In [3]:
```
import os
import sys
sys.path.append('/home/pn4twfnsg7/atec_project/train/harper/lib/python3.6/s
import json
from pandas.io.json import json_normalize

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
color = sns.color_palette()
sns.set(style='whitegrid', color_codes=True)
from pandas_profiling import ProfileReport
import sweetviz as sv

import warnings
def ignore_warn(*args,**kwargs):
    pass
warnings.warn=ignore_warn
```

1 数据集获取

1.1 .json格式转DataFrame

In [5]:
```
train_data_path = '/mnt/atec/train.jsonl'
```

读一行看下基本内容

In [3]:
```
with open(train_data_path, 'r', encoding='utf-8') as fp:
    for line in fp.readlines():
        first_line_data = json.loads(line)
        print(first_line_data)
        break
```

```
{'id': 0, 'x0': 0, 'x1': 55, 'x2': -100, 'x3': 0.0, 'x4': 0.0, 'x5': 0.0,
'x6': -100, 'x7': 0.153692461384769, 'x8': -1.0, 'x9': 1.0, 'x10': 22, 'x
11': 0.0, 'x12': 0, 'x13': 0, 'x14': 3, 'x15': -1, 'x16': -1, 'x17': 2,
'x18': -1.0, 'x19': 2, 'x20': -1, 'x21': 0.0, 'x22': 20.0, 'x23': 422, 'x
24': 5, 'x25': 0, 'x26': 0.0, 'x27': 1050.65, 'x28': 7.540229885057474,
'x29': 0.0, 'x30': 0.0, 'x31': 0, 'x32': -100.0, 'x33': -100.0, 'x34': -1
00, 'x35': -100.0, 'x36': 502, 'x37': 0.0, 'x38': 0, 'x39': 0.0, 'x40': 1
00.0, 'x41': 1.0, 'x42': -1.0, 'x43': 0.0, 'x44': -1, 'x45': 4, 'x46': 0,
'x47': 0.0, 'x48': -1.0, 'x49': 3, 'x50': 0.0, 'x51': 0.0, 'x52': 0.0, 'x
53': 2, 'x54': 0, 'x55': 0, 'x56': 0.0, 'x57': 0, 'x58': 40, 'x59': 349,
'x60': 0, 'x61': 0.0, 'x62': 549.65, 'x63': 0.0, 'x64': 0.0, 'x65': 0.0,
'x66': -1.0, 'x67': 0.0, 'x68': -1.0, 'x69': 0.0, 'x70': -1.0, 'x71': -1,
'x72': -99.0, 'x73': 1, 'x74': 0.0, 'x75': 0.0, 'x76': 0.0, 'x77': 0, 'x7
8': 0.154496541122214, 'x79': 0, 'x80': 1, 'x81': 0.0, 'x82': 8, 'x83':
0.0, 'x84': 2, 'x85': -100, 'x86': 0.0, 'x87': 2, 'x88': 1.93236328125,
'x89': -1.0, 'x90': 0, 'x91': -100, 'x92': 0.0, 'x93': -1, 'x94': 0.0, 'x
95': -100, 'x96': -1.002, 'x97': 0, 'x98': 0, 'x99': 0, 'x100': 0, 'x10
1': -99, 'x102': 100.0, 'x103': 5, 'x104': -1.0, 'x105': 0.0, 'x106': 54
9.65, 'x107': -100, 'x108': -1, 'x109': 650.65, 'x110': -100.0, 'x111': -
```

Notes:

- 数据是脱敏的
- 看起来都是数值(除了meme_polish)，但是有些应该是类别变量，比如label肯定就是类别变量
- id, x0~479, meme_polish, label

In [6]:
```
raw_data = pd.read_json(train_data_path,encoding='utf-8',lines=True)
```

