Feature Engineering

Papangkorn Inkeaw, Ph.D.

Feature Combining and Expending

Lab 5

Combining Features using Decision Trees

Dataset: Boston House Prices dataset (http://lib.stat.cmu.edu/datasets/boston)

- Import libraries

```
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV
import matplotlib.pyplot as plt
```

- Load the Boston House Prices dataset from scikit-learn

```
boston_dataset = load_boston()
data = pd.DataFrame(boston_dataset.data,columns=boston_dataset.feature_names)
data['MEDV'] = boston_dataset.target
```

Combining Features using Decision Trees

- Create a dictionary with the parameter to optimize.

```
param_grid = {'max_depth': [3, 4, None]}
```

- Set up the decision tree with 5-fold cross-validation, adding the dictionary with the parameters to optimize created in the previous step, and indicating the metric we would like to optimize:

```
tree_model = GridSearchCV(DecisionTreeRegressor(random_state=0), cv = 5,
    scoring = 'neg_mean_squared_error’, param_grid = param_grid)
```

- Train the decision tree using three selected features

```
tree_model.fit(data[['LSTAT', 'RM', 'NOX']], data['MEDV'])
```

- Derive the new feature using the decision tree:

```
data['new_feat'] = tree_model.predict(data[['LSTAT', 'RM’, 'NOX’]])
```

Combining Features using Decision Trees

- Create a scatter plot with the derived decision tree feature and the target:

```
plt.scatter(X_test['new_feat'], y_test)
plt.ylabel('MEDV')
plt.xlabel('new_feat')
plt.title('Tree derived feature vs House Price')
```

Performing Polynomial Expansion

Dataset: Boston House Prices dataset (http://lib.stat.cmu.edu/datasets/boston)

- Import libraries

```
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV
import matplotlib.pyplot as plt
```

- Load the Boston House Prices dataset from scikit-learn

```
boston_dataset = load_boston()
data = pd.DataFrame(boston_dataset.data,columns=boston_dataset.feature_names)
data['MEDV'] = boston_dataset.target
```

Performing Polynomial Expansion

- Set up the polynomial expansion transformer to create features by polynomial combination of a degree 2:

  ```
  poly = PolynomialFeatures(degree=2, interaction_only=False, include_bias=False)
  ```

- Fit the transformer to the train set so that it learns all of the possible polynomial combinations of three of the variables:

  ```
  poly.fit(data[['LSTAT', 'RM', 'NOX’]])
  ```

- Examine the names of the features created:

  ```
  poly.get_feature_names(['LSTAT', 'RM', 'NOX'])
  ```

- Create the new polynomial features in a new dataset:

  ```
  data_t = poly.transform(data[['LSTAT', 'RM', 'NOX’]])
  ```

- Capture the arrays with the polynomial features in a dataframe.

  ```
  data_t = pd.DataFrame(data_t)
  data_t.columns = poly.get_feature_names(['LSTAT', 'RM', 'NOX'])
  ```

Performing Polynomial Expansion

- Create a function to make multiple subplots, each displaying one of the new polynomial features in a scatter plot versus the target:

```
def plot_features(df, target):
     nb_rows = 5
     nb_cols = 4
     fig, axs = plt.subplots(nb_rows, nb_cols, figsize=(12, 12))
     plt.subplots_adjust(wspace=None, hspace=0.4)
     n = 0
     for i in range(0, nb_rows):
              for j in range(0, nb_cols):
                       if n!=19:
                                axs[i, j].scatter(df[df.columns[n]], target)
                                axs[i, j].set_title(df.columns[n])
                                n += 1
     plt.show( )
```

- Run the function using the polynomial features derived from the new dataset:

```
plot_features(data_t, data['MEDV'])
```

Your work!

1. Load the Breast Cancer Wisconsin Data Set (source: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)) from scikit-learn library
2. Investigate the dataset details
3. Construct new features (one or more) by combining some features using a decision tree.
4. Expand some features (more than 2) using polynomial expansion with a degree 2.
5. 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_05_XXXXXXXXX.py (XXXXXXXXX is your student ID)

References & Study Resources

- Soledad Galli. (2020). *Python Feature Engineering Cookbook*. Packt Publishing.
- http://lib.stat.cmu.edu/datasets/boston
- https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)