final project

December 8, 2018

```
In [61]: # Input data files are available in the "../input/" directory.
         # For example, running this (by clicking run or pressing Shift+Enter) will list the f

         import os
         print(os.listdir("../input"))

         # Any results you write to the current directory are saved as output.
```

```
['train_dataset.csv', 'test_dataset.csv']
```

```
In [62]: import pandas as pd
         import matplotlib.pyplot as plt
         import numpy as np
         import seaborn as sns
         from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
         from sklearn.decomposition import TruncatedSVD
         from sklearn import preprocessing, model_selection, metrics, ensemble
         # import lightgbm as lgb
```

1 Question 1

1.1 1a)

```
In [63]: data = pd.read_csv("../input/train_dataset.csv")
         test = pd.read_csv("../input/test_dataset.csv")
         # whole_data = pd.concat([data.drop('IsBadBuy',axis=1),test])
         data.head()
```

```
Out[63]:    RT13Id  IsBadBuy  PurchDate    Auction  PaintingYear  PaintingAge  \
         0   21003         0  4/21/2010  Christie's          2007            3
         1   57560         0   4/1/2009   Sotheby's          2004            5
         2   29868         0  3/31/2010  Christie's          2008            2
         3   64473         1  6/16/2010   Sotheby's          2003            7
         4   68666         0  11/4/2010   Sotheby's          2007            3

                          Artist        PaintingName Trim       SubType      ...        \
```

```
0            Cai Jin        TUCSON T2 T7   GLS      4D other      ...
1      Grandma Moses  Moses T6 3.9L T6 E   Bas  2D Landscape      ...
2        Frida Kahlo             AURA T6    XE   4D Genre XE      ...
3  Leonardo Da Vinci           MALIBU T6   Bas      4D Genre      ...
4  Leonardo Da Vinci              COBALT    LS   4D Genre LS      ...

  MMRCurrentRetailAveragePrice MMRCurrentRetailCleanPrice  PRIMEUNIT AUCGUART  \
0                      13641.0                    14951.0        NaN      NaN
1                       6122.0                     7474.0        NaN      NaN
2                      13509.0                    15918.0        NaN      NaN
3                       5243.0                     6541.0        NaN      NaN
4                       8228.0                     9300.0        NaN      NaN

   BYRNO VNZIP1 VNST PaintingBCost  IsOnlineSale  WarrantyCost
0   8655  75236   TX        8160.0             0           920
1  22808  71119   LA        6870.0             0           853
2  20928  32824   FL        8680.0             0          1373
3  21053  85226   AZ        4830.0             0          2508
4  22916  80817   CO        4965.0             0           671

[5 rows x 34 columns]
```

In [64]:
```python
test_rate = 0.3
test_size = int(data.shape[0] * test_rate)
# shuffle the data frame rows without producing new index
def split_set(data, test_size):
    data = data.sample(frac=1).reset_index(drop=True)
    return np.split(data, [test_size], axis=0)
test_data, train_data = split_set(data, test_size)
print(train_data.shape, test_data.shape)
train_data.head()
```

```
(35762, 34) (15326, 34)
```

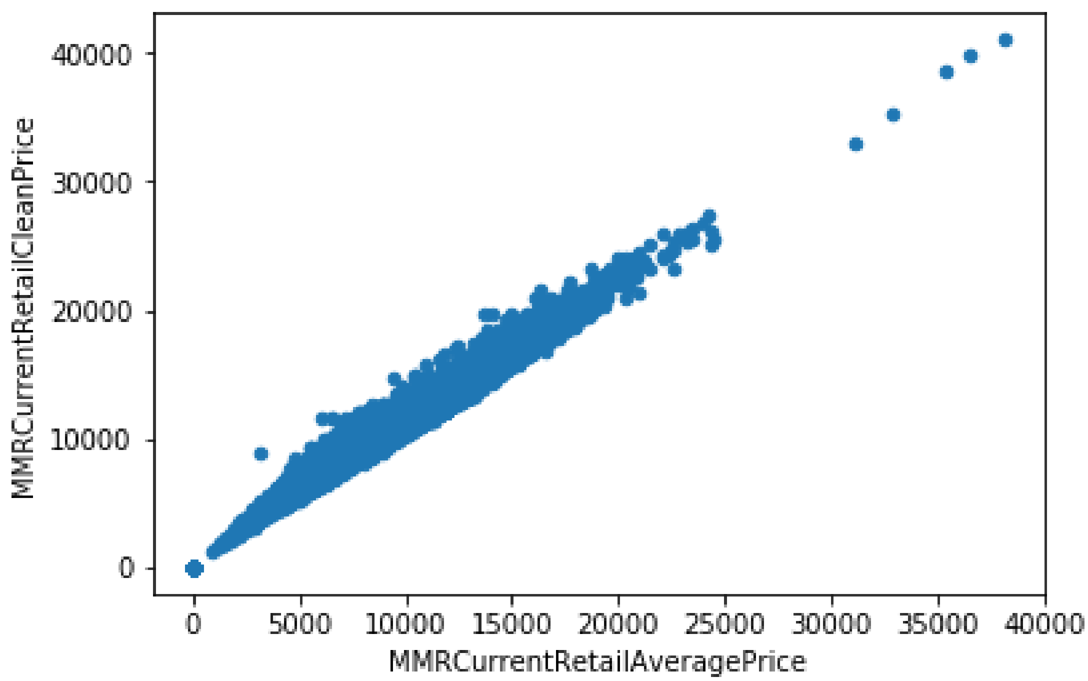
Out[64]:
```
       RT13Id  IsBadBuy  PurchDate     Auction  PaintingYear  PaintingAge  \
15326   18745         0  1/27/2009  Christie's          2005            4
15327   49865         0   8/5/2009       OTHER          2004            5
15328   65969         0  4/23/2009   Sotheby's          2004            5
15329   67174         0   5/6/2009   Sotheby's          2003            6
15330   68961         1  9/10/2009   Sotheby's          2004            5

              Artist         PaintingName Trim       SubType      ...       \
15326        Qu Ding  GRAND PRIX 3.8L T6 S  Bas      4D Genre      ...
15327  Pablo Picasso  STRATUS T6 2.7L T6 M   SE   4D Genre SE      ...
15328  Pablo Picasso     NEON 2.0L I4 T11I   SE      4D Genre      ...
15329  Grandma Moses     Hounds 2.0L I4 SPI   SE   4D Genre SE      ...
15330       Giovanni    LIBERTY T1 T6 3.7L  Spo      4D other      ...
```

```
       MMRCurrentRetailAveragePrice MMRCurrentRetailCleanPrice  PRIMEUNIT  \
15326                        6670.0                     8017.0        NaN
15327                        5817.0                     7704.0        NaN
15328                        3486.0                     4559.0        NaN
15329                        3496.0                     4544.0        NaN
15330                        7757.0                     9668.0        NaN

      AUCGUART  BYRNO  VNZIP1 VNST PaintingBCost  IsOnlineSale  WarrantyCost
15326      NaN  22916   80022   CO        5480.0             0          1974
15327      NaN    835   85009   AZ        4975.0             0          1215
15328      NaN  17675   28273   NC        4005.0             0           588
15329      NaN  20207   77086   TX        3175.0             0          1220
15330      NaN  21973   32219   FL        6900.0             0           983

[5 rows x 34 columns]
```

1.2 1b)

In [65]:
```
plt.figure(figsize=[16,9], dpi=100)
data.plot.scatter("MMRCurrentRetailAveragePrice", "MMRCurrentRetailCleanPrice")
plt.show()
```

```
<matplotlib.figure.Figure at 0x1a0d1b7550>
```



It seems that it is linear.

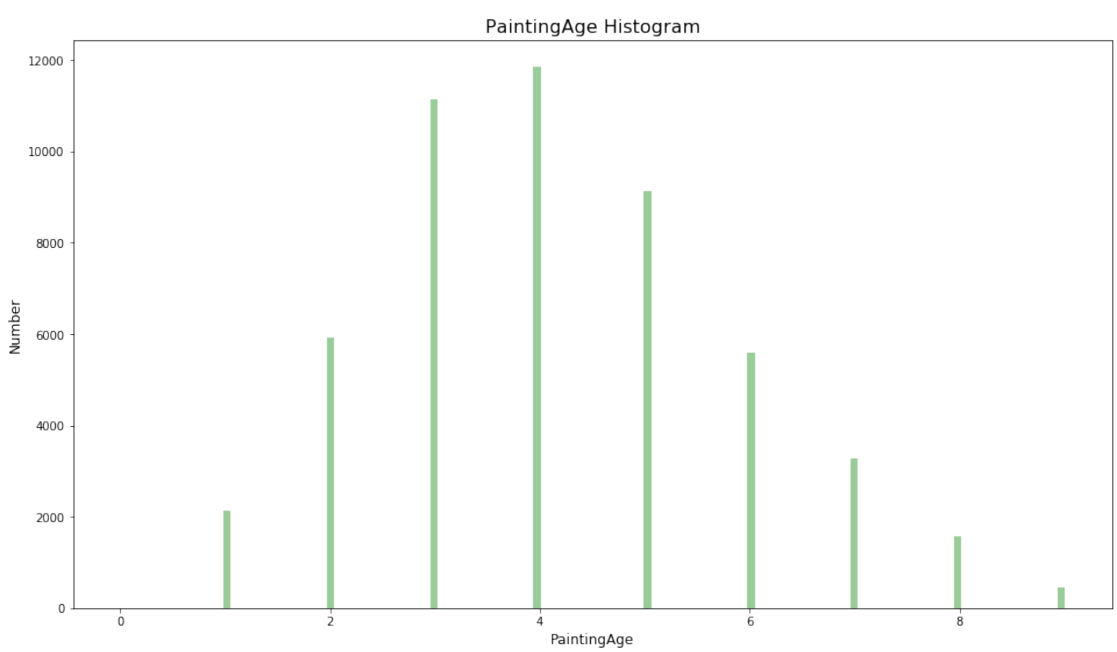
1.3 1c)