In [5]:
```
raw_data.set_index(['id'],inplace=True)
```

2 数据特征分析

2.1 基本分析

In [5]:
```
raw_data
```

Out[5]:

id	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	...	x472	x473	x474
0	0	55	-100	0.000	0.000	0.000	-100	0.153692	-1.0	1.0	...	0	0.000	0
1	1	-2	-100	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	...	0	0.000	0
2	1	90	-100	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	...	0	0.000	0
3	1	178	-100	0.000	0.000	0.000	-100	0.054845	-1.0	0.0	...	1	0.000	0
4	1	69	-100	0.000	0.000	0.000	-100	0.005406	-1.0	0.5	...	2	0.000	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57763	0	-2	-100	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	...	0	-1.001	-1
57764	1	-2	-100	2.000	2.000	0.000	-100	0.138289	-1.0	-100.0	...	0	0.000	0
57765	1	179	-100	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	...	0	0.000	0
57766	1	-2	-100	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	...	0	-1.001	-1
57767	1	-2	-100	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	...	4	-1.001	-1

57768 rows × 482 columns

In [6]:
```
y_train = raw_data.label.values
train = raw_data
train.drop(['label'], axis=1, inplace=True)
print("tain data size is : {}".format(train.shape))
```

```
tain data size is : (57768, 481)
```

In [8]:
```
raw_data_describe = raw_data.describe()
```

In [9]:
```
raw_data_describe
```

Out[9]:

	x0	x1	x2	x3	x4	x5	
count	57768.000000	57768.000000	57768.000000	57768.000000	57768.000000	57768.000000	57768.
mean	0.463873	68.027610	-100.455027	8.448485	2.964358	2.385734	-88.
std	23.585945	80.501583	21.443722	87.338553	37.855035	34.836236	39.
min	-1111.000000	-1111.000000	-1111.000000	-1111.000000	-1111.000000	-1111.000000	-1111.
25%	1.000000	-2.000000	-100.000000	0.000000	0.000000	0.000000	-100.
50%	1.000000	28.000000	-100.000000	0.000000	0.000000	0.000000	-100.
75%	1.000000	162.000000	-100.000000	0.000000	0.000000	0.000000	-100.
max	1.000000	179.000000	-100.000000	3250.000000	300.000000	300.000000	0.

8 rows × 481 columns

In [10]:
```
raw_data_describe.to_csv('raw_data_describe.csv')
```

2.2 缺失值

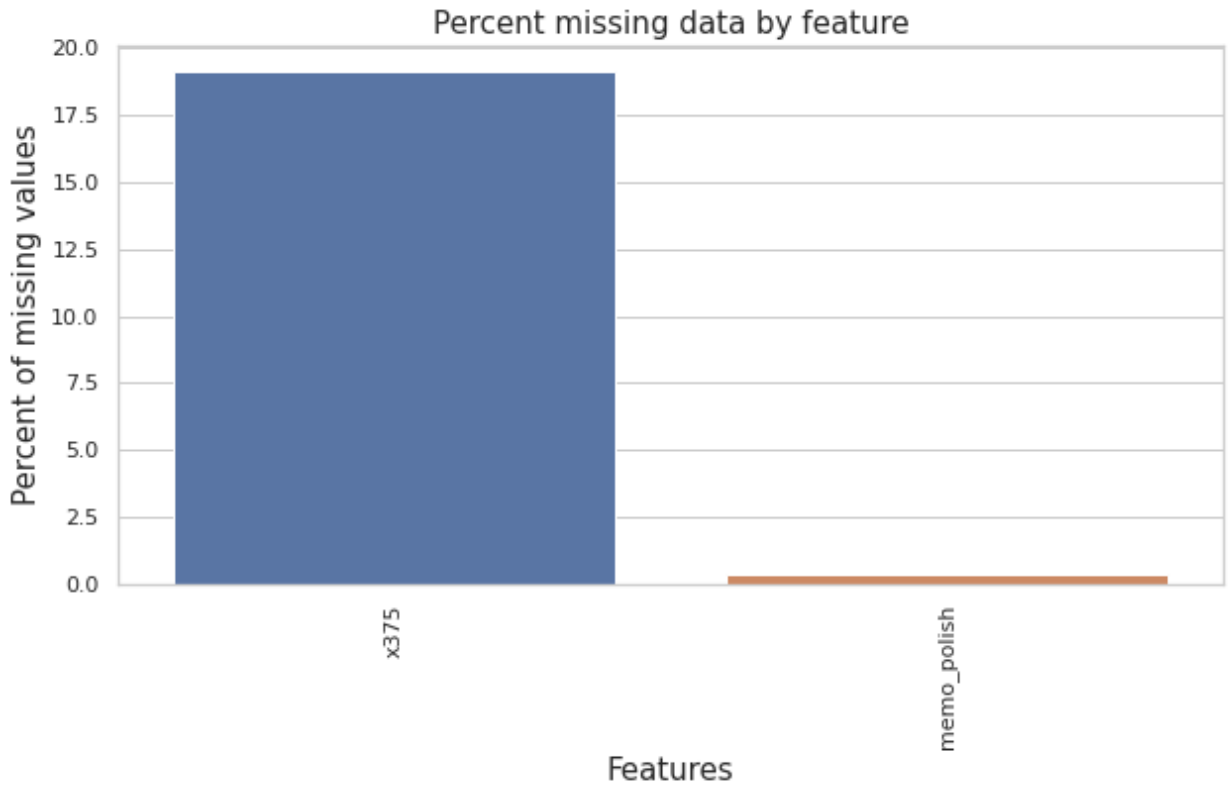
In [8]:
```
train_na = (train.isnull().sum() / len(train)) * 100
train_na = train_na.drop(train_na[train_na == 0].index).sort_values(ascendi
missing_data = pd.DataFrame({'Missing Ratio' :train_na})
missing_data.head(20)
```

