejelzések

```
In [1]: import pandas as pd
        import numpy as np
        from matplotlib import pyplot as plt
        %matplotlib inline

        df = pd.read_csv("train.csv")
```

```
In [2]: df.head(10)
```

```
Out[2]:     Loan_ID Gender Married Dependents     Education Self_Employed  \
        0  LP001002   Male      No          0      Graduate            No
        1  LP001003   Male     Yes          1      Graduate            No
        2  LP001005   Male     Yes          0      Graduate           Yes
        3  LP001006   Male     Yes          0  Not Graduate            No
        4  LP001008   Male      No          0      Graduate            No
        5  LP001011   Male     Yes          2      Graduate           Yes
        6  LP001013   Male     Yes          0  Not Graduate            No
        7  LP001014   Male     Yes         3+      Graduate            No
        8  LP001018   Male     Yes          2      Graduate            No
        9  LP001020   Male     Yes          1      Graduate            No

           ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
        0             5849                0.0         NaN             360.0
        1             4583             1508.0       128.0             360.0
        2             3000                0.0        66.0             360.0
        3             2583             2358.0       120.0             360.0
        4             6000                0.0       141.0             360.0
        5             5417             4196.0       267.0             360.0
        6             2333             1516.0        95.0             360.0
        7             3036             2504.0       158.0             360.0
        8             4006             1526.0       168.0             360.0
        9            12841            10968.0       349.0             360.0
```

```
   Credit_History Property_Area Loan_Status
0             1.0         Urban           Y
1             1.0         Rural           N
2             1.0         Urban           Y
3             1.0         Urban           Y
4             1.0         Urban           Y
5             1.0         Urban           Y
6             1.0         Urban           Y
7             0.0     Semiurban           N
8             1.0         Urban           Y
9             1.0     Semiurban           N
```

```
In [14]: df.describe()
```

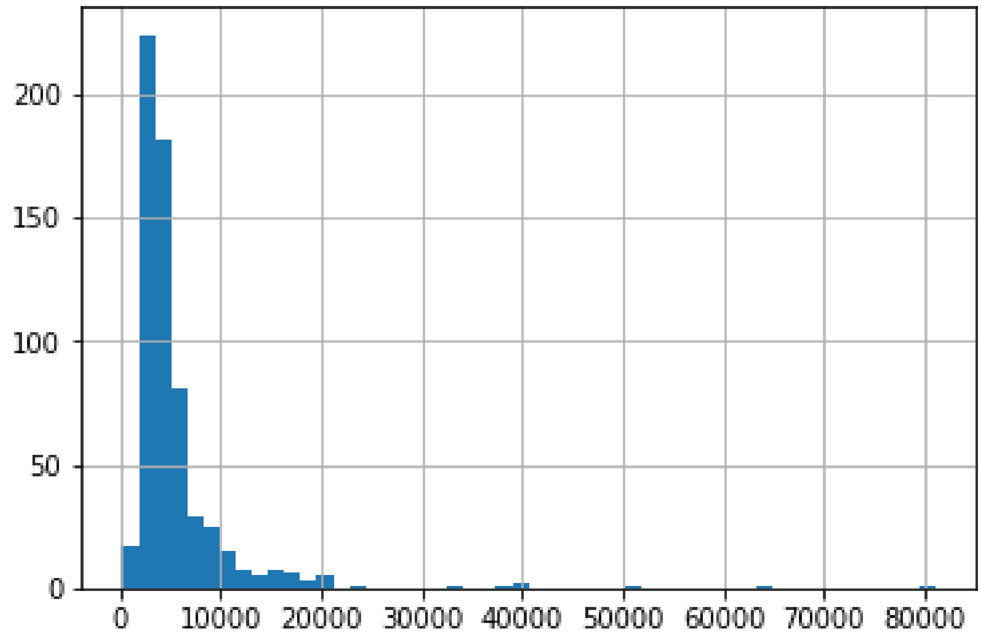
```
Out[14]:        ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
         count       614.000000         614.000000  592.000000         600.00000
         mean       5403.459283        1621.245798  146.412162         342.00000
         std        6109.041673        2926.248369   85.587325          65.12041
         min         150.000000           0.000000    9.000000          12.00000
         25%        2877.500000           0.000000  100.000000         360.00000
