0. 数据集

样本数量 39901

特征：['景点','昵称','等级','时间', '评论', '评分']

使用 '评论', '评分' 分别作为 X 和 Y，其他特征去除

1. 情感划分

由于评分 0～3 基本为负面评价，而4、5基本为正面评价，因此按此划分负向情感（0）和正向情感（1）

2. 分词

采用结巴分词

3. 停用词

对于 BOW 和 tf-idf，利用停用词表去除停用词，对于 word2vec 模型，没有去除停用词

4. 向量化

4.1 BOW 和 tf-idf

由于这两类向量化后特征过多（转换后将近4万维），因此尝试采用 LSA 潜在语义分析（TruncatedSVD）进行降维

4.1.1. tf-idf 实验

使用朴素贝叶斯模型进行二分类

降维前：

```
X_train shape: (31920, 39716)
X_test shape: (7981, 39716)
Y_train shape: (31920,)
Y_test shape: (7981,)
```

```
classification_report on test set:
              precision    recall  f1-score   support

           0       0.63      0.92      0.75      4063
           1       0.84      0.43      0.57      3918

    accuracy                           0.68      7981
   macro avg       0.73      0.68      0.66      7981
weighted avg       0.73      0.68      0.66      7981

classification_report on train set:
              precision    recall  f1-score   support

           0       0.65      0.95      0.77     16125
           1       0.91      0.48      0.63     15795

    accuracy                           0.72     31920
   macro avg       0.78      0.72      0.70     31920
weighted avg       0.78      0.72      0.70     31920
```

使用 LSA 降维至 200 时，解释方差占比 22%，模型拟合结果如下：

```
classification_report on test set:
              precision    recall  f1-score   support

           0       0.61      0.62      0.61      4063
           1       0.60      0.59      0.60      3918

    accuracy                           0.61      7981
   macro avg       0.61      0.60      0.60      7981
weighted avg       0.61      0.61      0.61      7981

classification_report on train set:
              precision    recall  f1-score   support

           0       0.61      0.64      0.62     16125
           1       0.61      0.59      0.60     15795

    accuracy                           0.61     31920
   macro avg       0.61      0.61      0.61     31920
weighted avg       0.61      0.61      0.61     31920
```

降维到 3000 时，解释方差占比 77%，但结果与上图没有太大变化

尝试不降维，直接将数据用 cat-boost 模型拟合：

```
classification_report on test set:
              precision    recall  f1-score   support

           0       0.73      0.75      0.74      4063
           1       0.73      0.72      0.72      3918

    accuracy                           0.73      7981
   macro avg       0.73      0.73      0.73      7981
weighted avg       0.73      0.73      0.73      7981
```

4.1.2 BOW 实验

使用朴素贝叶斯模型进行二分类

降维前：

```
X_train shape: (31920, 39716)
X_test shape: (7981, 39716)
Y_train shape: (31920,)
Y_test shape: (7981,)
```

```
classification_report on test set:
              precision    recall  f1-score   support

           0       0.63      0.92      0.75      4063
           1       0.84      0.43      0.57      3918

    accuracy                           0.68      7981
   macro avg       0.73      0.68      0.66      7981
weighted avg       0.73      0.68      0.66      7981

classification_report on train set:
              precision    recall  f1-score   support

           0       0.65      0.95      0.77     16125
           1       0.91      0.48      0.63     15795

    accuracy                           0.72     31920
   macro avg       0.78      0.72      0.70     31920
weighted avg       0.78      0.72      0.70     31920
```

使用 LSA 降维至 200 时，解释方差占比 38%，模型拟合结果如下：

```
classification_report on test set:
              precision    recall  f1-score   support

           0       0.61      0.60      0.61      4063
           1       0.59      0.61      0.60      3918

    accuracy                           0.60      7981
   macro avg       0.60      0.60      0.60      7981
weighted avg       0.60      0.60      0.60      7981

classification_report on train set:
              precision    recall  f1-score   support

           0       0.62      0.61      0.61     16125
           1       0.61      0.61      0.61     15795

    accuracy                           0.61     31920
   macro avg       0.61      0.61      0.61     31920
weighted avg       0.61      0.61      0.61     31920
```

可以看到，虽然一定程度上解决了维度爆炸的问题，但是模型精度下降严重

利用降维后的数据，采用十折交叉验证，比较不同模型上的效果如下：

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	CatBoost Classifier	0.667100	0.734700	0.656900	0.665000	0.660700	0.334100
1	Light Gradient Boosting Machine	0.662800	0.722400	0.649600	0.662000	0.655400	0.325400
2	Extra Trees Classifier	0.644800	0.717100	0.649600	0.637700	0.643300	0.289600
3	Gradient Boosting Classifier	0.640500	0.698100	0.604300	0.645500	0.623900	0.280300
4	Logistic Regression	0.637600	0.691500	0.585500	0.647400	0.614300	0.274300
5	Linear Discriminant Analysis	0.635600	0.685500	0.572100	0.649500	0.607800	0.270200
6	Extreme Gradient Boosting	0.635100	0.696200	0.593100	0.641300	0.616000	0.269500
7	Ridge Classifier	0.634600	0.000000	0.570300	0.648400	0.606300	0.268000
8	SVM - Linear Kernel	0.631300	0.000000	0.535100	0.656800	0.586100	0.261000
9	Quadratic Discriminant Analysis	0.620200	0.668200	0.451400	0.672300	0.539300	0.237500
10	Ada Boost Classifier	0.619000	0.664100	0.576400	0.623600	0.598700	0.237200
11	Random Forest Classifier	0.615400	0.669500	0.511200	0.638500	0.567400	0.228900
12	Decision Tree Classifier	0.589400	0.589900	0.584400	0.584700	0.584300	0.178700
13	K Neighbors Classifier	0.574600	0.599800	0.488000	0.583300	0.531100	0.147400
14	Naive Bayes	0.567600	0.595700	0.328300	0.617200	0.428200	0.130400