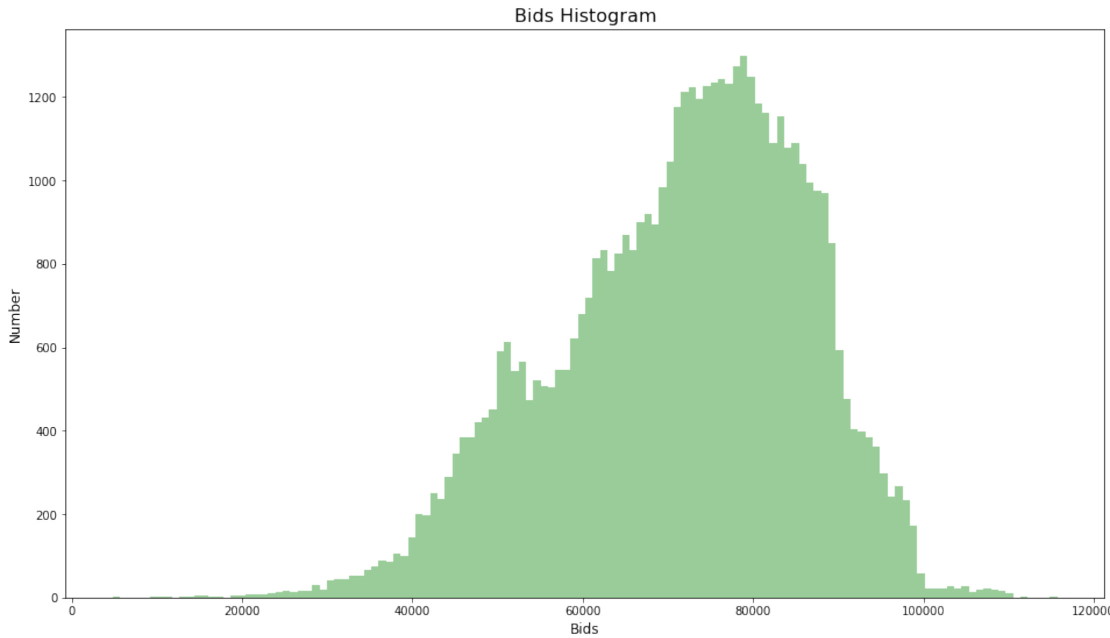
In [66]:
```
data.columns
```

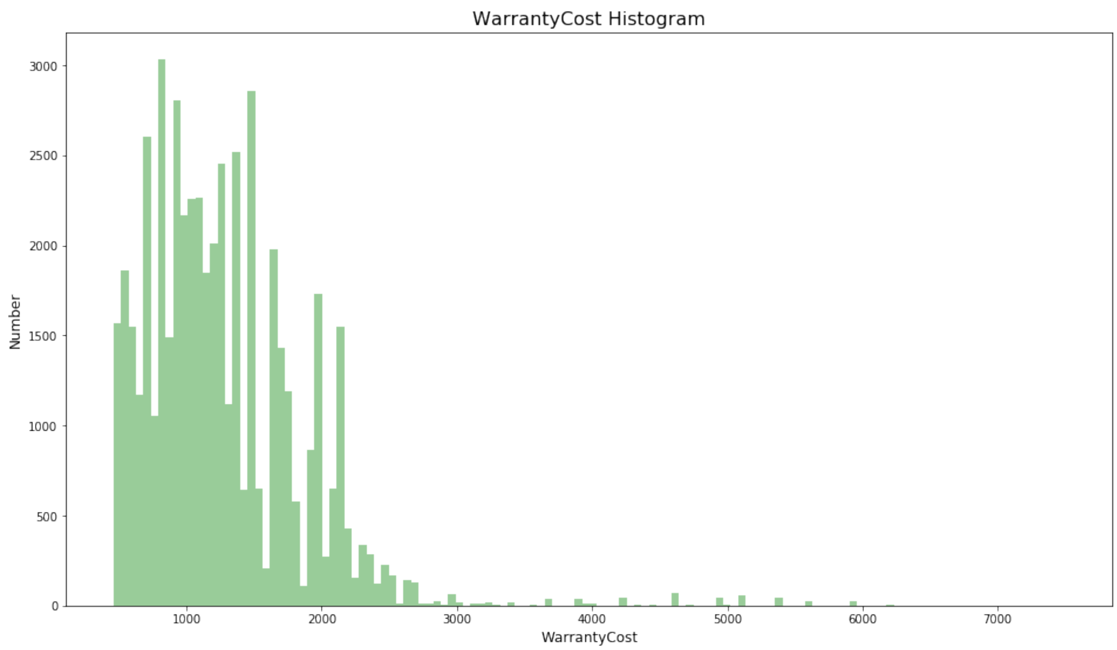
Out[66]:
```
Index(['RT13Id', 'IsBadBuy', 'PurchDate', 'Auction', 'PaintingYear',
       'PaintingAge', 'Artist', 'PaintingName', 'Trim', 'SubType',
       'CanvasColor', 'Market', 'FrameTypeID', 'FrameType', 'Bids',
       'Nationality', 'Size', 'TopThreeNYCName',
       'MMRAcquisitionAuctionAveragePrice', 'MMRAcquisitionAuctionCleanPrice',
       'MMRAcquisitionRetailAveragePrice', 'MMRAcquisitonRetailCleanPrice',
       'MMRCurrentAuctionAveragePrice', 'MMRCurrentAuctionCleanPrice',
       'MMRCurrentRetailAveragePrice', 'MMRCurrentRetailCleanPrice',
       'PRIMEUNIT', 'AUCGUART', 'BYRNO', 'VNZIP1', 'VNST', 'PaintingBCost',
       'IsOnlineSale', 'WarrantyCost'],
      dtype='object')
```

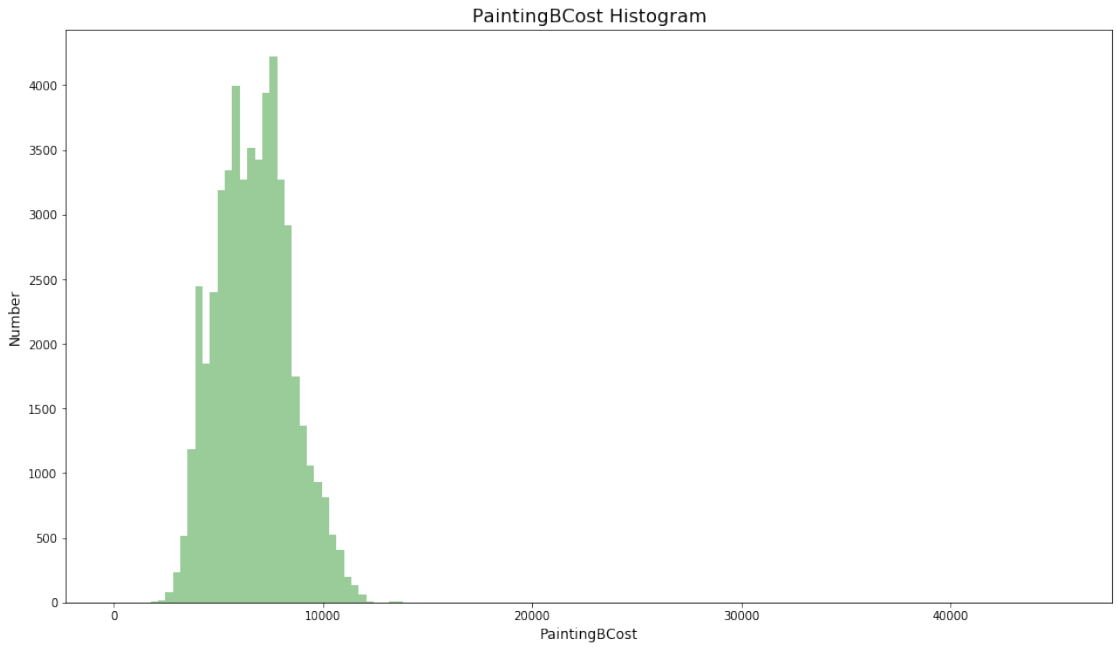
In [67]:
```
need_to_plot = ['PaintingAge', 'Bids', 'WarrantyCost', 'PaintingBCost']
def plot_one_distribution(name):
    plt.figure(figsize=(16, 9))
    sns.distplot(data[name].values, bins=128, kde=False, color='g')
    plt.xlabel(name, fontsize=12)
    plt.ylabel('Number', fontsize=12)
    plt.title("{} Histogram".format(name), fontsize=16)
    plt.show()
```

In [68]:
```
for n in need_to_plot:
    plot_one_distribution(n)
```




Bids Histogram
Number
Bids
0
200
400
600
800
1000
1200
20000
40000
60000
80000
100000
120000


WarrantyCost Histogram
Number
WarrantyCost
0
500
1000
1500
2000
2500
3000
4000
5000
6000
7000



Basically, all distributions are skewed. while 'painting age is slightly skewed, but bids and warranty cost are extremely skewed.

2 Question 2

2.1 2a)

In [69]:
```
data = data.drop('RT13Id', axis=1)
data.head()
```

Out[69]:
```
   IsBadBuy  PurchDate    Auction  PaintingYear  PaintingAge  \
0         0  4/21/2010  Christie's          2007            3
1         0   4/1/2009   Sotheby's          2004            5
2         0  3/31/2010  Christie's          2008            2
3         1  6/16/2010   Sotheby's          2003            7
4         0  11/4/2010   Sotheby's          2007            3

              Artist        PaintingName Trim       SubType CanvasColor  \
0            Cai Jin        TUCSON T2 T7  GLS      4D other      SILVER
1      Grandma Moses  Moses T6 3.9L T6 E  Bas  2D Landscape      SILVER
2        Frida Kahlo             AURA T6   XE   4D Genre XE      SILVER
3  Leonardo Da Vinci           MALIBU T6  Bas      4D Genre        GOLD
4  Leonardo Da Vinci              COBALT   LS   4D Genre LS         RED

        ...       MMRCurrentRetailAveragePrice  MMRCurrentRetailCleanPrice  \
0       ...                            13641.0                     14951.0
1       ...                             6122.0                      7474.0
```

```
2      ...                               13509.0                       15918.0
3      ...                                5243.0                        6541.0
4      ...                                8228.0                        9300.0

  PRIMEUNIT  AUCGUART  BYRNO VNZIP1 VNST  PaintingBCost  IsOnlineSale  \
0       NaN       NaN   8655  75236   TX         8160.0             0
1       NaN       NaN  22808  71119   LA         6870.0             0
2       NaN       NaN  20928  32824   FL         8680.0             0
3       NaN       NaN  21053  85226   AZ         4830.0             0
4       NaN       NaN  22916  80817   CO         4965.0             0

   WarrantyCost
0           920
1           853
2          1373
3          2508
4           671

[5 rows x 33 columns]
```

2.2 2b)

