SpamClassifier

May 22, 2018

1 Homework 5 Part I: Spam Classification in SciKit-Learn

This assignment uses data from https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

Data processing was inspired by https://www.kaggle.com/overflow012/d/uciml/sms-spam-collection-dataset/text-preprocessing-classification

Before getting started, run this to upgrade SciKit-Learn to 0.19.1. Then go to Kernel | Restart in Jupyter.

In [35]:
```
! pip install -U scikit-learn
```

```
Requirement already up-to-date: scikit-learn in /usr/local/lib/python3.5/dist-packages
You are using pip version 9.0.1, however version 9.0.3 is available.You should consider upgradin
```

In [36]:
```
import pandas as pd
```

```
####
# Helper function:
#  Return the k most frequently appearing keywords in the dataframe
def top_k(data_df, vec, k):
    X = vec.fit_transform(data_df['sms'].values)
    labels = vec.get_feature_names()

    return pd.DataFrame(columns = labels, data = X.toarray()).sum().sort_values(ascendi



sms_df = pd.read_csv('spam.csv', encoding='latin-1')
sms_df.columns = ['class', 'sms', 'a', 'b', 'c']
```

1.1 Step 1.1 Data Wrangling

Clean up sms_df. Delete 'a', 'b', 'c', lowercase the sms text

In [37]:
```
## TODO: Data wrangling / cleaning
sms_df1=sms_df.drop(sms_df.columns[[1, 2, 3, 4]], axis=1)
```

```
sms_df2=sms_df['sms'].str.lower()
sms_df2=pd.DataFrame({'sms': sms_df2})
sms_df=sms_df1.join(sms_df2)
```

1.2 Step 1.1 Results

In [38]:
```
sms_df
```

Out[38]:

	class	sms
0	ham	go until jurong point, crazy.. available only ...
1	ham	ok lar... joking wif u oni...
2	spam	free entry in 2 a wkly comp to win fa cup fina...
3	ham	u dun say so early hor... u c already then say...
4	ham	nah i don't think he goes to usf, he lives aro...
5	spam	freemsg hey there darling it's been 3 week's n...
6	ham	even my brother is not like to speak with me. ...
7	ham	as per your request 'melle melle (oru minnamin...
8	spam	winner!! as a valued network customer you have...
9	spam	had your mobile 11 months or more? u r entitle...
10	ham	i'm gonna be home soon and i don't want to tal...
11	spam	six chances to win cash! from 100 to 20,000 po...
12	spam	urgent! you have won a 1 week free membership ...
13	ham	i've been searching for the right words to tha...
14	ham	i have a date on sunday with will!!
15	spam	xxxmobilemovieclub: to use your credit, click ...
16	ham	oh k...i'm watching here:)
17	ham	eh u remember how 2 spell his name... yes i di...
18	ham	fine if thatåõs the way u feel. thatåõs the wa...
19	spam	england v macedonia - dont miss the goals/team...
20	ham	is that seriously how you spell his name?
21	ham	iĽûœm going to try for 2 months ha ha only joking
22	ham	so ì_ pay first lar... then when is da stock c...
23	ham	aft i finish my lunch then i go str down lor. ...
24	ham	fffffffff. alright no way i can meet up with ...
25	ham	just forced myself to eat a slice. i'm really ...
26	ham	lol your always so convincing.
27	ham	did you catch the bus ? are you frying an egg ...
28	ham	i'm back & we're packing the car now, i'll...
29	ham	ahhh. work. i vaguely remember that! what does...
...	...	...
5542	ham	armand says get your ass over to epsilon
5543	ham	u still havent got urself a jacket ah?
5544	ham	i'm taking derek & taylor to walmart, if i...
5545	ham	hi its in durban are you still on this number
5546	ham	ic. there are a lotta childporn cars then.
5547	spam	had your contract mobile 11 mnths? latest moto...
5548	ham	no, i was trying it all weekend ;v
5549	ham	you know, wot people wear. t shirts, jumpers, ...

```
5550   ham       cool, what time you think you can get here?
5551   ham  wen did you get so spiritual and deep. that's ...
5552   ham  have a safe trip to nigeria. wish you happines...
5553   ham                        hahaha..use your brain dear
5554   ham  well keep in mind i've only got enough gas for...
5555   ham  yeh. indians was nice. tho it did kane me off ...
5556   ham  yes i have. so that's why u texted. pshew...mi...
5557   ham  no. i meant the calculation is the same. that ...
5558   ham                             sorry, i'll call later
5559   ham  if you aren't here in the next  &lt;#&gt;  hou...
5560   ham                 anything lor. juz both of us lor.
5561   ham  get me out of this dump heap. my mom decided t...
5562   ham  ok lor... sony ericsson salesman... i ask shuh...
5563   ham                                 ard 6 like dat lor.
5564   ham  why don't you wait 'til at least wednesday to ...
5565   ham                                       huh y lei...
5566  spam  reminder from o2: to get 2.50 pounds free call...
5567  spam  this is the 2nd time we have tried 2 contact u...
5568   ham             will ì_ b going to esplanade fr home?
5569   ham  pity, * was in mood for that. so...any other s...
5570   ham  the guy did some bitching but i acted like i'd...
5571   ham                         rofl. its true to its name

[5572 rows x 2 columns]
```

```
In [39]: sms_df.groupby('class').describe()
```

```
Out[39]:         sms
               count unique                                                top freq
         class
         ham    4825   4515                           sorry, i'll call later   30
         spam    747    653  please call our customer service representativ...    4
```