可以看见，树模型效果较好

4.2 word2vec

除词向量大小和训练算法，其他超参数使用默认值

```
model = gensim.models.Word2Vec(
    sentences,            # 语料
    size=size,            # 词向量大小
    sg=sg,                # 模型的训练算法: 1: skip-gram; 0: CBOW
    window=5,             # 句子中当前单词和被预测单词的最大距离
    hs=0,                 # 1: 采用hierarchical softmax训练模型; 0: 使用负采样
    negative=5,           # 使用负采样，设置多个负采样(通常在5-20之间)
    ns_exponent=0.75,     # 负采样分布指数。1.0样本值与频率成正比，0.0样本所有单词均等，负值更多地采样低频词。
    min_count=5,          # 忽略词频小于此值的单词
    alpha=0.025,          # 初始学习率
    min_alpha=0.0001,     # 随着训练的进行，学习率线性下降到min_alpha
    sample=0.001,         # 高频词随机下采样的配置阈值
    cbow_mean=1,          # 0: 使用上下文单词向量的总和; 1: 使用均值，适用于使用CBOW。
    seed=1,               # 随机种子
    workers=4             # 线程数
)
```

训练后将每个句子中的词向量求和取平均，作为算法的输入

4.2.1 Skip_Gram

在 Skip_Gram 上尝试词向量大小为 100，使用朴素贝叶斯拟合结果如下：

```
classification_report on test set:
              precision    recall  f1-score   support

           0       0.61      0.70      0.65      4063
           1       0.63      0.54      0.58      3918

    accuracy                           0.62      7981
   macro avg       0.62      0.62      0.62      7981
weighted avg       0.62      0.62      0.62      7981

classification_report on train set:
              precision    recall  f1-score   support

           0       0.61      0.69      0.65     16125
           1       0.64      0.56      0.59     15795

    accuracy                           0.62     31920
   macro avg       0.63      0.62      0.62     31920
weighted avg       0.63      0.62      0.62     31920
```

4.2.2 CBOW

在 CBOW 上尝试词向量大小为 200，使用朴素贝叶斯拟合结果如下：

```
classification_report on test set:
              precision    recall  f1-score   support

           0       0.60      0.71      0.65      4063
           1       0.63      0.52      0.57      3918

    accuracy                           0.62      7981
   macro avg       0.62      0.61      0.61      7981
weighted avg       0.62      0.62      0.61      7981

classification_report on train set:
              precision    recall  f1-score   support

           0       0.60      0.71      0.65     16125
           1       0.64      0.52      0.58     15795

    accuracy                           0.62     31920
   macro avg       0.62      0.62      0.61     31920
weighted avg       0.62      0.62      0.61     31920
```

采用十折交叉验证，比较在不同模型上的效果：

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	Extra Trees Classifier	0.688300	0.768100	0.681200	0.686100	0.683400	0.376400
1	CatBoost Classifier	0.688300	0.758400	0.697500	0.679800	0.688200	0.376600
2	Light Gradient Boosting Machine	0.685100	0.748400	0.690600	0.678000	0.683900	0.370100
3	Extreme Gradient Boosting	0.663200	0.719200	0.684100	0.651900	0.667500	0.326700
4	Ridge Classifier	0.655500	0.000000	0.684100	0.642100	0.662300	0.311400
5	Linear Discriminant Analysis	0.654400	0.699200	0.675400	0.642800	0.658500	0.309200
6	Logistic Regression	0.653500	0.697000	0.675000	0.642000	0.658000	0.307400
7	Gradient Boosting Classifier	0.653000	0.720100	0.669900	0.643300	0.656100	0.306200
8	Random Forest Classifier	0.652800	0.724700	0.561600	0.680500	0.615100	0.304200
9	Quadratic Discriminant Analysis	0.651600	0.700300	0.681500	0.638300	0.659000	0.303600
10	Ada Boost Classifier	0.632900	0.680000	0.649600	0.623600	0.636200	0.266100
11	Naive Bayes	0.624300	0.668700	0.611600	0.622600	0.616700	0.248500
12	K Neighbors Classifier	0.621700	0.664400	0.659800	0.608200	0.632800	0.244000
13	SVM - Linear Kernel	0.620200	0.000000	0.714900	0.612800	0.642300	0.242000
14	Decision Tree Classifier	0.619500	0.619200	0.618800	0.614200	0.616400	0.239000

可以看见，树模型效果较好