In [70]:
```python
import re
categories = data[['PaintingName', 'SubType']]
```

In [71]:
```python
def extract_one_cate(name, subtype, values, idx):
    pattern = re.compile(r"T\d+")
    r1 = pattern.finditer(name)
    for rr in r1:
        values[idx][int(rr.group().replace("T", "")) - 1] = 1
    if not pd.isnull(subtype):
        r2 = pattern.finditer(subtype)
        for rr in r2:
            values[idx][int(rr.group().replace("T", "")) - 1] = 1
    if sum(values[idx]) == 0:
        values[idx][-1] = 1
    return
```

In [72]:
```python
one_hot_values = np.zeros([data.shape[0], 14])
for idx, row in categories.iterrows():
    extract_one_cate(row[0], row[1], one_hot_values, idx)
one_hot_values[:5]
```

Out[72]:
```
array([[0., 1., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]])
```

```
In [73]: # the last column means the category T is not given
         for i in range(14):
             data["T{}".format(i + 1)] = one_hot_values[:, i]
         data.head()
```

```
Out[73]:    IsBadBuy  PurchDate     Auction  PaintingYear  PaintingAge  \
         0         0  4/21/2010  Christie's          2007            3
         1         0   4/1/2009   Sotheby's          2004            5
         2         0  3/31/2010  Christie's          2008            2
         3         1  6/16/2010   Sotheby's          2003            7
         4         0  11/4/2010   Sotheby's          2007            3

                        Artist        PaintingName Trim       SubType CanvasColor ...   \
         0             Cai Jin        TUCSON T2 T7  GLS      4D other      SILVER ...
         1       Grandma Moses  Moses T6 3.9L T6 E  Bas  2D Landscape      SILVER ...
         2         Frida Kahlo             AURA T6   XE   4D Genre XE      SILVER ...
         3   Leonardo Da Vinci           MALIBU T6  Bas      4D Genre        GOLD ...
         4   Leonardo Da Vinci              COBALT   LS   4D Genre LS         RED ...

             T5   T6   T7   T8   T9  T10  T11  T12  T13  T14
         0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
         1  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
         2  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
         3  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
         4  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0

         [5 rows x 47 columns]
```

2.3 2c)

```
In [74]: def extract_one_num_check(name, values, idx):
             pattern = re.compile(r"I.?\d")
             outpattern = re.compile(r"\d+")
             r = pattern.search(name)
             if r:
                 rr = outpattern.search(r.group()).group()
                 values[idx] = int(rr)
                 return
```

```
In [75]: # 0 here means not given
         num_check = np.zeros(data.shape[0])
         names = data[['PaintingName']]
         for idx, row in names.iterrows():
             extract_one_num_check(row[0], num_check, idx)
         num_check[:10]
```

```
Out[75]: array([0., 0., 0., 0., 0., 0., 0., 0., 4., 0.])
```

```
In [76]: data["num_checked"] = num_check
```

2.4 2d)

```
In [77]: def extract_one_size(name, values, idx):
             pattern = re.compile(r"\d.?\d?L")
             outpattern = re.compile(r"\d.?\d?")
             r = pattern.search(name)
             if r:
                 rr = outpattern.search(r.group()).group()
                 values[idx] = float(rr)
                 return
```

```
In [78]: # zeros here means not given
         sizes = np.zeros(data.shape[0])
         names = data[['PaintingName']]
         for idx, row in names.iterrows():
             extract_one_size(row[0], sizes, idx)
         sizes[:10]
```

```
Out[78]: array([0. , 3.9, 0. , 0. , 0. , 0. , 0. , 0. , 2. , 0. ])
```

```
In [79]: data["given_size"] = sizes
```

2.5 2e)

```
In [80]: _cates = ['Not Given', 'Genre', 'History', 'Still Life', 'Real Life', 'Landscape', 'Po
         cates_to_id = {v: k for k, v in enumerate(_cates)}
         id_to_cates = {k: v for k, v in enumerate(_cates)}
```

```
In [81]: def extract_one_cate_2(subt, _cates, dic, values, idx):
             for c in _cates:
                 if  not pd.isnull(subt) and subt.find(c) >= 0:
                     values[idx] = dic[c]
                     return
             return
```

```
In [82]: types = data[['SubType']]
         cates_2 = np.zeros(data.shape[0], np.int8)
         for idx, row in types.iterrows():
             extract_one_cate_2(row[0], _cates, cates_to_id, cates_2, idx)
         print(cates_2[:10])
         for t in cates_2[:10]:
             print(id_to_cates[t])
```

```
[0 5 1 1 1 1 0 1 1 6]
Not Given
Landscape
Genre
Genre
Genre
```

```
Genre
Not Given
Genre
Genre
Portrait
```

```
In [83]: df_tmp = pd.DataFrame({'cate_': [id_to_cates[t] for t in cates_2]})
         df_tmp = pd.get_dummies(df_tmp, prefix=['cate_'], drop_first=True)
         data = pd.concat([data, df_tmp], axis=1)
         data.head()
```

```
Out[83]:    IsBadBuy  PurchDate     Auction  PaintingYear  PaintingAge  \
         0         0  4/21/2010  Christie's          2007            3
         1         0   4/1/2009   Sotheby's          2004            5
         2         0  3/31/2010  Christie's          2008            2
         3         1  6/16/2010   Sotheby's          2003            7
         4         0  11/4/2010   Sotheby's          2007            3

                       Artist        PaintingName Trim       SubType CanvasColor  \
         0            Cai Jin        TUCSON T2 T7  GLS      4D other      SILVER
         1      Grandma Moses  Moses T6 3.9L T6 E  Bas  2D Landscape      SILVER
         2        Frida Kahlo             AURA T6   XE   4D Genre XE      SILVER
         3  Leonardo Da Vinci          MALIBU T6  Bas      4D Genre        GOLD
         4  Leonardo Da Vinci             COBALT   LS   4D Genre LS         RED

                  ...          T14  num_checked given_size  cate__Genre cate__History  \
         0        ...          0.0          0.0        0.0            0             0
         1        ...          0.0          0.0        3.9            0             0
         2        ...          0.0          0.0        0.0            1             0
         3        ...          0.0          0.0        0.0            1             0
         4        ...          1.0          0.0        0.0            1             0

           cate__Landscape cate__Not Given  cate__Portrait  cate__Real Life  \
         0               0               1               0                0
         1               1               0               0                0
         2               0               0               0                0
         3               0               0               0                0
         4               0               0               0                0

            cate__Still Life
         0                 0
         1                 0
         2                 0
         3                 0
         4                 0

         [5 rows x 56 columns]
```

2.6 2f)

```
In [84]: prices = ['MMRAcquisitionAuctionAveragePrice', 'MMRAcquisitionAuctionCleanPrice',
                   'MMRAcquisitionRetailAveragePrice', 'MMRAcquisitonRetailCleanPrice',
                   'MMRCurrentAuctionAveragePrice', 'MMRCurrentAuctionCleanPrice',
                   'MMRCurrentRetailAveragePrice', 'MMRCurrentRetailCleanPrice',]
         for p in prices:
             data = data.loc[data[p] != 0]
             data["{}_ratio".format(p)] = data["Bids"] / data[p]

         data.head()
```

```
Out[84]:    IsBadBuy  PurchDate     Auction  PaintingYear  PaintingAge  \
         0         0  4/21/2010  Christie's          2007            3
         1         0   4/1/2009   Sotheby's          2004            5
         2         0  3/31/2010  Christie's          2008            2
         3         1  6/16/2010   Sotheby's          2003            7
         4         0  11/4/2010   Sotheby's          2007            3

                       Artist        PaintingName Trim       SubType CanvasColor  \
         0            Cai Jin        TUCSON T2 T7  GLS      4D other      SILVER
         1      Grandma Moses  Moses T6 3.9L T6 E  Bas  2D Landscape      SILVER
         2        Frida Kahlo             AURA T6   XE   4D Genre XE      SILVER
         3  Leonardo Da Vinci           MALIBU T6  Bas      4D Genre        GOLD
         4  Leonardo Da Vinci              COBALT   LS   4D Genre LS         RED

                         ...                cate__Real Life  cate__Still Life  \
         0               ...                              0                 0
         1               ...                              0                 0
         2               ...                              0                 0
         3               ...                              0                 0
         4               ...                              0                 0

           MMRAcquisitionAuctionAveragePrice_ratio  \
         0                                8.575855
         1                               15.002756
         2                                7.956712
         3                               33.343656
         4                               10.798450

            MMRAcquisitionAuctionCleanPrice_ratio  \
         0                               7.490121
         1                              12.026510
         2                               7.132863
         3                              23.385822
         4                               9.091841

           MMRAcquisitionRetailAveragePrice_ratio MMRAcquisitonRetailCleanPrice_ratio  \
         0                               6.209689                            5.602900
```

```
1                            12.803387                            10.425124
2                             6.008796                             5.527849
3                            14.476937                            12.280057
4                             6.870934                             6.042136

  MMRCurrentAuctionAveragePrice_ratio  MMRCurrentAuctionCleanPrice_ratio  \
0                            8.104874                           7.167936
1                           15.685747                          12.646740
2                            8.233203                           7.191053
3                           33.487689                          23.231645
4                           12.008621                          10.038778

   MMRCurrentRetailAveragePrice_ratio  MMRCurrentRetailCleanPrice_ratio
0                            5.863720                          5.349943
1                           13.338778                         10.925876
2                            6.068399                          5.150019
3                           14.786191                         11.852010
4                            7.110598                          6.290968

[5 rows x 64 columns]
```

2.7 2g)

In [85]:
```python
factors = ["Artist", "CanvasColor","Market","Nationality","Auction"]

for f in factors:
    df_tmp = data[[f]]
    df_tmp = pd.get_dummies(df_tmp, prefix=[f], drop_first=True)
    data = pd.concat([data, df_tmp], axis=1)
data.head(20)
```

Out[85]:
```
    IsBadBuy   PurchDate     Auction  PaintingYear  PaintingAge  \
0          0   4/21/2010  Christie's          2007            3
1          0    4/1/2009   Sotheby's          2004            5
2          0   3/31/2010  Christie's          2008            2
3          1   6/16/2010   Sotheby's          2003            7
4          0   11/4/2010   Sotheby's          2007            3
5          0   1/13/2010   Sotheby's          2006            4
6          1   2/19/2009  Christie's          2003            6
7          0   1/21/2010  Christie's          2006            4
8          1    9/3/2009  Christie's          2005            4
9          0   1/27/2009  Christie's          2005            4
10         0   10/7/2009   Sotheby's          2007            2
11         0  11/20/2009       OTHER          2005            4
12         0  11/11/2010       OTHER          2006            4
13         0   9/16/2010  Christie's          2005            5
14         0   2/25/2010  Christie's          2003            7
15         0   4/23/2009  Christie's          2006            3
```