1.3 Step 1.2. Vectorizing the Text

```
In [40]: ## TODO: Generate feature vectors
         import sklearn
         vec=sklearn.feature_extraction.text.CountVectorizer(decode_error = 'ignore', stop_words
         X = vec.fit_transform(sms_df['sms'].values)
```

1.4 Let's see the most frequent terms in spam

```
In [41]: top_spam = top_k(sms_df[sms_df['class'] == 'spam'], vec, 30)

         top_spam
```

```
Out[41]: free          224
         txt           163
         ur            144
```

```
mobile        127
text          125
stop          121
claim         113
reply         104
www            98
prize          93
just           78
cash           76
won            76
uk             74
150p           71
send           70
new            69
nokia          67
win            64
urgent         63
tone           60
week           60
50             57
contact        56
service        56
msg            54
com            54
18             51
16             51
guaranteed     50
dtype: int64
```

1.5 Vs ham...

In [42]:
```
top_ham = top_k(sms_df[sms_df['class'] == 'ham'], vec, 30)

top_ham
```

Out[42]:
```
gt       318
lt       316
just     293
ok       287
ll       265
ur       241
know     236
good     233
got      232
like     232
come     227
day      209
time     201
```

```
love     199
going    169
home     165
want     164
lor      162
need     158
sorry    157
don      151
da       150
today    139
later    135
dont     132
did      129
send     129
think    128
pls      123
hi       122
dtype: int64
```

1.6 Step 1.2.2 Regularize URLs and Numbers

Import *regularize* here, and use *regularize_urls* and *regularize_numbers* on the columns.

```
In [43]: # TODO: Regularize/tokenize URLs and numbers
         import regularize
         sms_df['sms']=regularize.regularize_urls(sms_df['sms'])
         sms_df['sms']=regularize.regularize_numbers(sms_df['sms'])
         sms_df
```

Out[43]:

	class	sms
0	ham	go until jurong point, crazy.. available only ...
1	ham	ok lar... joking wif u oni...
2	spam	free entry in _num_ a wkly comp to win fa cu...
3	ham	u dun say so early hor... u c already then say...
4	ham	nah i don't think he goes to usf, he lives aro...
5	spam	freemsg hey there darling it's been _num_ we...
6	ham	even my brother is not like to speak with me. ...
7	ham	as per your request 'melle melle (oru minnamin...
8	spam	winner!! as a valued network customer you have...
9	spam	had your mobile _num_ months or more? u r en...
10	ham	i'm gonna be home soon and i don't want to tal...
11	spam	six chances to win cash! from _num_ to _num...
12	spam	urgent! you have won a _num_ week free membe...
13	ham	i've been searching for the right words to tha...
14	ham	i have a date on sunday with will!!
15	spam	xxxmobilemovieclub: to use your credit, click ...
16	ham	oh k...i'm watching here:)
17	ham	eh u remember how _num_ spell his name... ye...