         50%        3812.500000        1188.500000  128.000000         360.00000
         75%        5795.000000        2297.250000  168.000000         360.00000
         max       81000.000000       41667.000000  700.000000         480.00000

                Credit_History
         count      564.000000
         mean         0.842199
         std          0.364878
         min          0.000000
         25%          1.000000
         50%          1.000000
         75%          1.000000
         max          1.000000
```

```
In [15]: df['Property_Area'].value_counts()
```

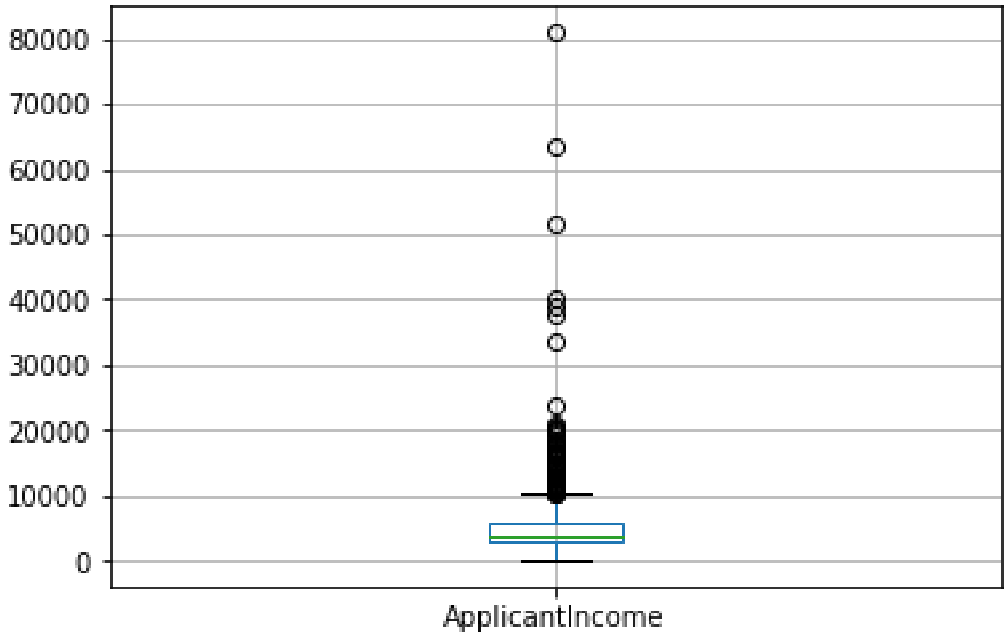
```
Out[15]: Semiurban    233
         Urban        202
         Rural        179
         Name: Property_Area, dtype: int64
```

```
In [16]: df['ApplicantIncome'].hist(bins=50)
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd0dacf2470>
```



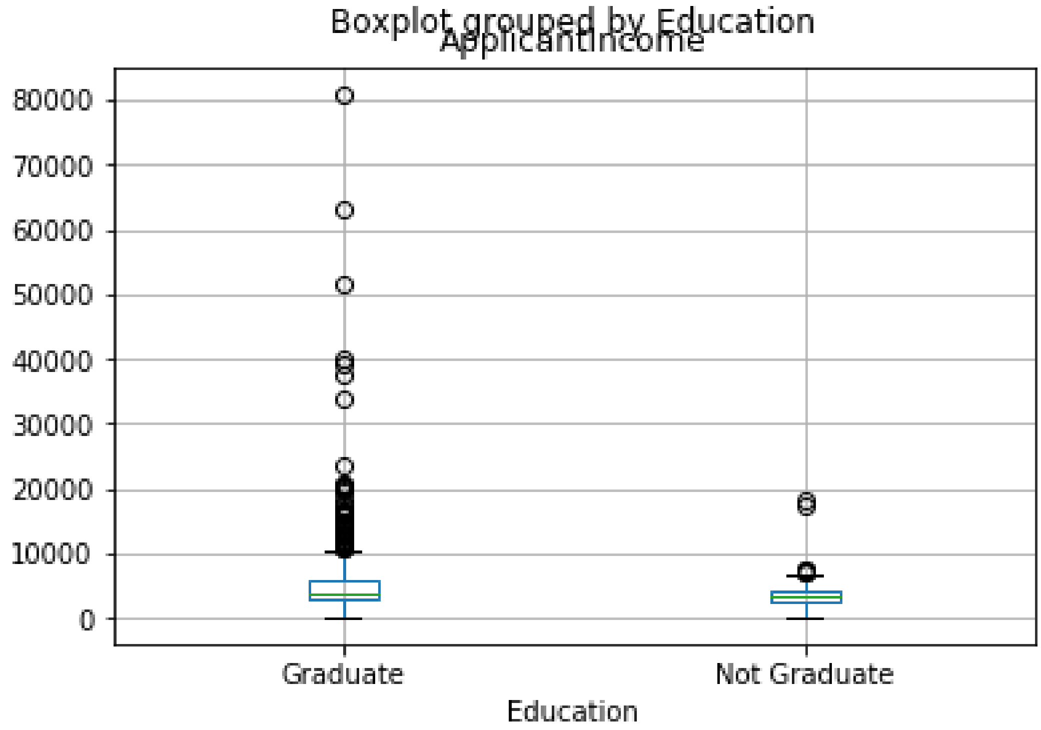
```
In [17]: df.boxplot(column='ApplicantIncome')

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd0da49f0f0>
```



```
In [18]: df.boxplot(column='ApplicantIncome', by = 'Education')
```

```
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/numpy/core/fromnumeric.py:57: FutureWarni
  return getattr(obj, method)(*args, **kwds)
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd0da1c4160>
```



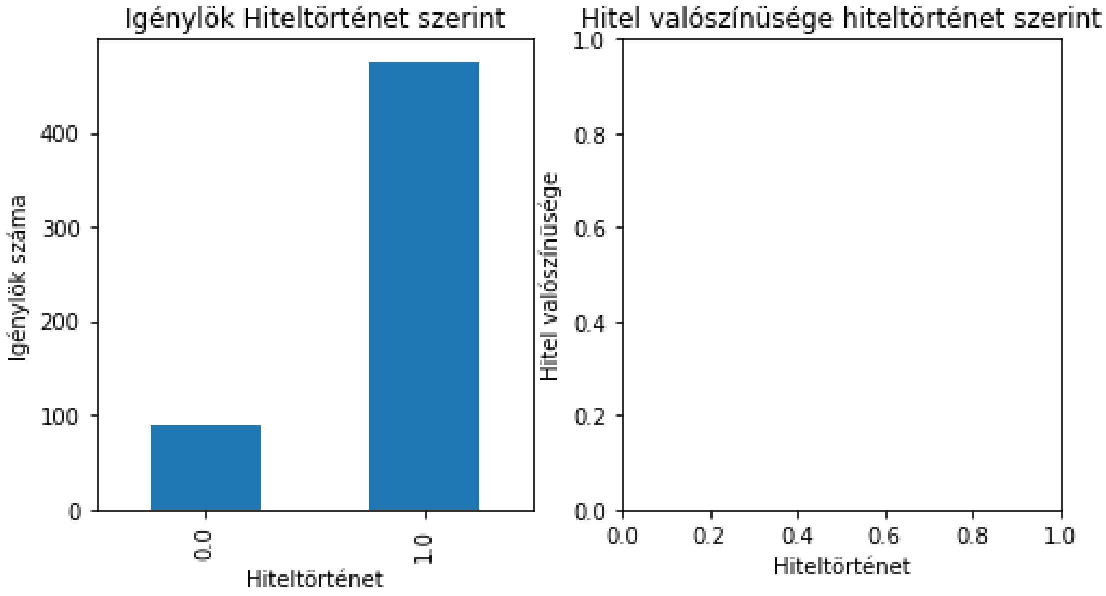
```
In [19]: temp1 = df['Credit_History'].value_counts(ascending=True)