16	0	5/25/2010	OTHER	2006	4
17	0	5/19/2009	OTHER	2002	7
18	0	1/14/2009	OTHER	2005	4
19	0	6/3/2009	Sotheby's	2006	3

	Artist	PaintingName	Trim	SubType \
0	Cai Jin	TUCSON T2 T7	GLS	4D other
1	Grandma Moses	Moses T6 3.9L T6 E	Bas	2D Landscape
2	Frida Kahlo	AURA T6	XE	4D Genre XE
3	Leonardo Da Vinci	MALIBU T6	Bas	4D Genre
4	Leonardo Da Vinci	COBALT	LS	4D Genre LS
5	Michelangelo	David	Bas	4D Genre
6	Leonardo Da Vinci	TRAILBLAZER T2 T8 4	LS	4D other 4.2L LS
7	Qu Ding	G6 T6	Bas	4D Genre
8	Pablo Picasso	NEON 2.0L I4 T11I	SXT	4D Genre
9	Pablo Picasso	CARAEnlarged GRAND T4 T6	SE	Portrait 3.3L
10	Leonardo Da Vinci	Monalisa T6 3.5L T6 T11	LT	4D Genre LT 3.5L
11	Frida Kahlo	VUE T2 T6	NaN	4D other 3.0L
12	Grandma Moses	Moses T6	Bas	2D Landscape
13	Grandma Moses	EXPLORER T2 T6	XLT	4D other 4.0L FFV XLT
14	Grandma Moses	Hounds	SE	4D Genre SE
15	Pablo Picasso	STRATUS T7 2.4L I4 S	SXT	4D Genre SXT
16	Leonardo Da Vinci	AVEO	LS	4D Genre LS
17	Buick	LE SABRE Unspecified	Cus	4D Genre CUSTOM
18	Andy Warhol	TOWN & COUNTRY T4 V	LX	Portrait 3.3L LX
19	Andy Warhol	Pacific T4 3.5L T6	Bas	4D SPORT

	CanvasColor	...	CanvasColor_RED	CanvasColor_SILVER \
0	SILVER	...	0	1
1	SILVER	...	0	1
2	SILVER	...	0	1
3	GOLD	...	0	0
4	RED	...	1	0
5	GREY	...	0	0
6	GREEN	...	0	0
7	RED	...	1	0
8	RED	...	1	0
9	GOLD	...	0	0
10	BLACK	...	0	0
11	BLUE	...	0	0
12	GREY	...	0	0
13	BLUE	...	0	0
14	SILVER	...	0	1
15	BLUE	...	0	0
16	RED	...	1	0
17	GREY	...	0	0
18	BLUE	...	0	0
19	SILVER	...	0	1

	CanvasColor_WHITE	CanvasColor_YELLOW	Market_Non Commercial \
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	1
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0

	Nationality_OTHER	Nationality_OTHER ASIAN	Nationality_TOP LINE ASIAN \
0	0	1	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	1
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0

	Auction_OTHER	Auction_Sotheby's
0	0	0
1	0	1

```
2              0                  0
3              0                  1
4              0                  1
5              0                  1
6              0                  0
7              0                  0
8              0                  0
9              0                  0
10             0                  1
11             1                  0
12             1                  0
13             0                  0
14             0                  0
15             0                  0
16             1                  0
17             1                  0
18             1                  0
19             0                  1

[20 rows x 116 columns]
```

In [86]:
```
f = "PaintingYear"
unique_years = list(set(data[f].tolist()))
dic = {v: k for k, v in enumerate(unique_years)}
years = [dic[y] for y in data[f].tolist()]
yys = np.zeros([data.shape[0], len(dic)])
for i, y in enumerate(years):
    yys[i][y] = 1

for i, k in enumerate(dic):
    data[str(k)] = yys[:, i]
data.head()
```

Out[86]:
```
   IsBadBuy  PurchDate     Auction  PaintingYear  PaintingAge  \
0         0  4/21/2010  Christie's          2007            3
1         0   4/1/2009   Sotheby's          2004            5
2         0  3/31/2010  Christie's          2008            2
3         1  6/16/2010   Sotheby's          2003            7
4         0  11/4/2010   Sotheby's          2007            3

              Artist        PaintingName Trim       SubType CanvasColor  ...   \
0            Cai Jin        TUCSON T2 T7  GLS      4D other      SILVER  ...
1      Grandma Moses  Moses T6 3.9L T6 E  Bas  2D Landscape      SILVER  ...
2        Frida Kahlo             AURA T6   XE   4D Genre XE      SILVER  ...
3  Leonardo Da Vinci           MALIBU T6  Bas      4D Genre        GOLD  ...
4  Leonardo Da Vinci              COBALT   LS   4D Genre LS         RED  ...

  2001  2002 2003  2004 2005 2006 2007  2008  2009  2010
```

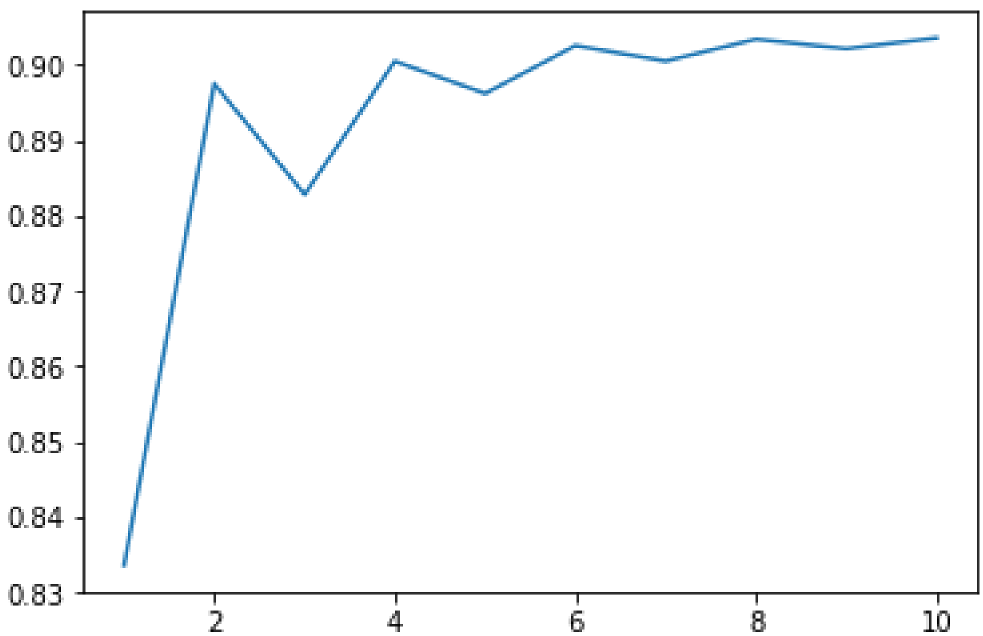
```
0  0.0   0.0  0.0   0.0  0.0  0.0  1.0   0.0   0.0   0.0
1  0.0   0.0  0.0   1.0  0.0  0.0  0.0   0.0   0.0   0.0
2  0.0   0.0  0.0   0.0  0.0  0.0  0.0   1.0   0.0   0.0
3  0.0   0.0  1.0   0.0  0.0  0.0  0.0   0.0   0.0   0.0
4  0.0   0.0  0.0   0.0  0.0  0.0  1.0   0.0   0.0   0.0

[5 rows x 126 columns]
```

3 Question 3

```
In [87]: # More feature engineering here

In [88]: #separating day, month, year from purchase date

In [89]: purdate = ['purchase_Month','purchase_Day','purchase_Year']

In [90]: for idx in range(len(purdate)):
             data[purdate[idx]] = data['PurchDate'].map(lambda x: int(x.split('/')[idx]))

In [91]: # now we have prepared the training dataset already, we will filter out the predictor

In [92]: features_not_used = ['PurchDate',"Auction",'PaintingYear','Artist','PaintingName','Tr
                              'CanvasColor','Market','FrameType','TopThreeNYCName','Size',"PRI
                             "AUCGUART","VNST","Nationality"]
         data_train = data.drop(features_not_used,axis=1)
         data_train = data_train.dropna(axis=0)
         data_train = data_train

In [93]: from sklearn.model_selection import cross_val_score
         X = data_train.drop('IsBadBuy',axis=1)
         y = data_train['IsBadBuy']

In [94]: # model performance
         model_name = []
         model_accuracy = []

In [95]: ## logistic regression

In [96]: from sklearn.linear_model import LogisticRegression

         glm_fit = LogisticRegression()  #
         scores_glm = cross_val_score(glm_fit, X, y,cv=5).mean()

In [97]: model_name.append('logistic')
         model_accuracy.append(scores_glm)
         scores_glm

Out[97]: 0.9042922827965787

In [98]: ##Knn
```

In [99]:
```
from sklearn.neighbors import KNeighborsClassifier
# model_knn = KNeighborsClassifier(n_neighbors = 1)
# cross_val_score(model_knn,X,y,cv=5).mean()
score = []
for k in range(10):
    model_knn = KNeighborsClassifier(n_neighbors = k+1)
    score.append(cross_val_score(model_knn,X,y,cv=5).mean())
```

In [100]:
```
plt.plot(np.array(range(10))+1 ,score)
plt.show()
```



In [101]:
```
knn_acc = max(score)
model_name.append('knn')
model_accuracy.append(knn_acc)
knn_acc
```

Out[101]: 0.9036265003708437

In [102]:
```
#####random forest
```

In [103]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.tree import DecisionTreeClassifier
```

In [104]:
```
clf_dt = DecisionTreeClassifier(max_depth=None,
                                min_samples_split=2,random_state=0)
```

```
scores_dt = cross_val_score(clf_dt, X, y,cv=5).mean()
print(scores_dt)


clf_rf = RandomForestClassifier(n_estimators=10, max_depth=None,
                                min_samples_split=2, random_state=0)
scores_rf = cross_val_score(clf_rf, X, y,cv=5).mean()
print(scores_rf)

clf_et = ExtraTreesClassifier(n_estimators=10, max_depth=None,
                              min_samples_split=2, random_state=0)
scores_et = cross_val_score(clf_et, X, y,cv=5).mean()
print(scores_et)
```