18	ham	fine if thatåõs the way u feel. thatåõs the wa...
19	spam	england v macedonia - dont miss the goals/team...
20	ham	is that seriously how you spell his name?
21	ham	iĽûœm going to try for _num_ months ha ha on...
22	ham	so ì_ pay first lar... then when is da stock c...
23	ham	aft i finish my lunch then i go str down lor. ...
24	ham	fffffffff. alright no way i can meet up with ...
25	ham	just forced myself to eat a slice. i'm really ...
26	ham	lol your always so convincing.
27	ham	did you catch the bus ? are you frying an egg ...
28	ham	i'm back & we're packing the car now, i'll...
29	ham	ahhh. work. i vaguely remember that! what does...
...	...	...
5542	ham	armand says get your ass over to epsilon
5543	ham	u still havent got urself a jacket ah?
5544	ham	i'm taking derek & taylor to walmart, if i...
5545	ham	hi its in durban are you still on this number
5546	ham	ic. there are a lotta childporn cars then.
5547	spam	had your contract mobile _num_ mnths? latest...
5548	ham	no, i was trying it all weekend ;v
5549	ham	you know, wot people wear. t shirts, jumpers, ...
5550	ham	cool, what time you think you can get here?
5551	ham	wen did you get so spiritual and deep. that's ...
5552	ham	have a safe trip to nigeria. wish you happines...
5553	ham	hahaha..use your brain dear
5554	ham	well keep in mind i've only got enough gas for...
5555	ham	yeh. indians was nice. tho it did kane me off ...
5556	ham	yes i have. so that's why u texted. pshew...mi...
5557	ham	no. i meant the calculation is the same. that ...
5558	ham	sorry, i'll call later
5559	ham	if you aren't here in the next <#> hou...
5560	ham	anything lor. juz both of us lor.
5561	ham	get me out of this dump heap. my mom decided t...
5562	ham	ok lor... sony ericsson salesman... i ask shuh...
5563	ham	ard _num_ like dat lor.
5564	ham	why don't you wait 'til at least wednesday to ...
5565	ham	huh y lei...
5566	spam	reminder from o _num_ : to get _num_ . _num_ ...
5567	spam	this is the _num_ nd time we have tried _num...
5568	ham	will ì_ b going to esplanade fr home?
5569	ham	pity, * was in mood for that. so...any other s...
5570	ham	the guy did some bitching but i acted like i'd...
5571	ham	rofl. its true to its name

[5572 rows x 2 columns]

1.7 Step 1.2.2 Results

Re-run the CountVectorizer, re-create vector X, and re-compute the top-30 spam terms. Output the top-30 spam terms.

```
In [44]: # TODO: Top-30 spam terms
         X = vec.fit_transform(sms_df['sms'].values)
         top_spam = top_k(sms_df[sms_df['class'] == 'spam'], vec, 30)
         #top_ham = top_k(sms_df[sms_df['class'] == 'ham'], vec, 30)
         top_spam
```

```
Out[44]: _num_         3289
         free           228
         txt            165
         ur             144
         _url_          141
         mobile         129
         stop           126
         text           125
         claim          113
         reply          104
         prize           92
         just            78
         won             76
         cash            76
         nokia           71
         send            70
         win             70
         new             69
         urgent          63
         week            60
         tone            59
         box             57
         msg             56
         service         56
         contact         56
         guaranteed      50
         ppm             49
         customer        49
         mins            47
         phone           46
         dtype: int64
```

1.8 Step 1.3 Creating Features

Take the top-30 spam + top-30 ham words, and create a new CountVectorizer, called *relevant_vec*, which *only* includes those words. See http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

In [45]:
```
# TODO: Vector of 'important' words
top_ham = top_k(sms_df[sms_df['class'] == 'ham'], vec, 30)
spam_ham=top_spam.append(top_ham)
spam_ham=spam_ham.drop_duplicates()
vecab=spam_ham.index
vecab=vecab.drop_duplicates()


relevant_vec=sklearn.feature_extraction.text.CountVectorizer(decode_error = 'ignore', s

#relevant_vec = vec.fit_transform(list_voc)
```