         temp2 = df.pivot_table(values='Loan_Status',index=['Credit_History'],aggfunc=lambda x:
         print ('A hiteltörténet gyakorisági eloszlása:')
         print (temp1)
         print ('\nAz egyes osztályok milyen valószínséggel kapnak hitelt:')
         print (temp2)
```

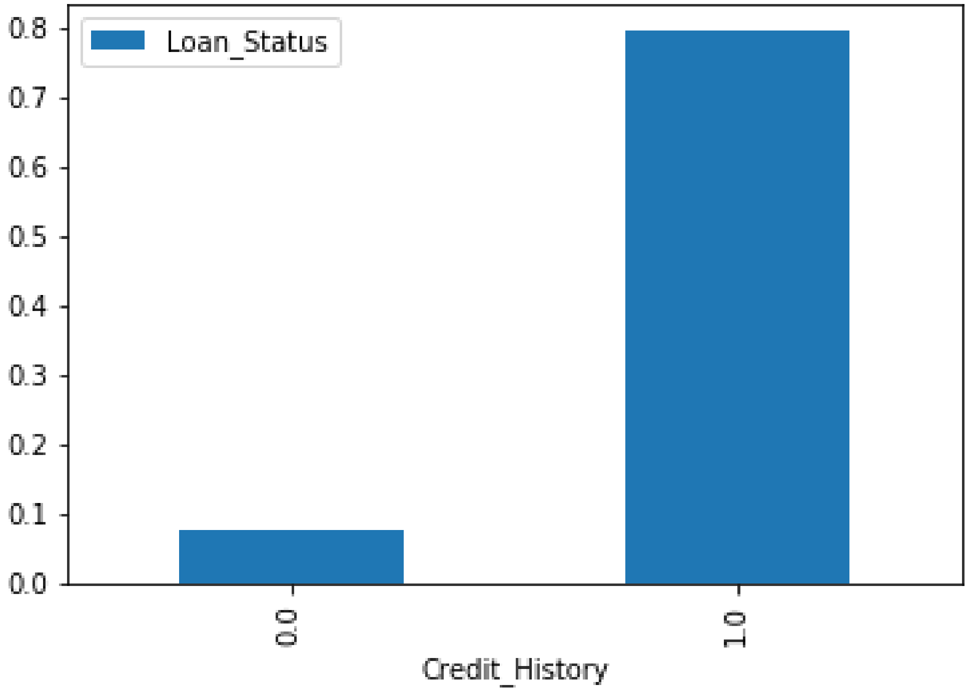
```
A hiteltörténet gyakorisági eloszlása:
0.0     89
1.0    475
Name: Credit_History, dtype: int64

Az egyes osztályok milyen valószínséggel kapnak hitelt:
                Loan_Status
Credit_History
0.0                0.078652
```

```
1.0                0.795789
```

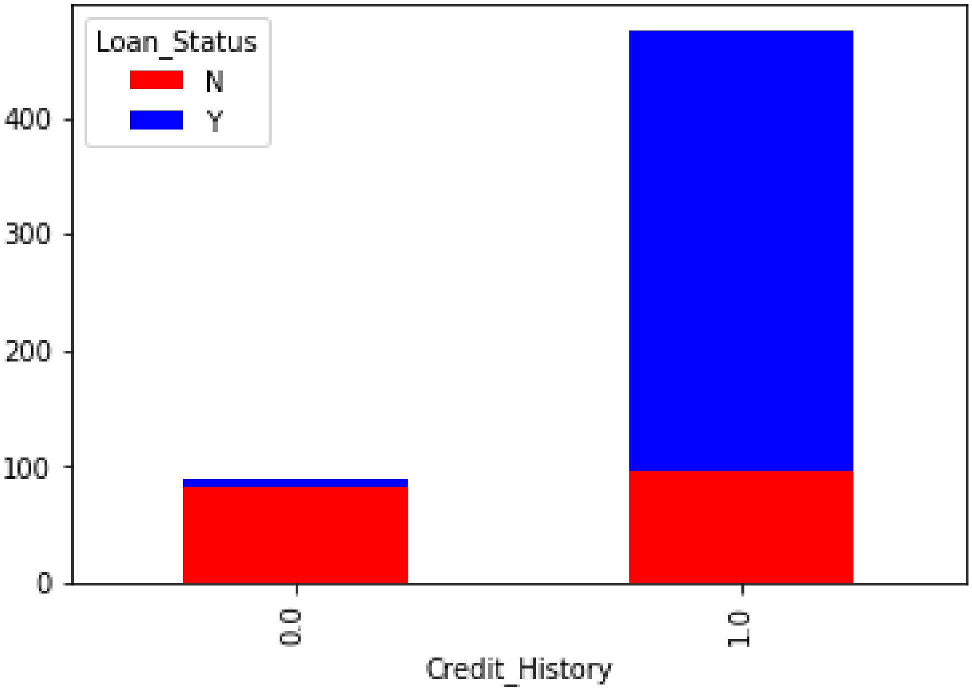
```
In [20]: fig = plt.figure(figsize=(8,4))
         ax1 = fig.add_subplot(121)
         ax1.set_xlabel('Hiteltörténet')
         ax1.set_ylabel('Igénylök száma')
         ax1.set_title("Igénylök Hiteltörténet szerint")
         temp1.plot(kind='bar')
         ax2 = fig.add_subplot(122)
         temp2.plot(kind = 'bar')
         ax2.set_xlabel('Hiteltörténet')
         ax2.set_ylabel('Hitel valószínüsége')
         ax2.set_title("Hitel valószínüsége hiteltörténet szerint")
```

```
Out[20]: Text(0.5,1,'Hitel valószínüsége hiteltörténet szerint')
```





```
In [29]: temp3 = pd.crosstab(df['Credit_History'], df['Loan_Status'])
         temp3.plot(kind='bar', stacked=True, color=['red','blue'], grid=False)

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd0d9d40fd0>
```



```