```
0.8189666909220369
0.9027318222276595
0.9000270562313769
```

In [105]:
```
rf_acc = max(scores_dt,scores_rf,scores_et)
model_name.append('random forest')
model_accuracy.append(rf_acc)
rf_acc
```

Out[105]: 0.9027318222276595

In [106]:
```
feat = X.columns
```

In [107]:
```
clf_rf = RandomForestClassifier(n_estimators=10, max_depth=None,
                                min_samples_split=2, random_state=0)
clf_rf_fit = clf_rf.fit(X,y)
importances = clf_rf_fit.feature_importances_
indices = np.argsort(importances)[::-1]
for f in range(X.shape[1]):
    print("%2d) %-*s %f" %(f+1,30,feat[f],importances[indices[f]]))
```

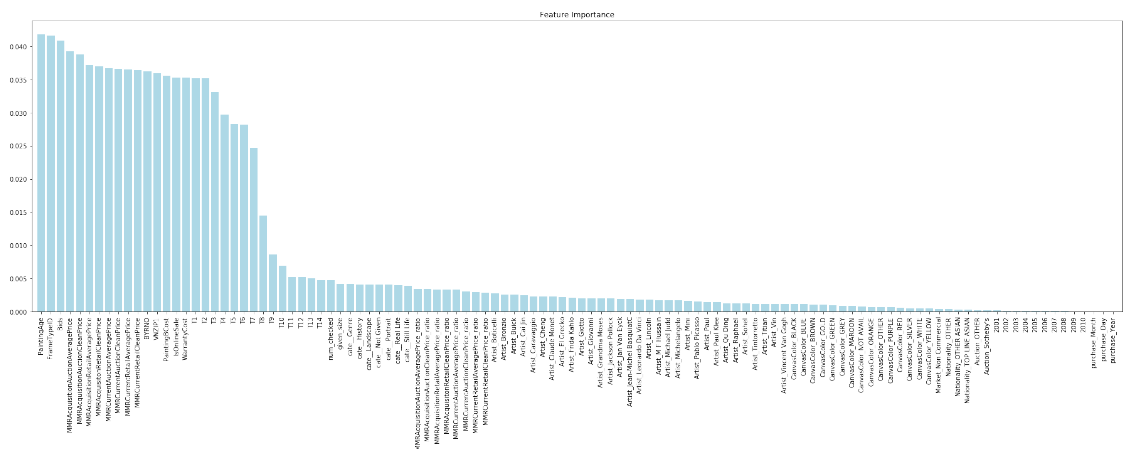
```
 1) PaintingAge                    0.041733
 2) FrameTypeID                    0.041544
 3) Bids                           0.040878
 4) MMRAcquisitionAuctionAveragePrice 0.039208
 5) MMRAcquisitionAuctionCleanPrice 0.038716
 6) MMRAcquisitionRetailAveragePrice 0.037156
 7) MMRAcquisitonRetailCleanPrice  0.037007
 8) MMRCurrentAuctionAveragePrice  0.036681
 9) MMRCurrentAuctionCleanPrice    0.036547
10) MMRCurrentRetailAveragePrice   0.036486
11) MMRCurrentRetailCleanPrice     0.036365
12) BYRNO                          0.036227
13) VNZIP1                         0.035952
```

```
14) PaintingBCost                  0.035571
15) IsOnlineSale                   0.035288
16) WarrantyCost                   0.035246
17) T1                             0.035169
18) T2                             0.035139
19) T3                             0.033085
20) T4                             0.029642
21) T5                             0.028236
22) T6                             0.028222
23) T7                             0.024669
24) T8                             0.014471
25) T9                             0.008623
26) T10                            0.006954
27) T11                            0.005215
28) T12                            0.005210
29) T13                            0.005044
30) T14                            0.004744
31) num_checked                    0.004734
32) given_size                     0.004217
33) cate__Genre                    0.004172
34) cate__History                  0.004122
35) cate__Landscape                0.004119
36) cate__Not Given                0.004108
37) cate__Portrait                 0.004092
38) cate__Real Life                0.004011
39) cate__Still Life               0.003887
40) MMRAcquisitionAuctionAveragePrice_ratio 0.003404
41) MMRAcquisitionAuctionCleanPrice_ratio 0.003387
42) MMRAcquisitionRetailAveragePrice_ratio 0.003376
43) MMRAcquisitonRetailCleanPrice_ratio 0.003337
44) MMRCurrentAuctionAveragePrice_ratio 0.003330
45) MMRCurrentAuctionCleanPrice_ratio 0.003021
46) MMRCurrentRetailAveragePrice_ratio 0.002978
47) MMRCurrentRetailCleanPrice_ratio 0.002856
48) Artist_Boticelli               0.002744
49) Artist_Bronzio                 0.002625
50) Artist_Buick                   0.002542
51) Artist_Cai Jin                 0.002438
52) Artist_Caravaggio              0.002338
53) Artist_Cheng                   0.002316
54) Artist_Claude Monet            0.002250
55) Artist_El Grecko               0.002190
56) Artist_Frida Kahlo             0.002096
57) Artist_Giotto                  0.002035
58) Artist_Giovanni                0.002002
59) Artist_Grandma Moses           0.001974
60) Artist_Jackson Pollock         0.001970
61) Artist_Jan Van Eyck            0.001923
```

```
62) Artist_Jean-Michel BasquiatC   0.001906
63) Artist_Leonardo Da Vinci       0.001849
64) Artist_Lincoln                 0.001809
65) Artist_M F Hussain             0.001771
66) Artist_Michael Judd            0.001733
67) Artist_Michelangelo            0.001694
68) Artist_Mini                    0.001617
69) Artist_Pablo Picasso           0.001553
70) Artist_Paul                    0.001484
71) Artist_Paul Klee               0.001453
72) Artist_Qu Ding                 0.001290
73) Artist_Raphael                 0.001234
74) Artist_Sohel                   0.001216
75) Artist_Tintorretto             0.001206
76) Artist_Titian                  0.001202
77) Artist_Vin                     0.001147
78) Artist_Vincent Van Gogh        0.001145
79) CanvasColor_BLACK              0.001128
80) CanvasColor_BLUE               0.001128
81) CanvasColor_BROWN              0.001111
82) CanvasColor_GOLD               0.001107
83) CanvasColor_GREEN              0.001004
84) CanvasColor_GREY               0.000915
85) CanvasColor_MAROON             0.000857
86) CanvasColor_NOT AVAIL          0.000775
87) CanvasColor_ORANGE             0.000688
88) CanvasColor_OTHER              0.000672
89) CanvasColor_PURPLE             0.000657
90) CanvasColor_RED                0.000596
91) CanvasColor_SILVER             0.000498
92) CanvasColor_WHITE              0.000473
93) CanvasColor_YELLOW             0.000454
94) Market_Non Commercial          0.000411
95) Nationality_OTHER              0.000404
96) Nationality_OTHER ASIAN        0.000344
97) Nationality_TOP LINE ASIAN     0.000294
98) Auction_OTHER                  0.000234
99) Auction_Sotheby's              0.000232
100) 2001                           0.000189
101) 2002                           0.000162
102) 2003                           0.000159
103) 2004                           0.000139
104) 2005                           0.000114
105) 2006                           0.000107
106) 2007                           0.000095
107) 2008                           0.000080
108) 2009                           0.000032
109) 2010                           0.000009
```

```
110) purchase_Month                    0.000000
111) purchase_Day                      0.000000
112) purchase_Year                     0.000000
```

```
In [108]: # plot the importance matrix
          plt.figure(figsize=(25,10))
          plt.title('Feature Importance')
          plt.bar(range(X.shape[1]),
                 importances[indices],
                 color='lightblue',
                 align='center')
          plt.xticks(range(X.shape[1]),
                    feat,rotation=90)
          plt.xlim([-1,X.shape[1]])
          plt.tight_layout()
          plt.show()
```




```
In [109]: # use the first 25 variables
          top20feat = feat[:25]
          clf_rf = RandomForestClassifier(n_estimators=10, max_depth=None,
                                          min_samples_split=2, random_state=0)
          scores_rf = cross_val_score(clf_rf, X[top20feat], y,cv=5).mean()
          print(scores_rf)
```

```
0.9015042485426108
```

SVM needs a lot of time for training, so here I comment the code

```
In [110]: ######SVM
```

```
In [111]: #svmlinearrbfsigmoid
          # from sklearn import svm
          # clf_linear = svm.SVC(kernel='linear')
          # clf_rbf = svm.SVC(kernel='rbf')
          # clf_sigmoid = svm.SVC(kernel='sigmoid')

          # score_svm_linear = cross_val_score(clf_linear, X, y,cv=5).mean()
          # print(score_svm_linear)
          # score_svm_rbf = cross_val_score(clf_linear, X, y,cv=5).mean()
          # print(score_svm_rbf)
          # score_svm_sigmoid = cross_val_score(clf_linear, X, y,cv=5).mean()
          # print(score_svm_sigmoid)

          # score_svm = max(score_svm_linear,score_svm_rbf,score_svm_sigmoid)
          # model_name.append('svm')
          # model_accuracy.append(score_svm)
          # score_svm

In [112]: ####xgboost

In [113]: # from sklearn.cross_validation import train_test_split
          import xgboost as xgb
          test_size = int(data.shape[0] * 0.33)
          test_X, train_X = split_set(X, test_size)
          test_Y, train_Y = split_set(y, test_size)
          # train_X,test_X,train_Y,test_Y = train_test_split(X, y, test_size=0.33, random_stat
          xg_train = xgb.DMatrix(train_X, label=train_Y)
          xg_test = xgb.DMatrix(test_X, label=test_Y)

