Building Machine Learning Classifiers: Evaluate Random Forest with GridSearchCV

Grid-search: Exhaustively search all parameter combinations in a given grid to determine the best model.

Cross-validation: Divide a dataset into k subsets and repeat the holdout method k times where a different subset is used as the holdout set in each iteration.

Read in text

```
import nltk
import pandas as pd
import re
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
import string

stopwords = nltk.corpus.stopwords.words('english')
ps = nltk.PorterStemmer()

data = pd.read_csv("SMSSpamCollection.tsv", sep='\t')
data.columns = ['label', 'body_text']

def count_punct(text):
    count = sum([1 for char in text if char in string.punctuation])
    return round(count/(len(text) - text.count(" ")), 3)*100

data['body_len'] = data['body_text'].apply(lambda x: len(x) - x.count(" "))
data['punct%'] = data['body_text'].apply(lambda x: count_punct(x))

def clean_text(text):
    text = "".join([word.lower() for word in text if word not in string.punctuation])
    tokens = re.split('\W+', text)
    text = [ps.stem(word) for word in tokens if word not in stopwords]
    return text

# TF-IDF
tfidf_vect = TfidfVectorizer(analyzer=clean_text)
X_tfidf = tfidf_vect.fit_transform(data['body_text'])
X_tfidf = tfidf_vect.fit_transform(data['body_text'])
X_tfidf_feat = pd.concat([data['body_len'], data['punct%'], pd.DataFrame(X_tfidf.toarray())], axis=1)

# CountVectorizer
count_vect = CountVectorizer(analyzer=clean_text)
X_count = count_vect.fit_transform(data['body_text'])
X_count_feat = pd.concat([data['body_len'], data['punct%'], pd.DataFrame(X_count.toarray())], axis=1)

X_count_feat.head()
```

	body_len	punct%	0	1	2	3	4	5	6	7	...	8094	8095	8096	8097	8098	8099	8100	8101	8102	8103
0	128	4.7	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	49	4.1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	62	3.2	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	28	7.1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	135	4.4	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 8106 columns

Exploring parameter settings using GridSearchCV

```
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV


rf = RandomForestClassifier()
param = {'n_estimators': [10, 150, 300],
        'max_depth': [30, 60, 90, None]}

gs = GridSearchCV(rf, param, cv=5, n_jobs=-1)
gs_fit = gs.fit(X_tfidf_feat, data['label'])
pd.DataFrame(gs_fit.cv_results_).sort_values('mean_test_score', ascending=False)[0:5]
```

```
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
```

	mean_fit_time	mean_score_time	mean_test_score	mean_train_score	param_max_depth	param_n_estimators	params	rank_te
6	2.112777	0.080829	0.974852	0.997665	90	10	{'max_depth': 90, 'n_estimators': 10}	
10	17.175037	0.201542	0.974133	1.000000	None	150	{'max_depth': None, 'n_estimators': 150}	
11	26.942062	0.213621	0.973056	1.000000	None	300	{'max_depth': None, 'n_estimators': 300}	
8	31.748990	0.352917	0.972157	0.998922	90	300	{'max_depth': 90, 'n_estimators': 300}	
7	16.784482	0.227226	0.971978	0.998877	90	150	{'max_depth': 90, 'n_estimators': 150}	

```
rf = RandomForestClassifier()
param = {'n_estimators': [10, 150, 300],
        'max_depth': [30, 60, 90, None]}

gs = GridSearchCV(rf, param, cv=5, n_jobs=-1)
```

```
gs_fit = gs.fit(X_count_feat, data['label'])
pd.DataFrame(gs_fit.cv_results_).sort_values('mean_test_score', ascending=False)[0:5]
```

```
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
/Users/derekjedamski/anaconda3/lib/python3.6/site-packages/sklearn/utils/deprecation.py:122: FutureWarning: You are accessing a t
  warnings.warn(*warn_args, **warn_kwargs)
```

	mean_fit_time	mean_score_time	mean_test_score	mean_train_score	param_max_depth	param_n_estimators	params	rank_te
7	16.980228	0.238679	0.972696	0.998743	90	150	{'max_depth': 90, 'n_estimators': 150}	
8	31.826621	0.358872	0.972337	0.998743	90	300	{'max_depth': 90, 'n_estimators': 300}	
11	27.142404	0.212496	0.972337	1.000000	None	300	{'max_depth': None, 'n_estimators': 300}	
4	12.836672	0.179922	0.972157	0.993264	60	150	{'max_depth': 60, 'n_estimators': 150}	
10	17.303804	0.203654	0.971798	1.000000	None	150	{'max_depth': None, 'n_estimators': 150}	

5 rows × 22 columns