Feature Engineering

Papangkorn Inkeaw, Ph.D.

Feature Selection

Lab 10

Load dataset

Dataset: Twitter Sentiment Dataset (https://www.kaggle.com/datasets/saurabhshahane/twitter-sentiment-dataset)

The dataset has three sentiments namely, negative(-1), neutral(0), and positive(+1). It contains two fields for the tweet and label.

- Import libraries

```
import pandas as pd
import numpy as np
import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
nltk.download('punkt')
```

- Load the twitter sentiment dataset

```
data = pd.read_csv('Twitter_Data.csv’)
data.head()
```

Prepare dataset

- Convert data in clean_text column to list

```
X = data["clean_text"].values.tolist()
```

- Prepare labels

```
y = data["category"].values.tolist()
```

- Split dataset into training and test dataset

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3, random_state=42)
```

Extract BoW Features

- Extract BoW feature vectors

```
cv = CountVectorizer(X_train, stop_words =”english")
X_fv_train = cv.fit_transform(X_train).  # fit the cv using the training dataset and transform the
                                                training data
X_fv_test = cv.transform(X_test)      # apply the cv to the test dataset
```

- Scale the features with z-score standardization

```
scaler = StandardScaler( )
scaler.fit(X_fv_train)
X_fv_train = scaler.transform(X_fv_train)
X_fv_test = scaler.transform(X_fv_test)
```

Train a Random Forest

```
from sklearn.ensemble import RandomForestClassifier
```

- Train a random forest

```
rf_model = RandomForestClassifier(n_estimators=25,max_depth=5, random_state=0)
rf_model.fit(X_fv_train, y_train)
```

- Print the number of features that will be used in prediction

```
print(rf_model.n_features_in_)
```

- Predict the test data

```
y_predict = rf_model.predict(X_fv_test)
```

- Evaluate the performance

```
print(classification_report(y_test, y_predict))
```

Train a SVM with LASSO

```
from sklearn.svm import LinearSVC
```

- Train a linear SVM with L1-regularization

```
svm_model = LinearSVC(random_state=0, tol=1e-05, penalty='l1')
svm_model.fit(X_fv_train, y_train)
```

- Inspect the coefficients

```
print(svm_model.coef_)
```

- Predict the test data

```
y_predict = svm_model.predict(X_fv_test)
```

- Evaluate the performance

```
print(classification_report(y_test, y_predict))
```

Train a SVM without Feature Selection

```
from sklearn.svm import LinearSVC
```

- Train a linear SVM with L1-regularization

```
svm_wofs_model = LinearSVC(random_state=0, tol=1e-05
svm _wofs_model.fit(X_fv_train, y_train)
```

- Inspect the coefficients

```
print(svm_wofs _model.coef_)
```

- Predict the test data

```
y_predict = svm_wofs_model.predict(X_fv_test)
```

- Evaluate the performance

```
print(classification_report(y_test, y_predict))
```

Your work!

1. Load the SMS Spam Collection Dataset from https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset
2. Split the dataset into training and test dataset (Hint: use *train_test_split* method in sklearn library)
3. Extract feature vectors of samples on both training and test sets
4. Construct a classifier that embeds feature selection using the training samples for identifying the class of sms (i.e., spam or ham)
5. Evaluate performance of the classifier on the test set,
6. Submit your program to the assignment submission system (http://hw.cs.science.cmu.ac.th/).

Note:

- Put your name and student ID in the first cell using comment tag.
- Name your python notebook file with the pattern Lab_10_XXXXXXXXX.py (XXXXXXXXX is your student ID)

References & Study Resources

- https://www.kaggle.com/datasets/saurabhshahane/twitter-sentiment-dataset
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection
- https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
- https://scikit-learn.org/stable/modules/ensemble.html