In [46]:
```
import sklearn.model_selection as ms
from sklearn.feature_extraction.text import TfidfTransformer
import numpy as np

# X is the feature array, based off relevant words
X = relevant_vec.fit_transform(sms_df['sms'].values).toarray()

# Compute the length of each sms message, normalized
# by max lengthXlen = np.zeros((X.shape[0],1))

inx = 0
for v in sms_df['sms'].values:
        Xlen[inx,0] = len(v)
        inx += 1
Xlen = Xlen / max(Xlen)
# Add the length as another feature
X = np.hstack((X, Xlen))

y = np.array((sms_df['class'] == 'spam').astype(int))

# Now we split...
X_train, X_test, y_train, y_test = ms.train_test_split(X,
                                                       y, test_size=0.2, random_state=42)


X_train
```

Out[46]:
```
array([[0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.09110867],
       [3.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.16684962],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.05049396],
       ...,
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.04939627],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
```

```
        0.02854007],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.03841932]])
```

1.9 Step 1.4 Classifier Evaluation

In [47]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
import sklearn.model_selection as ms
from sklearn.linear_model import LogisticRegression
import numpy as np

# Results, as a list of dictionaries
classifier_results = []
```

In [48]:
```
## Sample depth-2 decision tree
dt_model = DecisionTreeClassifier(max_depth=1,random_state=42)
dt_model.fit(X_train, y_train)
y_pred_test = dt_model.predict(X_test)
test_score = dt_model.score(X_test, y_test)
classifier_results.append({'Classifier': 'DecTree', 'Depth': 1, 'Score': test_score})


dt_model = DecisionTreeClassifier(max_depth=2,random_state=42)
dt_model.fit(X_train, y_train)
y_pred_test = dt_model.predict(X_test)
test_score = dt_model.score(X_test, y_test)
classifier_results.append({'Classifier': 'DecTree', 'Depth': 2, 'Score': test_score})


dt_model = DecisionTreeClassifier(max_depth=3,random_state=42)
dt_model.fit(X_train, y_train)
y_pred_test = dt_model.predict(X_test)
test_score = dt_model.score(X_test, y_test)
classifier_results.append({'Classifier': 'DecTree', 'Depth': 3, 'Score': test_score})

dt_model = DecisionTreeClassifier(max_depth=4,random_state=42)
dt_model.fit(X_train, y_train)
y_pred_test = dt_model.predict(X_test)
test_score = dt_model.score(X_test, y_test)
classifier_results.append({'Classifier': 'DecTree', 'Depth': 4, 'Score': test_score})

dt_model = DecisionTreeClassifier(max_depth=5,random_state=42)
dt_model.fit(X_train, y_train)
y_pred_test = dt_model.predict(X_test)
test_score = dt_model.score(X_test, y_test)
classifier_results.append({'Classifier': 'DecTree', 'Depth': 5, 'Score': test_score})
```

```
classifier = LogisticRegression(penalty='l1',random_state=42,solver='liblinear')
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'LogReg-L1','Score': test_score})

classifier = LogisticRegression(penalty='l2',random_state=42,solver='liblinear')
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'LogReg-L2', 'Score': test_score})

classifier = SVC(random_state=42)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'SVM', 'Score': test_score})



# TODO: Code for creating and testing classifiers mentioned in Step 1.4 of HW document
```

1.10 Step 1.4 Results

```
In [49]: pd.DataFrame(classifier_results)
```

Out[49]:

	Classifier	Depth	Score
0	DecTree	1.0	0.939910
1	DecTree	2.0	0.939910
2	DecTree	3.0	0.947085
3	DecTree	4.0	0.950673
4	DecTree	5.0	0.961435
5	LogReg-L1	NaN	0.970404
6	LogReg-L2	NaN	0.970404
7	SVM	NaN	0.968610

1.11 Step 2.0 Ensembles

```
In [50]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.ensemble import AdaBoostClassifier
         from sklearn.ensemble import BaggingClassifier
```

1.12 Compute ensemble classifier results here