Out[8]:

	Missing Ratio
x375	19.126506
memo_polish	0.334095

In [18]:
```
f, ax = plt.subplots(figsize=(10, 5))
plt.xticks(rotation='90')
sns.barplot(x=train_na.index, y=train_na)
plt.xlabel('Features', fontsize=15)
plt.ylabel('Percent of missing values', fontsize=15)
plt.title('Percent missing data by feature', fontsize=15)
```

Out[18]:
```
Text(0.5, 1.0, 'Percent missing data by feature')
```



3 数据预处理

3.1 去除所有含有-1111的数据

26条

In [10]:
```
raw_data
```

Out[10]:

id	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	...	x472	x473	x474
0	0	55	-100	0.000	0.000	0.000	-100	0.153692	-1.0	1.0	...	0	0.000	0
1	1	-2	-100	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	...	0	0.000	0
2	1	90	-100	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	...	0	0.000	0
3	1	178	-100	0.000	0.000	0.000	-100	0.054845	-1.0	0.0	...	1	0.000	0
4	1	69	-100	0.000	0.000	0.000	-100	0.005406	-1.0	0.5	...	2	0.000	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57763	0	-2	-100	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	...	0	-1.001	-1
57764	1	-2	-100	2.000	2.000	0.000	-100	0.138289	-1.0	-100.0	...	0	0.000	0
57765	1	179	-100	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	...	0	0.000	0
57766	1	-2	-100	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	...	0	-1.001	-1
57767	1	-2	-100	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	...	4	-1.001	-1

57768 rows × 482 columns

In [6]:
```
raw_data = raw_data.drop(raw_data[(raw_data['x0']==-1111)].index)
```

In [12]:
```
raw_data_report_d1111 = sv.analyze(raw_data, pairwise_analysis='off')
raw_data_report_d1111.show_html('drop_1111.html')
```

```
                                                |           | [  0%]   00:00
-> (? left)

Report drop_1111.html was generated! NOTEBOOK/COLAB USERS: the web browse
r MAY not pop up, regardless, the report IS saved in your notebook/colab
files.
```

3.2 去掉此时只有一个值的变量

In [7]:
```
columns = raw_data.columns.values.tolist()
drop_list = []
```

In [8]:
```
for col in columns:
    num = len(raw_data[col].unique())
    if num == 1:
        drop_list.append(col)
```

In [9]:
```
drop_list
```

Out[9]:
```
['x2',
 'x55',
 'x91',
 'x96',
 'x107',
 'x184',
 'x198',
 'x207',
 'x209',
 'x261',
 'x319',
 'x384',
 'x452',
 'x456']
```

