Data Preprocessing Techniques

Numan Shafi

November 6, 2024

Contents

1 Introduction to Data Preprocessing

Data preprocessing is a crucial step in the data mining process, transforming raw data into a suitable format for analysis. It includes data cleaning, transformation, reduction, and feature selection to enhance data quality and model accuracy.

2 Data Cleaning Techniques

Data cleaning focuses on handling missing values, removing duplicates, and detecting outliers.

2.1 Handling Missing Values

Imputation: Filling missing values with the mean, median, or mode of the column.

```
import pandas as pd
data = pd.DataFrame({'Age': [25, None, 30, None, 35]})
data['Age'] = data['Age'].fillna(data['Age'].mean())
```

Dropping Missing Values:

```
data.dropna(inplace=True)
```

2.2 Removing Duplicates

Duplicates in the data can lead to biased analysis. Remove duplicates as follows:

```
data.drop_duplicates(inplace=True)
```

2.3 Outlier Detection and Removal

Detect outliers using the **Z-score Method**:

```
from scipy import stats
import numpy as np
data = data[(np.abs(stats.zscore(data['Age'])) < 3)]
```

3 Data Transformation Techniques

Data transformation includes scaling and encoding categorical variables.

3.1 Normalization

Normalization scales values to a range between 0 and 1.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data[['Age']] = scaler.fit_transform(data[['Age']])
```

3.2 Standardization

Standardization scales values to have a mean of 0 and a standard deviation of 1.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data[['Age']] = scaler.fit_transform(data[['Age']])
```

3.3 Encoding Categorical Variables

One-Hot Encoding converts categorical data into binary columns.

```
data = pd.get_dummies(data, columns=['Category'])
```

4 Data Reduction Techniques

Data reduction helps simplify data analysis by reducing the number of features or dimensions.

4.1 Principal Component Analysis (PCA)

PCA projects data onto a lower-dimensional space.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
reduced_data = pca.fit_transform(data)
```

4.2 Feature Aggregation

Combine multiple features by summing or averaging to reduce dimensions.

5 Feature Selection Techniques

Feature selection reduces the number of input variables by selecting the most important ones, which improves model performance and interpretability.

5.1 Filter Methods

Chi-square Test: This method is often used for categorical data to select features with the highest relevance.

```
from sklearn.feature_selection import SelectKBest, chi2
X = data.drop('target', axis=1)
y = data['target']
selector = SelectKBest(chi2, k=2)
X_selected = selector.fit_transform(X, y)
```

5.2 Wrapper Methods

Wrapper methods use a model to evaluate feature subsets. Here's an example using **Recursive Feature Elimination (RFE)** with a logistic regression model:

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
rfe = RFE(model, n_features_to_select=3)
X_selected = rfe.fit_transform(X, y)
```

5.3 Embedded Methods

Embedded methods perform feature selection during model training. For instance, **Lasso Regression** performs feature selection by shrinking some coefficients to zero.

```
from sklearn.linear_model import Lasso
model = Lasso(alpha=0.1)
model.fit(X, y)
# Select features with non-zero coefficients
selected_features = X.columns[model.coef_ != 0]
```

6 Conclusion

Data preprocessing is essential for preparing raw data for analysis. Techniques like data cleaning, transformation, reduction, and feature selection enhance data quality, boost model accuracy, and simplify analysis.