```
In [51]: # TODO: Code for classifier construction and testing mentioned in Step 2.0 of HW docume
         classifier = RandomForestClassifier(n_estimators=31,random_state=314)
         classifier.fit(X_train, y_train)
```

```
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'RandomForest','Count': '31', 'Score': test_sc

#BaggingClassifier
dt_model = DecisionTreeClassifier(random_state=42)
classifier = BaggingClassifier(dt_model,n_estimators=31,random_state=314)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'Bag-DecTree', 'Count': '32', 'Score': test_sc

clf1 = LogisticRegression(penalty='l1',random_state=42,solver='liblinear')
classifier = BaggingClassifier(clf1,n_estimators=31,random_state=314)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'Bag-LogReg-L1', 'Count': '32', 'Score': test_


clf2 = LogisticRegression(penalty='l2',random_state=42,solver='liblinear')
classifier = BaggingClassifier(clf2,n_estimators=31,random_state=314)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'Bag-LogReg-L2', 'Count': '32', 'Score': test_

clf3 = SVC(random_state=42)
classifier = BaggingClassifier(clf3,n_estimators=31,random_state=314)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'Bag-SVM', 'Count': '32', 'Score': test_score}

# AdaBoostClassifier
dt_model = DecisionTreeClassifier(random_state=42)
classifier = AdaBoostClassifier(dt_model,n_estimators=31,random_state=314)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'Boost-DecTree','Count': '32', 'Score': test_s

clf1 = LogisticRegression(penalty='l1',random_state=42,solver='liblinear')
classifier = AdaBoostClassifier(clf1,n_estimators=31,random_state=314)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'Boost-LogReg-L1', 'Count': '32', 'Score': tes
```

```
clf2 = LogisticRegression(penalty='l2',random_state=42,solver='liblinear')
classifier = AdaBoostClassifier(clf2,n_estimators=31,random_state=314)
classifier.fit(X_train, y_train)
y_pred_test = classifier.predict(X_test)
test_score = classifier.score(X_test, y_test)
classifier_results.append({'Classifier': 'Boost-LogReg-L2', 'Count': '32', 'Score': tes

clff = SVC(random_state=42)
classifierr = AdaBoostClassifier(clff,n_estimators=31,random_state=314,algorithm='SAMME
classifierr.fit(X_train, y_train)
y_pred_test = classifierr.predict(X_test)
test_score = classifierr.score(X_test, y_test)
classifier_results.append({'Classifier': 'Boost-SVM', 'Count': '32', 'Score': test_scor
```

1.13 Step 2.0 Results

```
In [52]: pd.DataFrame(classifier_results)
```

Out[52]:

	Classifier	Count	Depth	Score
0	DecTree	NaN	1.0	0.939910
1	DecTree	NaN	2.0	0.939910
2	DecTree	NaN	3.0	0.947085
3	DecTree	NaN	4.0	0.950673
4	DecTree	NaN	5.0	0.961435
5	LogReg-L1	NaN	NaN	0.970404
6	LogReg-L2	NaN	NaN	0.970404
7	SVM	NaN	NaN	0.968610
8	RandomForest	31	NaN	0.980269
9	Bag-DecTree	32	NaN	0.975785
10	Bag-LogReg-L1	32	NaN	0.970404
11	Bag-LogReg-L2	32	NaN	0.970404
12	Bag-SVM	32	NaN	0.970404
13	Boost-DecTree	32	NaN	0.964126
14	Boost-LogReg-L1	32	NaN	0.865471
15	Boost-LogReg-L2	32	NaN	0.947982
16	Boost-SVM	32	NaN	0.865471

1.14 Step 3.0 Neural Networks

```
In [53]: from sklearn.linear_model import Perceptron
         from sklearn.neural_network import MLPClassifier

In [54]: # TODO: Code for classifier construction and testing mentioned in Step 3.0 of HW docume
         classifier = Perceptron(random_state=42)
         classifier.fit(X_train, y_train)
         y_pred_test = classifier.predict(X_test)
```

