Feature selection and extraction

1 Introduction

If you have a large data set, you can use filtering methods to perform feature selection to minimize the number of features. In this homework, you will learn to the following for feature selection:

- Variance thresholding.
- Correlation.
- Chi-square test.
- Generate predictions from the model.

You will also learn to extract features and reduce the amount of data you need in your machine learning pipeline using Principal Component Analysis (PCA).

1.1 Variance Thresholding

The variance of a feature will help determine if the feature is worth keeping. If the variance is too low, the feature does not provide much information, and thus can be removed in further analysis (Figure 1).
To calculate the variance of a data set use, the `var()` function. Here is a sample code to visualize the variance of a data set.

```
data_var = data_normalized.var()
data_var = data_var.sort_values(ascending=False)
data_var.head(10).plot(kind='bar')
```

To filter features based on variance thresholding, use the following code.

```
from sklearn.feature_selection import VarianceThreshold
v_threshold = VarianceThreshold(threshold=0.03)
high_variance_features = v_threshold.fit_transform(X)
filt = v_threshold.get_support()
names_of_selected_columns = data_normalized.columns[filt]
```

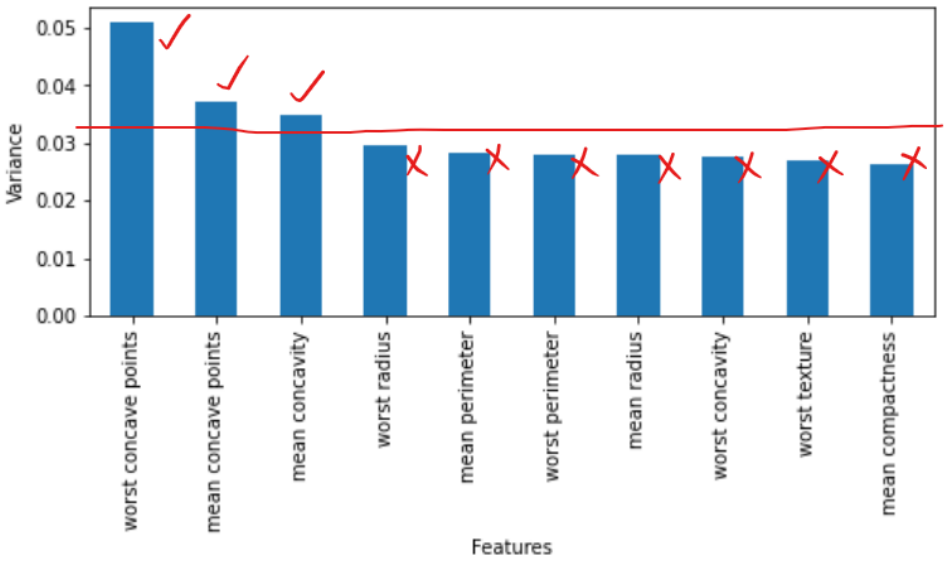

Figure 1: An example tree model to determine if a patient is a smoker depending on his blood pressure.

1.2 Chi-square test

The Chi-square test can only be done on categorical variables. Here is a sample code to select the 3 best features.

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
chi2_features = SelectKBest(chi2, k=3)
high_chi2_features = chi2_features.fit_transform(X, Y)
feature_scores = pd.DataFrame({'Score':chi2_features.scores_,
                    'P_value':chi2_features.pvalues_},
                    index=feature_names)
feature_scores.sort_values('P_value')
```

Note that chi-square test only works on categorical data. If you have numerical data, you can convert it to categorical data by using the `pd.cut()` function to discretize the numerical data into bins. Here is a sample code to do so with 10 bins.

```
data_categorical = data_normalized.apply(lambda column:
pd.cut(column,10,labels=range(10)))
```

1.3 Correlation

Identify closely correlated features, and you can retain any one of them. Also find the correlation to the target variable. The features that can influence the target variable the most should be retained. Here is sample code to find the correlation of features.

```
data_combined = pd.concat([data_normalized, pd.Series(raw_data.target,
name='target')], axis=1) # combine the features and target values
data_correlations = abs(data_combined.corr())
ordered_features = data_correlations['target'].sort_values(ascending=False)[1:]
ordered_features.plot(kind='bar') # Skip 0 since it is the target itself
plt.xlabel('Features');
plt.ylabel('Coorelation to target');
best_features = ordered_features[0:3]
print(best_features)
filt = best_features.index
print(filt)
high_corr_features = data_normalized[filt]
```

1.4 Principal Component Analysis

Performing PCA on your input data finds the directions of highest variance in your data. These direction vectors are called Principal Components (PCs). The PCs with low variances can be dropped to have a reduced dataset. Here is a sample code.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=5)
X_PCA = pca.fit_transform(data)
X_PCA = pd.DataFrame(X_PCA)
print(pca.explained_variance_)
plt.bar(range(pca.n_components_), pca.explained_variance_ratio_*100)
plt.xlabel('PCs')
plt.ylabel('Explained Variance ratio (%)')
```

2 Tasks

Complete the following task, and export your work as a pdf file. Submit the report along with the additional files (saved figures and data files) on Canvas.

1. Load the `concrete.csv` data file. The target value is strength.
2. Normalize the features between 0 and 1
3. Use variance thresholding to retain features within a threshold of 0.05.

   Which features have the highest variances?

4. Convert your features into categorical data with 10 bins.

5. Use the Chi-square test on the categorical data and find the best 3 features.

   Which features were selected? What were their p-values?

6. Compute the correlation between the features and the target value.

   Plot a heatmap of the correlation values.

   Which feature is the most correlated with the target value?

   Which feature is the most negatively correlated with the target?

   Which 5 features would you keep? Plot a bar chart to verify.

7. Use PCA to transform the data. Keep only the top 5.

   What is the Explained variance of the first PC? Plot a bar chart to verify.

   Which feature has the highest weight in the first PC? Plot a bar chart of the PC weights (eigenvector).

   Which feature has the highest weight in the second PC? Plot a bar chart of the PC weights (eigenvector).

8. Use a Decision Tree regressor to compare the performance of the original data set, and the feature selected data set. Use the features selected based on correlation to the target variable (Task 6).