          # setup parameters for xgboost
          selection = dict()
          param = {}
          # use softmax multi-class classification
          param['objective'] = 'multi:softprob'
          # scale weight of positive examples
          param['num_class'] = 2
          param['silent'] = 0
          param['eta'] = 0.05
          param['max_depth'] = 1
          param['nthread'] = 1
          param['eval_metric']='mlogloss'
          num_round = 30
          watchlist = [ (xg_train,'train'), (xg_test, 'test') ]
          xgbst = xgb.train(param, xg_train, num_round, watchlist )

          yprob = xgbst.predict(xg_test)
          ylabel = np.argmax(yprob, axis=1)  # return the index of the biggest pro
          xg_accuracy = (ylabel == test_Y).mean()
```

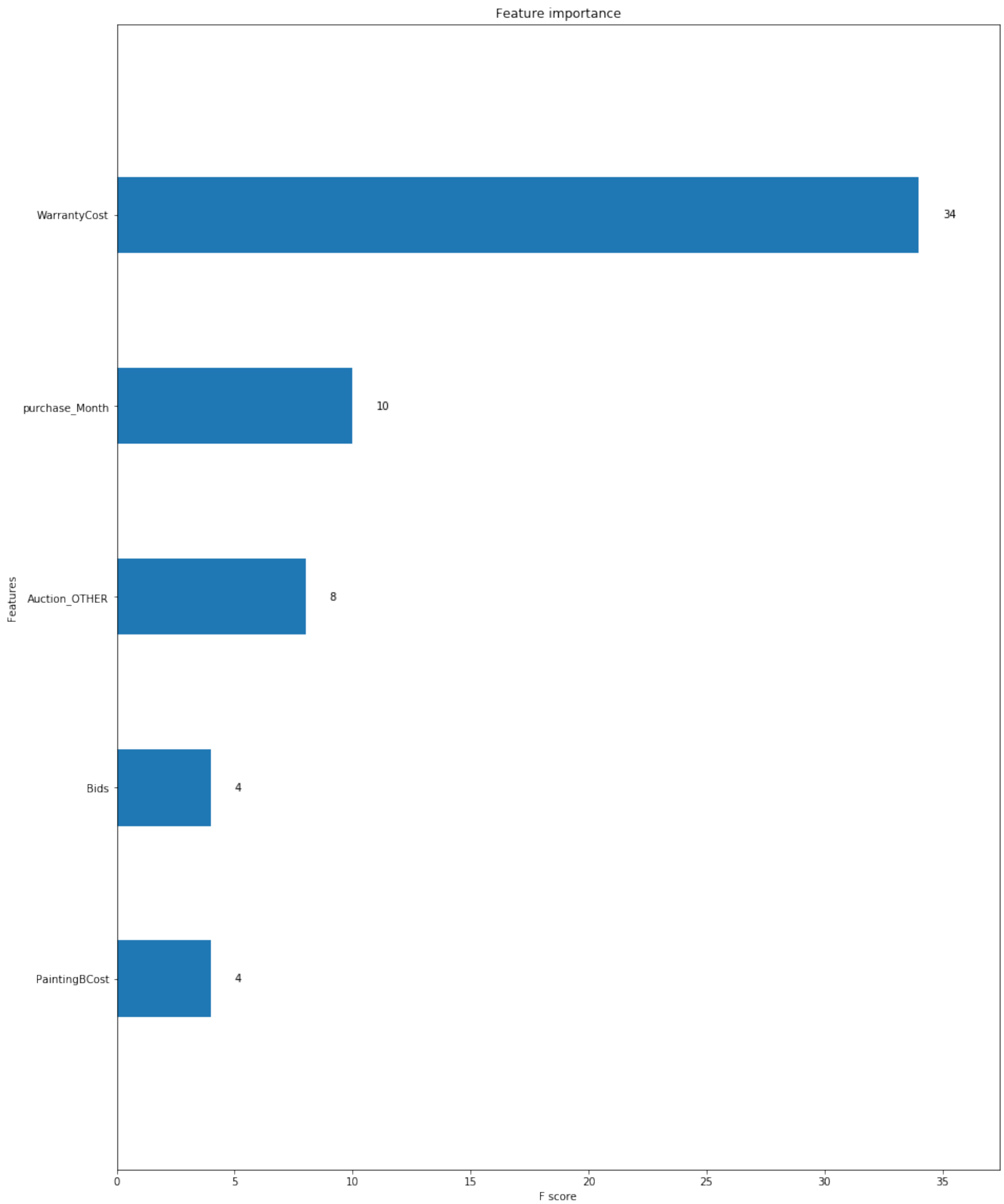
```python
# for eta in [0.05,0.1,0.2,0.3]:
#     param['eta'] = eta
#     for max_depth in range(1,40,2):
#         param['max_depth'] = max_depth
#         for nthread in range(1,11,1):
#             param['nthread'] = nthread
#             # param['eval_metric']='mlogloss'
#             param['eval_metric']='mlogloss'
#             num_round = 30

#             watchlist = [ (xg_train,'train'), (xg_test, 'test') ]
#             xgbst = xgb.train(param, xg_train, num_round, watchlist )

#             yprob = xgbst.predict(xg_test)
#             ylabel = np.argmax(yprob, axis=1)  # return the index of the biggest p
#             xg_accuracy = (ylabel == test_Y).mean()
# #             selection.append({'xg_accuracy':xg_accuracy,'eta':eta,'max_depth':ma
#             print(xg_accuracy,eta,max_depth,nthread)
# 0.904284685549688 0.05 1 1
```

```
[17:06:50] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:50] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[0]	train-mlogloss:0.661168	test-mlogloss:0.661329
[17:06:50] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:50] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[1]	train-mlogloss:0.632257	test-mlogloss:0.63257
[17:06:50] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:50] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[2]	train-mlogloss:0.606037	test-mlogloss:0.606496
[17:06:50] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[3]	train-mlogloss:0.582196	test-mlogloss:0.582793
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[4]	train-mlogloss:0.560466	test-mlogloss:0.561195
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[5]	train-mlogloss:0.540621	test-mlogloss:0.541478
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[6]	train-mlogloss:0.522467	test-mlogloss:0.523437
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[7]	train-mlogloss:0.505834	test-mlogloss:0.506922
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned no
[8]	train-mlogloss:0.490576	test-mlogloss:0.491775
```

```
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[9]        train-mlogloss:0.476562       test-mlogloss:0.477872
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[10]        train-mlogloss:0.46368       test-mlogloss:0.465089
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[11]        train-mlogloss:0.451828       test-mlogloss:0.453336
[17:06:51] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[12]        train-mlogloss:0.440917       test-mlogloss:0.442526
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[13]        train-mlogloss:0.430865       test-mlogloss:0.432566
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[14]        train-mlogloss:0.421601       test-mlogloss:0.423395
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[15]        train-mlogloss:0.41306       test-mlogloss:0.414941
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[16]        train-mlogloss:0.405183       test-mlogloss:0.407148
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17]        train-mlogloss:0.397917       test-mlogloss:0.399969
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[18]        train-mlogloss:0.391213       test-mlogloss:0.393348
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[19]        train-mlogloss:0.385027       test-mlogloss:0.38724
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[20]        train-mlogloss:0.37932       test-mlogloss:0.381613
[17:06:52] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[21]        train-mlogloss:0.374053       test-mlogloss:0.37642
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[22]        train-mlogloss:0.369194       test-mlogloss:0.371635
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[23]        train-mlogloss:0.364711       test-mlogloss:0.367227
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[24]        train-mlogloss:0.360575       test-mlogloss:0.363162
```

```
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[25]        train-mlogloss:0.356761        test-mlogloss:0.359413
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[26]        train-mlogloss:0.353244        test-mlogloss:0.355958
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[27]        train-mlogloss:0.350002        test-mlogloss:0.352784
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[28]        train-mlogloss:0.347014        test-mlogloss:0.349859
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:53] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[29]        train-mlogloss:0.344262        test-mlogloss:0.347168
```

In [114]:
```
from xgboost import plot_importance
fig,ax = plt.subplots(figsize=(15,20))
plot_importance(xgbst,height=0.4,max_num_features=60,ax=ax,grid=False)
plt.show()
```




```
In [115]: yprob = xgbst.predict(xg_test)
          ylabel = np.argmax(yprob, axis=1)  # return the index of the biggest pro
          xg_accuracy = (ylabel == test_Y).mean()
          xg_accuracy

Out[115]: 0.903084493518963

In [116]: xgbst_new = xgb.train(param, xg_train, 30, watchlist )
```

```
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[0]         train-mlogloss:0.661168       test-mlogloss:0.661329
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[1]         train-mlogloss:0.632257       test-mlogloss:0.63257
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[2]         train-mlogloss:0.606037       test-mlogloss:0.606496
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[3]         train-mlogloss:0.582196       test-mlogloss:0.582793
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[4]         train-mlogloss:0.560466       test-mlogloss:0.561195
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:54] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[5]         train-mlogloss:0.540621       test-mlogloss:0.541478
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[6]         train-mlogloss:0.522467       test-mlogloss:0.523437
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[7]         train-mlogloss:0.505834       test-mlogloss:0.506922
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[8]         train-mlogloss:0.490576       test-mlogloss:0.491775
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[9]         train-mlogloss:0.476562       test-mlogloss:0.477872
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[10]        train-mlogloss:0.46368        test-mlogloss:0.465089
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[11]        train-mlogloss:0.451828       test-mlogloss:0.453336
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[12]        train-mlogloss:0.440917       test-mlogloss:0.442526
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[13]        train-mlogloss:0.430865       test-mlogloss:0.432566
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[14]        train-mlogloss:0.421601       test-mlogloss:0.423395
[17:06:55] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[15]        train-mlogloss:0.41306        test-mlogloss:0.414941
```

```
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[16]        train-mlogloss:0.405183        test-mlogloss:0.407148
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17]        train-mlogloss:0.397917        test-mlogloss:0.399969
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[18]        train-mlogloss:0.391213        test-mlogloss:0.393348
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[19]        train-mlogloss:0.385027        test-mlogloss:0.38724
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[20]        train-mlogloss:0.37932        test-mlogloss:0.381613
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[21]        train-mlogloss:0.374053        test-mlogloss:0.37642
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[22]        train-mlogloss:0.369194        test-mlogloss:0.371635
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:56] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[23]        train-mlogloss:0.364711        test-mlogloss:0.367227
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[24]        train-mlogloss:0.360575        test-mlogloss:0.363162
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[25]        train-mlogloss:0.356761        test-mlogloss:0.359413
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[26]        train-mlogloss:0.353244        test-mlogloss:0.355958
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[27]        train-mlogloss:0.350002        test-mlogloss:0.352784
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[28]        train-mlogloss:0.347014        test-mlogloss:0.349859
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:06:57] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[29]        train-mlogloss:0.344262        test-mlogloss:0.347168
```

In [117]:
```
yprob = xgbst_new.predict(xg_test)
ylabel = np.argmax(yprob, axis=1)  # return the index of the biggest pro
xg_accuracy_new = (ylabel == test_Y).mean()
model_name.append('xgboost')
```

```
    model_accuracy.append(xg_accuracy_new)
    xg_accuracy_new
```

Out[117]: 0.903084493518963