```
        test_score = classifier.score(X_test, y_test)
        classifier_results.append({'Classifier': 'Perceptron','Score': test_score})

        classifier = MLPClassifier(hidden_layer_sizes=(3,),random_state=42)
        classifier.fit(X_train, y_train)
        y_pred_test = classifier.predict(X_test)
        test_score = classifier.score(X_test, y_test)
        classifier_results.append({'Classifier': 'MLPClassifier','Hidden': '(3,)','Score': test

        classifier = MLPClassifier(hidden_layer_sizes=(10,),random_state=42)
        classifier.fit(X_train, y_train)
        y_pred_test = classifier.predict(X_test)
        test_score = classifier.score(X_test, y_test)
        classifier_results.append({'Classifier': 'MLPClassifier','Hidden': '(10,)','Score': tes


        classifier = MLPClassifier(hidden_layer_sizes=(10,10,10),random_state=42)
        classifier.fit(X_train, y_train)
        y_pred_test = classifier.predict(X_test)
        test_score = classifier.score(X_test, y_test)
        classifier_results.append({'Classifier': 'MLPClassifier','Hidden': '(10,10,10)','Score'
```

```
/usr/local/lib/python3.5/dist-packages/sklearn/linear_model/stochastic_gradient.py:128: FutureWa
  "and default tol will be 1e-3." % type(self), FutureWarning)
```

```
In [55]: pd.DataFrame(classifier_results)
```

Out[55]:

	Classifier	Count	Depth	Hidden	Score
0	DecTree	NaN	1.0	NaN	0.939910
1	DecTree	NaN	2.0	NaN	0.939910
2	DecTree	NaN	3.0	NaN	0.947085
3	DecTree	NaN	4.0	NaN	0.950673
4	DecTree	NaN	5.0	NaN	0.961435
5	LogReg-L1	NaN	NaN	NaN	0.970404
6	LogReg-L2	NaN	NaN	NaN	0.970404
7	SVM	NaN	NaN	NaN	0.968610
8	RandomForest	31	NaN	NaN	0.980269
9	Bag-DecTree	32	NaN	NaN	0.975785
10	Bag-LogReg-L1	32	NaN	NaN	0.970404
11	Bag-LogReg-L2	32	NaN	NaN	0.970404
12	Bag-SVM	32	NaN	NaN	0.970404
13	Boost-DecTree	32	NaN	NaN	0.964126
14	Boost-LogReg-L1	32	NaN	NaN	0.865471
15	Boost-LogReg-L2	32	NaN	NaN	0.947982
16	Boost-SVM	32	NaN	NaN	0.865471
17	Perceptron	NaN	NaN	NaN	0.954260
18	MLPClassifier	NaN	NaN	(3,)	0.973991

```
19    MLPClassifier   NaN    NaN       (10,)  0.971300
20    MLPClassifier   NaN    NaN  (10,10,10)  0.974888
```

1.15 Step 4.0 TensorFlow

In [56]:
```
! pip install tensorflow
import tensorflow as tf

column=list(relevant_vec.vocabulary)
column.append('length')
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.5/dist-packages
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.5/dist-packages (from tens
Requirement already satisfied: protobuf>=3.4.0 in /usr/local/lib/python3.5/dist-packages (from t
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.5/dist-packages (from ten
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.5/dist-packages (from
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.5/dist-packages (from tensc
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.5/dist-packages (from tensc
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.5/dist-packages (from ten
Requirement already satisfied: absl-py>=0.1.6 in /usr/local/lib/python3.5/dist-packages (from te
Requirement already satisfied: tensorboard<1.7.0,>=1.6.0 in /usr/local/lib/python3.5/dist-packag
Requirement already satisfied: gast>=0.2.0 in /usr/local/lib/python3.5/dist-packages (from tensc
Requirement already satisfied: setuptools in /usr/local/lib/python3.5/dist-packages (from protob
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.5/dist-packages (from t
Requirement already satisfied: bleach==1.5.0 in /usr/local/lib/python3.5/dist-packages (from ten
Requirement already satisfied: html5lib==0.9999999 in /usr/local/lib/python3.5/dist-packages (fr
Requirement already satisfied: werkzeug>=0.11.10 in /usr/local/lib/python3.5/dist-packages (from
You are using pip version 9.0.1, however version 9.0.3 is available.You should consider upgradin
```