In [32]: df.apply(lambda x: sum(x.isnull()),axis=0)
```

```
Out[32]: Loan_ID               0
         Gender               13
         Married               3
         Dependents           15
         Education             0
         Self_Employed        32
         ApplicantIncome       0
         CoapplicantIncome     0
         LoanAmount            0
         Loan_Amount_Term     14
         Credit_History       50
         Property_Area         0
         Loan_Status           0
         dtype: int64
```

```
In [31]:  df['LoanAmount'].fillna(df['LoanAmount'].mean(), inplace=True)
```

```
In [33]: df.apply(lambda x: sum(x.isnull()),axis=0)
```

```
Out[33]: Loan_ID               0
         Gender               13
         Married               3
         Dependents           15
         Education             0
         Self_Employed        32
         ApplicantIncome       0
         CoapplicantIncome     0
         LoanAmount            0
         Loan_Amount_Term     14
         Credit_History       50
         Property_Area         0
         Loan_Status           0
         dtype: int64
```

```
In [34]:  df['Self_Employed'].value_counts()
```

```
Out[34]: No     500
         Yes     82
         Name: Self_Employed, dtype: int64
```

```
In [36]: table = df.pivot_table(values='LoanAmount', index='Self_Employed' ,columns='Education',

         def fage(x):
          return table.loc[x['Self_Employed'],x['Education']]
```