Determining the most contributing features for SVM classifier in sklearn

13

I have a dataset and I want to train my model on that data. After training, I need to know the features that are major contributors in the classification for a SVM classifier.

There is something called feature importance for forest algorithms, is there anything similar?

python machine-learning scikit-learn svm

8

edited Jan 11 '17 at 15:24
Jeru Luke
8,955 9 31 47

asked Jan 11 '17 at 13:47
Jibin Mathew
503 1 5 19

1 Have a look at these answers: stackoverflow.com/questions/11116697/... If you are using a linear SVM, the examples should work for you. – vpekar Jan 11 '17 at 19:12

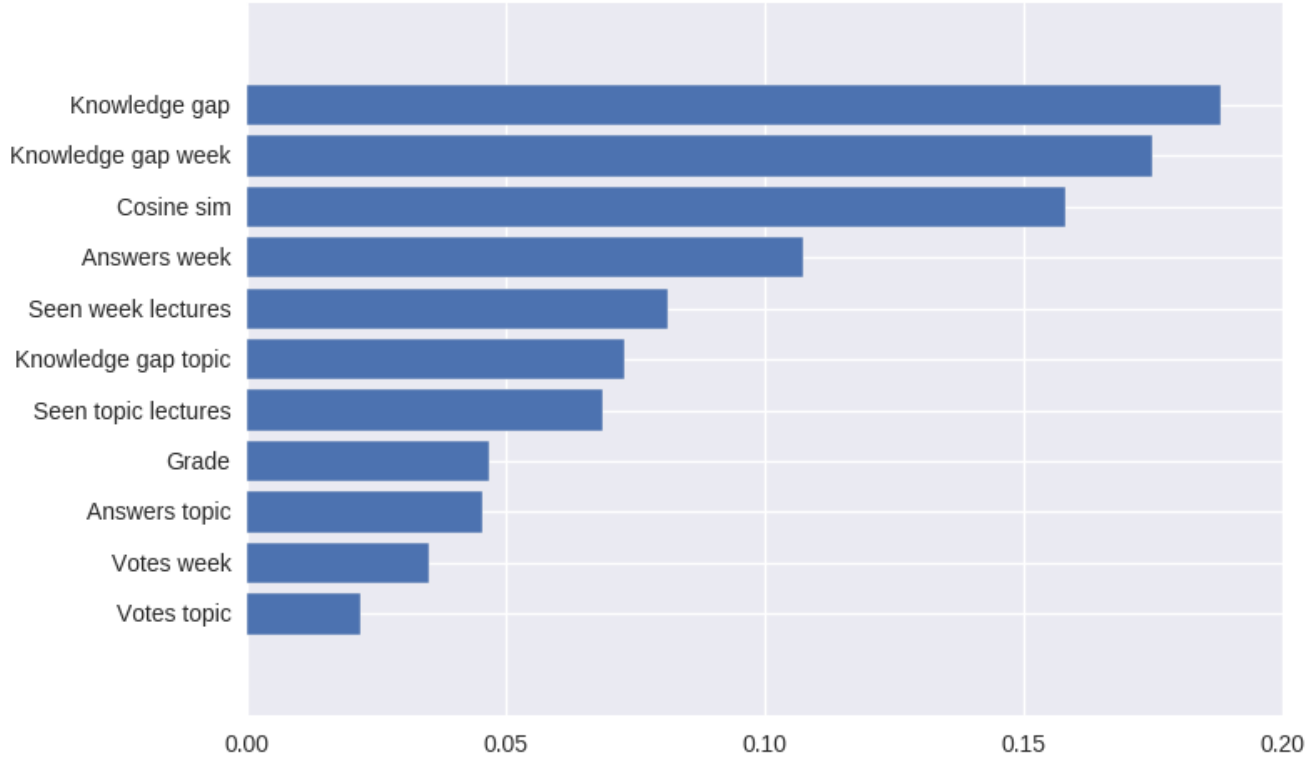
3 Answers

16

Yes, there is attribute `coef_` for SVM classifier but it only works for SVM with **linear kernel**. For other kernels it is not possible because data are transformed by kernel method to another space, which is not related to input space, check the explanation.

```
from matplotlib import pyplot as plt
from sklearn import svm

def f_importances(coef, names):
    imp = coef
    imp,names = zip(*sorted(zip(imp,names)))
    plt.barh(range(len(names)), imp, align='center')
    plt.yticks(range(len(names)), names)
    plt.show()

features_names = ['input1', 'input2']
svm = svm.SVC(kernel='linear')
svm.fit(X, Y)
f_importances(svm.coef_, features_names)
```