In [10]:
```
raw_data = raw_data.drop(columns = drop_list)
```

In [31]:
```
raw_data
```

Out[31]:

id	x0	x1	x3	x4	x5	x6	x7	x8	x9	x10	...	x472	x473	x474
0	0	55	0.000	0.000	0.000	-100	0.153692	-1.0	1.0	22	...	0	0.000	0
1	1	-2	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	19	...	0	0.000	0
2	1	90	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	15	...	0	0.000	0
3	1	178	0.000	0.000	0.000	-100	0.054845	-1.0	0.0	15	...	1	0.000	0
4	1	69	0.000	0.000	0.000	-100	0.005406	-1.0	0.5	14	...	2	0.000	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57763	0	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	10	...	0	-1.001	-1
57764	1	-2	2.000	2.000	0.000	-100	0.138289	-1.0	-100.0	2	...	0	0.000	0
57765	1	179	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	1	...	0	0.000	0
57766	1	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	22	...	0	-1.001	-1
57767	1	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	20	...	4	-1.001	-1

57742 rows × 468 columns

In [26]:
```
raw_data['label'].unique()
```

Out[26]:
```
array([-1,  0,  1])
```

In [11]:
```
df_1 = raw_data[raw_data.label == 1]
df_0 = raw_data[raw_data.label == 0]
df__1 = raw_data[raw_data.label == -1]
```

In [33]:
```
raw_data_report_df_1 = sv.analyze(df_1, pairwise_analysis='off')
raw_data_report_df_1.show_html('df_1.html')
```

```
                                             |          | [  0%]   00:00
-> (? left)

Report df_1.html was generated! NOTEBOOK/COLAB USERS: the web browser MAY
not pop up, regardless, the report IS saved in your notebook/colab files.
```

In [12]:
```
raw_data_report_df_0 = sv.analyze(df_0, pairwise_analysis='off')
raw_data_report_df_0.show_html('df_0.html')
```

```
                                             |          | [  0%]   00:00
-> (? left)

Report df_0.html was generated! NOTEBOOK/COLAB USERS: the web browser MAY
not pop up, regardless, the report IS saved in your notebook/colab files.
```

In [13]:
```
raw_data_report_df__1 = sv.analyze(df__1, pairwise_analysis='off')
raw_data_report_df__1.show_html('df_-1.html')
```

```
                                             |          | [  0%]   00:00
-> (? left)

Report df_-1.html was generated! NOTEBOOK/COLAB USERS: the web browser MA
Y not pop up, regardless, the report IS saved in your notebook/colab file
s.
```

3.3 处理缺失值

主要是处理x375这个变量，baseline嘛，先都弄成1吧，简单粗暴一点

In [15]:
```
raw_data['x375'] = raw_data['x375'].fillna(raw_data['x375'].mode()[0])
```

In [16]:
```
raw_data
```

Out[16]:

id	x0	x1	x3	x4	x5	x6	x7	x8	x9	x10	...	x472	x473	x474
0	0	55	0.000	0.000	0.000	-100	0.153692	-1.0	1.0	22	...	0	0.000	0
1	1	-2	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	19	...	0	0.000	0
2	1	90	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	15	...	0	0.000	0
3	1	178	0.000	0.000	0.000	-100	0.054845	-1.0	0.0	15	...	1	0.000	0
4	1	69	0.000	0.000	0.000	-100	0.005406	-1.0	0.5	14	...	2	0.000	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57763	0	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	10	...	0	-1.001	-1
57764	1	-2	2.000	2.000	0.000	-100	0.138289	-1.0	-100.0	2	...	0	0.000	0
57765	1	179	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	1	...	0	0.000	0
57766	1	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	22	...	0	-1.001	-1
57767	1	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	20	...	4	-1.001	-1

57742 rows × 468 columns

In [11]:
```
y_train = raw_data['label']
```

In [12]:
```
X_train = raw_data
```

In [13]:
```
X_train.drop(['label'], axis=1, inplace=True)
X_train.drop(['memo_polish'], axis=1, inplace=True)
```