```
/usr/local/lib/python3.5/dist-packages/h5py/__init__.py:36: FutureWarning: Conversion of the sec
  from ._conv import register_converters as _register_converters
```

In [57]:
```
# TODO: Create function input_fn(x,y)
def input_fn(x,y):
  feature_cols = {column[k]: tf.constant(x[:,k])
                      for k in range (len(column))}
  feature_cols = dict(feature_cols.items())

  label = tf.constant(y)

  return feature_cols, label

# TODO: Create function train_input_fn()

# TODO: Create function test_input_fn()

def train_input_fn():
```

```
    return input_fn(X_train,y_train)


def test_input_fn():
    return input_fn(X_test,y_test)
```

1.16 Step 4.3.1

```
In [58]: # TODO: Create DNNClassifier
         features=train_input_fn()
         features=features[0]
         feature_list=[]
         for k in features.keys():
             feature_list.append(tf.feature_column.numeric_column(k))
         classifier = tf.estimator.DNNClassifier(feature_columns=feature_list,hidden_units=[5, 5
```

```
INFO:tensorflow:Using default config.
WARNING:tensorflow:Using temporary folder as model directory: /tmp/tmp9o_ppdsy
INFO:tensorflow:Using config: {'_log_step_count_steps': 100, '_save_checkpoints_steps': None, '_
```

1.17 Step 4.3.1 Results

```
In [59]: # TODO: train
         classifier.train(input_fn=train_input_fn,steps=1000)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 1 into /tmp/tmp9o_ppdsy/model.ckpt.
INFO:tensorflow:step = 1, loss = 3128.6008
INFO:tensorflow:global_step/sec: 235.693
INFO:tensorflow:step = 101, loss = 338.79132 (0.428 sec)
INFO:tensorflow:global_step/sec: 351.612
INFO:tensorflow:step = 201, loss = 295.07962 (0.285 sec)
INFO:tensorflow:global_step/sec: 357.703
INFO:tensorflow:step = 301, loss = 280.5817 (0.279 sec)
INFO:tensorflow:global_step/sec: 364.664
INFO:tensorflow:step = 401, loss = 271.95996 (0.274 sec)
INFO:tensorflow:global_step/sec: 351.658
INFO:tensorflow:step = 501, loss = 266.29855 (0.285 sec)
INFO:tensorflow:global_step/sec: 352.016
INFO:tensorflow:step = 601, loss = 262.68198 (0.284 sec)
INFO:tensorflow:global_step/sec: 360.957
INFO:tensorflow:step = 701, loss = 259.9052 (0.278 sec)
```

```
INFO:tensorflow:global_step/sec: 361.852
INFO:tensorflow:step = 801, loss = 256.66263 (0.273 sec)
INFO:tensorflow:global_step/sec: 358.949
INFO:tensorflow:step = 901, loss = 254.2088 (0.279 sec)
INFO:tensorflow:Saving checkpoints for 1000 into /tmp/tmp9o_ppdsy/model.ckpt.
INFO:tensorflow:Loss for final step: 252.3229.
```

```
Out[59]: <tensorflow.python.estimator.canned.dnn.DNNClassifier at 0x7fde6d1f8160>
```

```
In [60]: # TODO: evaluate
         eval_result = classifier.evaluate(test_input_fn,steps=1000)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2018-04-13-01:56:27
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from /tmp/tmp9o_ppdsy/model.ckpt-1000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Evaluation [100/1000]
INFO:tensorflow:Evaluation [200/1000]
INFO:tensorflow:Evaluation [300/1000]
INFO:tensorflow:Evaluation [400/1000]
INFO:tensorflow:Evaluation [500/1000]
INFO:tensorflow:Evaluation [600/1000]
INFO:tensorflow:Evaluation [700/1000]
INFO:tensorflow:Evaluation [800/1000]
INFO:tensorflow:Evaluation [900/1000]
INFO:tensorflow:Evaluation [1000/1000]
INFO:tensorflow:Finished evaluation at 2018-04-13-01:56:36
INFO:tensorflow:Saving dict for global step 1000: accuracy = 0.9721973, accuracy_baseline = 0.86
```