```
In [118]: ##neural network
```

```
In [120]: from sklearn.neural_network import MLPClassifier
          from sklearn.model_selection import cross_val_score
          clf_nn = MLPClassifier(hidden_layer_sizes=(32, 8), random_state=1)

          scores_nn = cross_val_score(clf_nn, X, y,cv=5).mean()
          print(scores_nn)
```

0.9042506703127211

```
In [121]: model_name.append('neural network')
          model_accuracy.append(scores_nn)
```

```
In [122]: # transform test dataset into the target foramt.
          def transform(data, _cates, cates_to_id, id_to_cates, unique_years, dic):
              data = data.drop('RT13Id', axis=1)
              categories = data[['PaintingName', 'SubType']]
              one_hot_values = np.zeros([data.shape[0], 14])
              for idx, row in categories.iterrows():
                  extract_one_cate(row[0], row[1], one_hot_values, idx)
              # the last column means the category T is not given
              for i in range(14):
                  data["T{}".format(i + 1)] = one_hot_values[:, i]

              # 0 here means not given
              num_check = np.zeros(data.shape[0])
              names = data[['PaintingName']]
              for idx, row in names.iterrows():
                  extract_one_num_check(row[0], num_check, idx)
                  data["num_checked"] = num_check
              # zeros here means not given
              sizes = np.zeros(data.shape[0])
              names = data[['PaintingName']]
              for idx, row in names.iterrows():
                  extract_one_size(row[0], sizes, idx)

              data["given_size"] = sizes

              types = data[['SubType']]
              cates_2 = np.zeros(data.shape[0], np.int8)
              for idx, row in types.iterrows():
                  extract_one_cate_2(row[0], _cates, cates_to_id, cates_2, idx)
```

```
        df_tmp = pd.DataFrame({'cate_': [id_to_cates[t] for t in cates_2]})
        df_tmp = pd.get_dummies(df_tmp, prefix=['cate_'], drop_first=True)
        data = pd.concat([data, df_tmp], axis=1)
        prices = ['MMRAcquisitionAuctionAveragePrice', 'MMRAcquisitionAuctionCleanPrice'
               'MMRAcquisitionRetailAveragePrice', 'MMRAcquisitonRetailCleanPrice',
               'MMRCurrentAuctionAveragePrice', 'MMRCurrentAuctionCleanPrice',
               'MMRCurrentRetailAveragePrice', 'MMRCurrentRetailCleanPrice',]

#     for p in prices:
#         data = data.loc[data[p] != 0]
#         data["{}_ratio".format(p)] = data["Bids"] / data[p]

#     factors = ["Artist", "CanvasColor","Market","Nationality","Auction"]
#     for f in factors:
#         df_tmp = data[[f]]
#         df_tmp = pd.get_dummies(df_tmp, prefix=[f], drop_first=True)
#         data = pd.concat([data, df_tmp], axis=1)

        f = "PaintingYear"

        years = [dic[y] for y in data[f].tolist()]
        yys = np.zeros([data.shape[0], len(dic)])
        for i, y in enumerate(years):
            yys[i][y] = 1

        for i, k in enumerate(dic):
            data[str(k)] = yys[:, i]
        # separate purchase date into year, month, day
        for idx in range(len(purdate)):
            data[purdate[idx]] = data['PurchDate'].map(lambda x: int(x.split('/')[idx]))

        return data
```

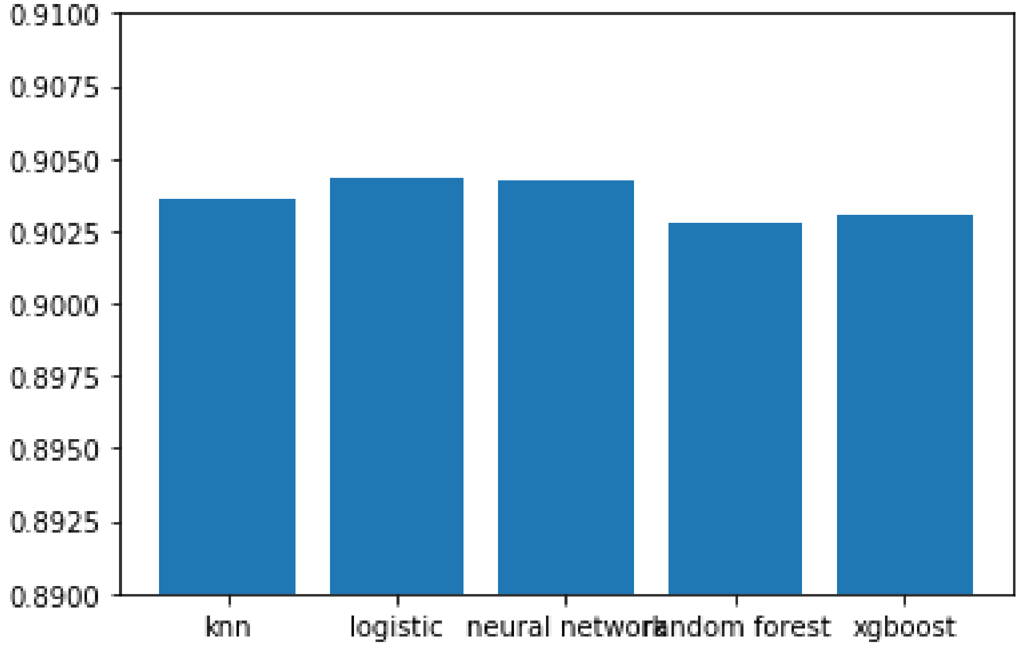
In [123]:
```
model_name
```

Out[123]: ['logistic', 'knn', 'random forest', 'xgboost', 'neural network']

In [124]:
```
model_accuracy
```

Out[124]:
```
[0.9042922827965787,
 0.9036265003708437,
 0.9027318222276595,
 0.903084493518963,
 0.9042506703127211]
```

In [125]:
```
plt.bar(model_name,model_accuracy)
plt.ylim(0.89,0.91)
plt.show()
```




```
In [ ]: ##here, we choose knn as the ultimate model for prediction
```

```
In [ ]: ##because "Artist", "Auction","CanvasColor","Market","Nationality" are all not
        ## the top 100 important features and the features in training dataset
        ## and test dataset are not the same. so here, we retrain the model
        ## by excluding these feature-related variables.
```

```
In [ ]: # data_train_new = pd.read_csv("train_dataset.csv")
        # data_train_new = transform(data_train_new, _cates, cates_to_id,
        #                            id_to_cates, unique_years, dic)
```

```
In [ ]: # data_train_new = data_train_new.drop(features_not_used,axis=1)
        # data_train_new = data_train_new.dropna(axis=0)
        # data_train_new.shape
```

```
In [126]: data_train_new = pd.read_csv("../input/train_dataset.csv")
          data_train_new = transform(data_train_new, _cates, cates_to_id,
                                     id_to_cates, unique_years, dic)

          data_train_new = data_train_new.drop(features_not_used,axis=1)
          data_train_new = data_train_new.fillna(method='ffill')
          data_train_new.shape

          train_label = data_train_new['IsBadBuy']
```

```
In [67]: #knn
```

In [68]:
```
model_knn_new = KNeighborsClassifier(n_neighbors = score.index(max(score)))
model_knn_new = model_knn_new.fit(data_train_new.drop('IsBadBuy',axis=1),
                                  train_label)
```

In [69]:
```
cross_val_score(model_knn_new,data_train_new,train_label,cv=5).mean()
```

Out[69]:
```
0.8728272791746046
```

In [139]:
```
model_knn_new = KNeighborsClassifier(n_neighbors = score.index(max(score)))
model_knn_new = model_knn_new.fit(data_train_new.drop('IsBadBuy',axis=1),
                                  train_label)
```

In [70]:
```
#logistic
```

In [130]:
```
test_model = LogisticRegression().fit(data_train_new.drop('IsBadBuy',axis=1),train_la
```

In [131]:
```
#xgboost
test_size = int(data_train_new.shape[0] * 0.33)
test_X, train_X = split_set(data_train_new.drop('IsBadBuy',axis=1), test_size)
test_Y, train_Y = split_set(train_label, test_size)
# train_X,test_X,train_Y,test_Y = train_test_split(X, y, test_size=0.33, random_stat
xg_train = xgb.DMatrix(train_X, label=train_Y)
xg_test = xgb.DMatrix(test_X, label=test_Y)
watchlist = [ (xg_train,'train'), (xg_test, 'test') ]
xgbst = xgb.train(param, xg_train, 100, watchlist)
yprob = xgbst.predict(xg_test)
ylabel = np.argmax(yprob, axis=1)  # return the index of the biggest pro
xg_accuracy = (ylabel == test_Y).mean()
xg_accuracy
```

```
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[0]     train-mlogloss:0.665272        test-mlogloss:0.665538
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[1]     train-mlogloss:0.640077        test-mlogloss:0.640596
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[2]     train-mlogloss:0.617242        test-mlogloss:0.618008
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[3]     train-mlogloss:0.596497        test-mlogloss:0.597495
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[4]     train-mlogloss:0.577613        test-mlogloss:0.578837
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[5]     train-mlogloss:0.560394        test-mlogloss:0.561838
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
```

```
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[6]       train-mlogloss:0.54467       test-mlogloss:0.546322
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[7]       train-mlogloss:0.530292      test-mlogloss:0.532142
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[8]       train-mlogloss:0.517131      test-mlogloss:0.519175
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[9]       train-mlogloss:0.505073      test-mlogloss:0.507303
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[10]       train-mlogloss:0.494016      test-mlogloss:0.496429
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[11]       train-mlogloss:0.483871      test-mlogloss:0.486456
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[12]       train-mlogloss:0.474558      test-mlogloss:0.477312
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[13]       train-mlogloss:0.466004      test-mlogloss:0.46892
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[14]       train-mlogloss:0.458144      test-mlogloss:0.46122
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[15]       train-mlogloss:0.450921      test-mlogloss:0.454148
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[16]       train-mlogloss:0.444281      test-mlogloss:0.447655
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17]       train-mlogloss:0.438177      test-mlogloss:0.441695
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:33] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[18]       train-mlogloss:0.432565      test-mlogloss:0.43622
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[19]       train-mlogloss:0.427405      test-mlogloss:0.431196
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[20]       train-mlogloss:0.422661      test-mlogloss:0.426583
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[21]       train-mlogloss:0.4183       test-mlogloss:0.422347
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
```