And the output of the function looks like this:



edited Jul 25 '17 at 11:05

Paddy
1,994 1 16 28

answered Jan 11 '17 at 21:47

Jakub Macina
482 3 13

how to find feature importance for kernal other than linear, It would be great if you could post answer for the same – Jibin Mathew Jan 13 '17 at 5:55

2 I updated the answer, it is not possible for non-linear kernel. – Jakub Macina Jan 17 '17 at 17:53

what about weights with a high negative impact? – Raphael Schumann Mar 22 '18 at 19:46

For more genereic cases and to see the effects (in same cases negative effects) you can see this [question] (stackoverflow.com/a/49937090/7127519) – Rafael Valero Apr 20 '18 at 8:34

For other classifiers there is eli5 library for example. Here example to calculate too the weight for negative effects. @raphael-schumann – Rafael Valero Apr 20 '18 at 8:40

0

In only one line of code:

fit an SVM model:

```
from sklearn import svm
svm = svm.SVC(gamma=0.001, C=100., kernel = 'linear')
```

and implement the plot as follows:

```
pd.Series(abs(svm.coef_[0]),
index=features.columns).nlargest(10).plot(kind='barh')
```

The resuit will be:

[the most contributing features of the SVM model in absolute values]

edited Mar 6 at 15:57

answered Mar 6 at 15:52

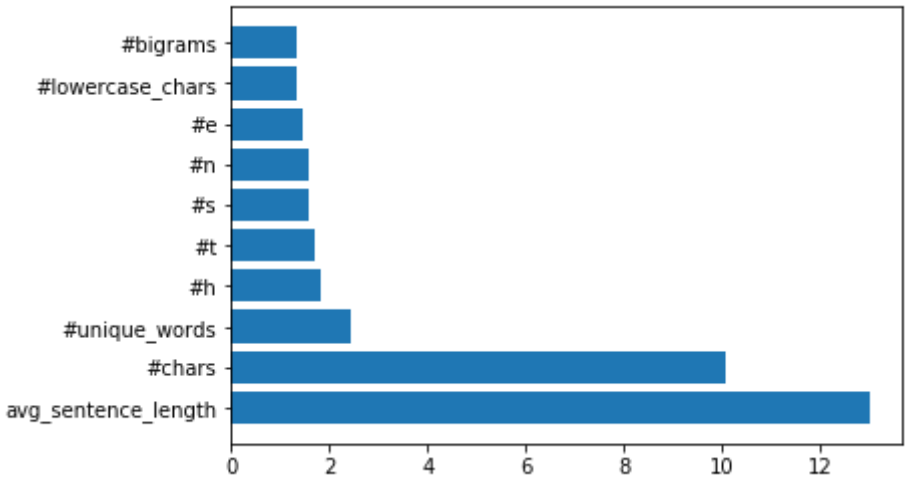
Dor

1 1

0

I created a solution which also works for Python 3 and is based on Jakub Macina's code snippet.

```
from matplotlib import pyplot as plt
from sklearn import svm

def f_importances(coef, names, top=-1):
    imp = coef
    imp, names = zip(*sorted(list(zip(imp, names))))

    # Show all features
    if top == -1:
        top = len(names)

    plt.barh(range(top), imp[::-1][0:top], align='center')
    plt.yticks(range(top), names[::-1][0:top])
    plt.show()

# whatever your features are called
features_names = ['input1', 'input2', ...]
svm = svm.SVC(kernel='linear')
svm.fit(X_train, y_train)

# Specify your top n features you want to visualize.
# You can also discard the abs() function
# if you are interested in negative contribution of features
f_importances(abs(clf.coef_[0]), feature_names, top=10)
```



edited Mar 29 at 15:07

answered Mar 29 at 14:50

Christoph Schmidl

31 3