```
In [61]: # TODO: results
         eval_result
         for key in sorted(eval_result):
           print('%s: %s' % (key, eval_result[key]))
```

```
accuracy: 0.9721973
accuracy_baseline: 0.8654709
auc: 0.9701451
auc_precision_recall: 0.94483364
average_loss: 0.10400946
global_step: 1000
label/mean: 0.13452914
loss: 115.97055
prediction/mean: 0.13058513
```

1.18 Step 4.3.2

```
In [62]: # TODO: Create LinearClassifier
         classifier = tf.estimator.LinearClassifier(feature_columns=feature_list,n_classes=2)
```

```
INFO:tensorflow:Using default config.
WARNING:tensorflow:Using temporary folder as model directory: /tmp/tmpt749ijz3
INFO:tensorflow:Using config: {'_log_step_count_steps': 100, '_save_checkpoints_steps': None, '_
```

1.19 Step 4.3.2 Results

```
In [63]: # TODO: train
         classifier.train(input_fn=train_input_fn,steps=1000)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 1 into /tmp/tmpt749ijz3/model.ckpt.
INFO:tensorflow:step = 1, loss = 3089.3513
INFO:tensorflow:global_step/sec: 249.741
INFO:tensorflow:step = 101, loss = 520.20966 (0.404 sec)
INFO:tensorflow:global_step/sec: 449.598
INFO:tensorflow:step = 201, loss = 429.15964 (0.220 sec)
INFO:tensorflow:global_step/sec: 465.133
INFO:tensorflow:step = 301, loss = 395.75574 (0.219 sec)
INFO:tensorflow:global_step/sec: 445.107
INFO:tensorflow:step = 401, loss = 378.98245 (0.221 sec)
INFO:tensorflow:global_step/sec: 464.133
INFO:tensorflow:step = 501, loss = 369.1893 (0.218 sec)
INFO:tensorflow:global_step/sec: 460.491
INFO:tensorflow:step = 601, loss = 362.92154 (0.217 sec)
INFO:tensorflow:global_step/sec: 462.74
INFO:tensorflow:step = 701, loss = 358.64813 (0.214 sec)
INFO:tensorflow:global_step/sec: 450.571
INFO:tensorflow:step = 801, loss = 355.59384 (0.222 sec)
INFO:tensorflow:global_step/sec: 453.776
INFO:tensorflow:step = 901, loss = 353.3285 (0.224 sec)
INFO:tensorflow:Saving checkpoints for 1000 into /tmp/tmpt749ijz3/model.ckpt.
INFO:tensorflow:Loss for final step: 351.61218.
```

```
Out[63]: <tensorflow.python.estimator.canned.linear.LinearClassifier at 0x7fde6ca1a748>
```

```
In [64]: # TODO: evaluate
         eval_result = classifier.evaluate(test_input_fn,steps=1000)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2018-04-13-01:56:46
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from /tmp/tmpt749ijz3/model.ckpt-1000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Evaluation [100/1000]
INFO:tensorflow:Evaluation [200/1000]
INFO:tensorflow:Evaluation [300/1000]
INFO:tensorflow:Evaluation [400/1000]
INFO:tensorflow:Evaluation [500/1000]
INFO:tensorflow:Evaluation [600/1000]
INFO:tensorflow:Evaluation [700/1000]
INFO:tensorflow:Evaluation [800/1000]
INFO:tensorflow:Evaluation [900/1000]
INFO:tensorflow:Evaluation [1000/1000]
INFO:tensorflow:Finished evaluation at 2018-04-13-01:56:54
INFO:tensorflow:Saving dict for global step 1000: accuracy = 0.97399104, accuracy_baseline = 0.8
```

```
In [65]: # TODO: results
         eval_result
         for key in sorted(eval_result):
           print('%s: %s' % (key, eval_result[key]))
```

```
accuracy: 0.97399104
accuracy_baseline: 0.8654709
auc: 0.9738273
auc_precision_recall: 0.93958175
average_loss: 0.10593279
global_step: 1000
label/mean: 0.13452914
loss: 118.11505
prediction/mean: 0.13300756
```

```
In [68]: print("DNN is more accuarate")
```

```
DNN is more accuarate
```