Zero-Variance Features

Instruction

Please ensure you update all the details:

Name: ULLI VENKATA SAI KUMAR

Batch Id: 04072024HYD10AM

Topic: Data Pre-Processing

Variance measures how far a set of data is spread out. A variance of zero indicates that all the data values are identical. There are various techniques to remove this for transforming the data into the suitable one for prediction.

Problem statement:

Find which columns of the given dataset with zero variance, and explore various techniques used to remove the zero variance from the dataset to perform certain analysis.

```
import pandas as pd

df = pd.read_csv(r"Z_dataset.csv")

print(df.head())

variance = df.var(numeric_only=True)

print(variance)

unique_colours = df['colour'].unique()

print(unique_colours)

from sklearn.feature_selection import VarianceThreshold

# Separating features and target variable if applicable

X = df.drop(columns=['Id'])  # Assuming 'Id' is not a feature

selector = VarianceThreshold(threshold=0.0)

# For variance thresholding, we need to convert categorical variables to numerical

# Let's use get_dummies for 'colour'

X_encoded = pd.get_dummies(X, drop_first=True)

# Fit the selector

selector.fit(X_encoded)

# Get the mask of features that pass the threshold

features_to_keep = X_encoded.columns[selector.get_support()]
```

```
# Create the reduced dataframe

X_reduced = X_encoded[features_to_keep]

print("Features before variance thresholding:", X_encoded.columns.tolist())

print("Features after variance thresholding:", features_to_keep.tolist())

# Identify columns with zero variance

zero_variance_cols = [col for col in df.columns if df[col].nunique() == 1]

print("Columns with zero variance:", zero_variance_cols)

# Drop zero variance columns

df_reduced = df.drop(columns=zero_variance_cols)

print("DataFrame after removing zero variance columns:")

print(df_reduced.head())
```

```
In [2]: df = pd.read_csv(r"Z_dataset.csv")

In [3]: print(df.head())
   Id  square.length  square.breadth  rec.Length  rec.breadth colour
0   1            5.1             3.5         1.4          0.2   Blue
1   2            4.9             3.0         1.4          0.2   Blue
2   3            4.7             3.2         1.3          0.2   Blue
3   4            4.6             3.1         1.5          0.2   Blue
4   5            5.0             3.6         1.4          0.2   Blue

In [4]: variance = df.var(numeric_only=True)
   ...: print(variance)
Id                1887.500000
square.length        0.685694
square.breadth       0.189979
rec.Length           3.116278
rec.breadth          0.581006
dtype: float64

In [5]: unique_colours = df['colour'].unique()
   ...: print(unique_colours)
['Blue' 'Green' 'Orange']
```

```
Features before variance thresholding: ['square.length', 'square.breadth', 'rec.Length', 'rec.breadth', 'colour_Green',
'colour_Orange']
Features after variance thresholding: ['square.length', 'square.breadth', 'rec.Length', 'rec.breadth', 'colour_Green',
'colour_Orange']

In [7]:
   ...: zero_variance_cols = [col for col in df.columns if df[col].nunique() == 1]
   ...: print("Columns with zero variance:", zero_variance_cols)
   ...: # Drop zero variance columns
   ...: df_reduced = df.drop(columns=zero_variance_cols)
   ...: print("DataFrame after removing zero variance columns:")
   ...: print(df_reduced.head())
Columns with zero variance: []
DataFrame after removing zero variance columns:
   Id  square.length  square.breadth  rec.Length  rec.breadth colour
0   1            5.1             3.5         1.4          0.2   Blue
1   2            4.9             3.0         1.4          0.2   Blue
2   3            4.7             3.2         1.3          0.2   Blue
3   4            4.6             3.1         1.5          0.2   Blue
4   5            5.0             3.6         1.4          0.2   Blue
```