```
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[22]        train-mlogloss:0.41429       test-mlogloss:0.418458
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[23]        train-mlogloss:0.410606       test-mlogloss:0.414892
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[24]        train-mlogloss:0.40722       test-mlogloss:0.411619
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[25]        train-mlogloss:0.404109       test-mlogloss:0.40862
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[26]        train-mlogloss:0.401252       test-mlogloss:0.405871
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[27]        train-mlogloss:0.398628       test-mlogloss:0.403351
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[28]        train-mlogloss:0.396219       test-mlogloss:0.40104
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[29]        train-mlogloss:0.394009       test-mlogloss:0.398929
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[30]        train-mlogloss:0.39198       test-mlogloss:0.396996
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[31]        train-mlogloss:0.39012       test-mlogloss:0.395228
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[32]        train-mlogloss:0.388415       test-mlogloss:0.393607
[17:10:34] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[33]        train-mlogloss:0.386851       test-mlogloss:0.392131
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[34]        train-mlogloss:0.385419       test-mlogloss:0.390783
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[35]        train-mlogloss:0.384107       test-mlogloss:0.38955
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[36]        train-mlogloss:0.382905       test-mlogloss:0.388427
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[37]        train-mlogloss:0.381806       test-mlogloss:0.387401
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
```

```
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[38]        train-mlogloss:0.3808       test-mlogloss:0.386464
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[39]        train-mlogloss:0.379879     test-mlogloss:0.385613
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[40]        train-mlogloss:0.379037     test-mlogloss:0.384841
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[41]        train-mlogloss:0.378268     test-mlogloss:0.384138
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[42]        train-mlogloss:0.377565     test-mlogloss:0.383499
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[43]        train-mlogloss:0.376922     test-mlogloss:0.382917
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[44]        train-mlogloss:0.376336     test-mlogloss:0.382385
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[45]        train-mlogloss:0.3758       test-mlogloss:0.381908
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[46]        train-mlogloss:0.375311     test-mlogloss:0.381475
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[47]        train-mlogloss:0.374865     test-mlogloss:0.381083
[17:10:35] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[48]        train-mlogloss:0.374458     test-mlogloss:0.380726
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[49]        train-mlogloss:0.374087     test-mlogloss:0.380403
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[50]        train-mlogloss:0.373748     test-mlogloss:0.380112
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[51]        train-mlogloss:0.373439     test-mlogloss:0.379851
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[52]        train-mlogloss:0.373158     test-mlogloss:0.379616
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[53]        train-mlogloss:0.372901     test-mlogloss:0.379404
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
```

```
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[54]	train-mlogloss:0.372667	test-mlogloss:0.379213
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[55]	train-mlogloss:0.372454	test-mlogloss:0.379035
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[56]	train-mlogloss:0.37226	test-mlogloss:0.378879
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[57]	train-mlogloss:0.372082	test-mlogloss:0.378737
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[58]	train-mlogloss:0.371921	test-mlogloss:0.378612
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[59]	train-mlogloss:0.371774	test-mlogloss:0.378498
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[60]	train-mlogloss:0.37164	test-mlogloss:0.378397
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[61]	train-mlogloss:0.371518	test-mlogloss:0.378309
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[62]	train-mlogloss:0.371406	test-mlogloss:0.37823
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[63]	train-mlogloss:0.371304	test-mlogloss:0.378155
[17:10:36] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[64]	train-mlogloss:0.371211	test-mlogloss:0.378094
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[65]	train-mlogloss:0.371127	test-mlogloss:0.378032
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[66]	train-mlogloss:0.371049	test-mlogloss:0.377982
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[67]	train-mlogloss:0.370978	test-mlogloss:0.377938
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[68]	train-mlogloss:0.370914	test-mlogloss:0.377897
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[69]	train-mlogloss:0.370855	test-mlogloss:0.377863
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
```

```
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[70]        train-mlogloss:0.370801        test-mlogloss:0.37783
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[71]        train-mlogloss:0.370751        test-mlogloss:0.377804
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[72]        train-mlogloss:0.370706        test-mlogloss:0.377779
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[73]        train-mlogloss:0.370664        test-mlogloss:0.377761
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[74]        train-mlogloss:0.370626        test-mlogloss:0.377745
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[75]        train-mlogloss:0.370591        test-mlogloss:0.377733
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[76]        train-mlogloss:0.370558        test-mlogloss:0.377719
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[77]        train-mlogloss:0.370528        test-mlogloss:0.377707
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[78]        train-mlogloss:0.370501        test-mlogloss:0.3777
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:37] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[79]        train-mlogloss:0.370475        test-mlogloss:0.377692
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[80]        train-mlogloss:0.370452        test-mlogloss:0.377681
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[81]        train-mlogloss:0.37043        test-mlogloss:0.377673
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[82]        train-mlogloss:0.37041        test-mlogloss:0.377671
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[83]        train-mlogloss:0.370391        test-mlogloss:0.377668
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[84]        train-mlogloss:0.370373        test-mlogloss:0.377668
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[85]        train-mlogloss:0.370357        test-mlogloss:0.377666
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
```

```
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[86]        train-mlogloss:0.370341        test-mlogloss:0.377667
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[87]        train-mlogloss:0.370327        test-mlogloss:0.377668
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[88]        train-mlogloss:0.370313        test-mlogloss:0.377672
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[89]        train-mlogloss:0.3703        test-mlogloss:0.377671
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[90]        train-mlogloss:0.370288        test-mlogloss:0.377669
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[91]        train-mlogloss:0.370277        test-mlogloss:0.377666
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[92]        train-mlogloss:0.370266        test-mlogloss:0.377668
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[93]        train-mlogloss:0.370256        test-mlogloss:0.377675
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[94]        train-mlogloss:0.370246        test-mlogloss:0.37768
[17:10:38] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[95]        train-mlogloss:0.370237        test-mlogloss:0.377684
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[96]        train-mlogloss:0.370228        test-mlogloss:0.377688
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[97]        train-mlogloss:0.37022        test-mlogloss:0.37769
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[98]        train-mlogloss:0.370211        test-mlogloss:0.377692
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[17:10:39] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 2 extra nodes, 0 pruned noc
[99]        train-mlogloss:0.370203        test-mlogloss:0.377694
```

Out[131]: 0.8745477193190581

In [132]:
```
# Using KFold to train multiple models and average the result
test = pd.read_csv("../input/test_dataset.csv")
test_name = test.RT13Id
```

```
test = transform(test, _cates, cates_to_id,
                 id_to_cates, unique_years, dic)
test = test.drop(features_not_used,axis=1)
test = test.fillna(method='ffill')
test.shape
```

Out[132]: (21895, 52)

In [133]:
```
test_xgbst = xgb.DMatrix(test)
output_xgbst = xgbst.predict(xg_test)
output_xgbst = np.argmax(output_xgbst, axis=1)
```

In [134]:
```
# output = test_model.predict(test)
```

In [135]:
```
clf_nn = MLPClassifier(hidden_layer_sizes=(32, 8), random_state=1)

clf_nn.fit(data_train_new.drop('IsBadBuy',axis=1),train_label)
output = clf_nn.predict(test)
```

In [136]:
```
###knn
output = model_knn_new.predict(test)
```

In [ ]:
```
# # LGBM feature importance analysis
# def lgb_model(x_train, x_test, y_train, y_test):
#     params = {
#         "objective" : "binary",
#         "metric" : "binary_error",
#         "num_leaves" : 20,
#         "learning_rate" : 0.1,
#         "bagging_fraction" : 0.8,
#         "feature_fraction" : 0.8,
#         "bagging_frequency" : 5,
#         "bagging_seed" : 42,
#         "verbosity" : -1,
#         "max_depth" : 6
#     }
#     train = lgb.Dataset(x_train, label=y_train)
#     test = lgb.Dataset(x_test, label=y_test)
#     model = lgb.train(params, train, 1000, valid_sets=[test], early_stopping_rounds=
#     return model
```

In [ ]:
```
# num_kfold = 3
# kf = model_selection.KFold(n_splits=num_kfold, shuffle=True, random_state=666)

# target = pd.DataFrame(columns=['RT13Id', 'IsBadBuy'])
# target['RT13Id'] = test['RT13Id']
# target['IsBadBuy'] = 0
# for t_idx, v_idx in kf.split(x_train):
#     t_x, v_x = x_train.loc[t_idx, :], x_train.loc[v_idx, :]
```

```
#     t_y, v_y = y_train[t_idx], y_train[v_idx]
#     model = lgb_model(t_x, v_x, t_y, v_y)
#     target['IsBadBuy'] += model.predict(x_test)
```

In [ ]:
```
# target['IsBadBuy'] = round(target['IsBadBuy'] / float(num_kfold), 0)
# target.to_csv('submission.csv', index=False)
```

In [132]:
```
sum(train_label.tolist())/len(train_label)
```

Out[132]: 0.12310131537738804

In [79]:
```
# output.to_csv("../output/Shin_Gao_tg2618.csv",index=False,sep=',',encoding='utf-8-s
```

In [137]:
```
output = pd.DataFrame(output)
test_output = pd.concat([pd.DataFrame(test_name),output],axis=1)
test_output.rename(columns={test_output.columns[1]:'IsBadBuy'},inplace=True)
test_output
```

Out[137]:

	RT13Id	IsBadBuy
0	1	0
1	7	0
2	8	0
3	20	0
4	21	0
5	25	0
6	27	0
7	30	0
8	32	0
9	33	0
10	36	1
11	41	0
12	43	0
13	47	0
14	51	0
15	56	0
16	59	0
17	62	0
18	67	0
19	70	0
20	71	0
21	74	0
22	79	0
23	82	0
24	86	0
25	89	0
26	90	0
27	91	0
28	94	0
29	100	0

```
...        ...        ...
21865    72918          0
21866    72922          0
21867    72924          0
21868    72930          0
21869    72933          0
21870    72935          0
21871    72943          0
21872    72946          0
21873    72947          0
21874    72949          0
21875    72952          0
21876    72956          0
21877    72958          0
21878    72964          0
21879    72971          0
21880    72972          0
21881    72974          0
21882    72976          1
21883    72977          0
21884    72978          0
21885    72982          0
21886    72983          0
21887    72985          0
21888    72986          0
21889    72989          0
21890    72997          0
21891    73001          0
21892    73002          0
21893    73007          0
21894    73013          0

[21895 rows x 2 columns]
```

```
In [138]: test_output.to_csv("../output/Shin_Gao_tg2618.csv",index=False,sep=',',encoding='utf
```

```
In [84]: # data_train_new.to_csv("../output/final_train.csv",index=False,sep=',',encoding='utf
         # test.to_csv("../output/final_test.csv",index=False,sep=',',encoding='utf-8-sig')
```