In [14]:
```
X_train
```

Out[14]:

id	x0	x1	x3	x4	x5	x6	x7	x8	x9	x10	...	x470	x471	x472
0	0	55	0.000	0.000	0.000	-100	0.153692	-1.0	1.0	22	...	0	0	0
1	1	-2	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	19	...	4	0	0
2	1	90	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	15	...	0	0	0
3	1	178	0.000	0.000	0.000	-100	0.054845	-1.0	0.0	15	...	2	0	1
4	1	69	0.000	0.000	0.000	-100	0.005406	-1.0	0.5	14	...	6	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
57763	0	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	10	...	2	-1	0
57764	1	-2	2.000	2.000	0.000	-100	0.138289	-1.0	-100.0	2	...	5	0	0
57765	1	179	0.000	0.000	0.000	-100	0.000000	-1.0	-100.0	1	...	4	0	0
57766	1	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	22	...	3	-1	0
57767	1	-2	-1.001	-1.001	-1.001	-100	-100.000000	-1.0	-100.0	20	...	2	-1	4

57742 rows × 466 columns

4 模型

4.1 Baseline

In [25]:
```
import sklearn
from xgboost import XGBClassifier
from xgboost import plot_importance
```

In [ ]:
```
xgb_model = XGBClassifier()
xgb_model.fit(X_train, y_train)
```

```
[15:57:56] WARNING: ../src/learner.cc:1115: Starting in XGBoost 1.3.0, th
e default evaluation metric used with the objective 'multi:softprob' was
changed from 'merror' to 'mlogloss'. Explicitly set eval_metric if you'd
like to restore the old behavior.
```

Out[17]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=
False,
              gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=
nan,
              monotone_constraints='()', n_estimators=100, n_jobs=8,
              num_parallel_tree=1, objective='multi:softprob', predictor
='auto',
              random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight
=None,
              subsample=1, tree_method='exact', validate_parameters=1,
              verbosity=None)
```

In [18]:
```
print(xgb_model.feature_importances_)
```

```
[8.42634763e-04 1.56711612e-03 9.65127081e-04 2.95761763e-03
 1.41869823e-03 7.02603115e-03 1.53518375e-03 7.47786136e-04
 2.06307182e-03 1.27250946e-03 1.11998932e-03 8.04271898e-04
 0.00000000e+00 1.17484517e-02 1.13003317e-03 1.13802403e-03
 4.96536493e-03 0.00000000e+00 4.58448939e-03 2.73112929e-03
 1.61031401e-03 1.37993682e-03 1.29933306e-03 1.42840995e-03
 0.00000000e+00 1.05588918e-03 1.82469725e-03 2.16316478e-03
 1.80153083e-03 1.06416102e-02 0.00000000e+00 4.13561153e-04
 0.00000000e+00 8.71244760e-04 1.73257058e-03 1.69216038e-03
 2.82849767e-03 1.01189862e-03 1.04415789e-03 1.54156180e-03
 1.46741467e-03 0.00000000e+00 1.13403657e-03 1.96261937e-03
 2.78198649e-03 4.14234400e-03 8.09533929e-04 1.04309747e-03
 1.18183356e-03 1.60076364e-03 9.10182658e-04 1.48866745e-03
 2.01016851e-03 0.00000000e+00 1.31317391e-03 0.00000000e+00
 1.43471966e-03 1.55319076e-03 0.00000000e+00 5.53733378e-04
 4.14082967e-03 1.67274498e-03 0.00000000e+00 1.28552190e-03
 1.52001658e-03 0.00000000e+00 1.24104717e-03 1.58436771e-03
 1.18680904e-03 1.39876141e-03 0.00000000e+00 1.29556085e-03
 1.29171892e-03 1.69218937e-03 1.59047521e-03 2.12543830e-03
```

In [20]:
```
plt.bar(range(len(xgb_model.feature_importances_)),xgb_model.feature_import
```

Out[20]:
```
<BarContainer object of 466 artists>
```

In [26]:
```
plot_importance(xgb_model)
```

Out[26]:
```
<AxesSubplot:title={'center':'Feature importance'}, xlabel='F score', yla
bel='Features'>
```

In [27]:
```
plt.show()
```

In [28]:
```
xgb_model.save_model('baseline.json')
```

In [ ]:
```

```