group: 931888
group member:
z5099411 Zexin Lan(leader)
z5127440 Yifan Zhao

COMP-9318 Final Project

Introduction:

This final project request student to devise an algorithm to fool a binary classifier . We used Python 3.6 to achieve this goal.

step 1: Data processing

Firstly , we use a dictionary to store the content in class0 and class1.
words_dict ={key = every content (in class0 and 1)}

```
5   def fool_classifier(test_data):
6       strategy_instance = helper.strategy()
7
8       words_dict = {}
9       num = 0
10
11      for line in strategy_instance.class0:
12          for content in line:
13              if content not in words_dict.keys():
14                  words_dict[content] = num
15                  num+=1
16      for line in strategy_instance.class1:
17          for content in line:
18              if content not in words_dict.keys():
19                  words_dict[content] = num
20                  num+=1
```

Then , change this dict to two list data_x and data_y .
data_x contains every words in class0 and class1.
data_y contains the times of every words in data_x.

```
22          data_x = []
23          data_y = []
24          for line in strategy_instance.class0:
25              son_x = [0 for _ in range(len(words_dict))]
26              for content in line:
27                  num = words_dict[content]
28                  son_x[num] += 1
29              data_x.append(son_x)
30              data_y.append(0)
31          for line in strategy_instance.class1:
32              son_x = [0 for _ in range(len(words_dict))]
33              for content in line:
34                  num = words_dict[content]
35                  son_x[num] += 1
36              data_x.append(son_x)
37              data_y.append(1)
```

step 2: Choose SVM kernel

In this step ,we use data from step 1 to set up a training set and then choose a Support Vector Machine kernel.

```
39          clf = strategy_instance.train_svm({'gamma':0.1,'C':0.01,'kernel':'linear',
40                                             'degree':2.0,'coef0':0.01},np.array(data_x),np.array(data_y))
```

Because in this project ,we have several eigenvalues , the linear kernel is chosen in the program.

step 3: Update modified_data

The SVM classifier has eigenvalue weights , so we remove the word with high weights in class1 from modified_data.txt .

```
46          with open(modified_data,'w') as fp:
47              for line in t_data:
48                  remain_word = []
49                  remove_word = []
50                  remove_word_weight = []
51                  for content in line:
52                      if content in words_dict.keys():
53                          if content in remove_word:
54                              continue
55                          num = words_dict[content]
56                          weight = word_weights[num]
57                          insert_index = 0
58                          for w in remove_word_weight:
59                              if w < weight:
60                                  break
61                              insert_index += 1
62                          remove_word_weight.insert(insert_index,weight)
63                          remove_word.insert(insert_index,content)
64                      else:
65                          if content not in remain_word:
66                              remain_word.append(content)
```

If the number of words is less than 20 ,we need to add the high weights words in class0.

```
67                  if len(remove_word) > 20:
68                      remove_word = remove_word[20:]
69                      for w in remove_word:
70                          fp.write(w+" ")
71                      for w in remain_word:
72                          fp.write(w + " ")
73                      fp.write("\n")
74                  else:
75                      if len(remain_word) >= 20 - len(remove_word):
76                          remain_word = remove_word[20 - len(remove_word)]
77                          for w in remain_word:
78                              fp.write(w + " ")
79                          fp.write("\n")
80                      else:
81                          count = 20 - len(remove_word) - len(remove_word)
82                          i = 0
83                          for key,value in words_dict:
84                              if key not in line:
85                                  i +=1
86                                  fp.write(key + " ")
87                                  if i == count:
88                                      break
89                          fp.write("\n")
```

Conclusion：

In this project , we made lots of mistakes , but we also got a lot knowledge from it . We have a clearly vision of SVM which I think is very important in this class and future study. In our project , there are some points to improve ,such as the parameter value and so on . In this group ,we learn a lot from each other and help each other, that make us feel study in group is a wonderful experience .

Thanks for watching!