DISCRETIZATION

Instructions:

Please share your answers filled inline in the word document. Submit Python code and R code files wherever applicable.

Please ensure you update all the details:

Name: ULLI VENKATA SAI KUMAR

Batch Id: 04072024HYD10AM

Topic: Data Pre-Processing

Problem Statement:

Everything will revolve around the data in Analytics world. Proper data will help you to make useful predictions that improve your business. Sometimes the usage of original data as it is does not help to have accurate solutions. It is needed to convert the data from one form to another form to have better predictions. Explore various techniques to transform the data for better model performance. you can go through this link:

https://360digitmg.com/mindmap-data-science

1) Convert the continuous data into discrete classes on the iris dataset.

Prepare the dataset by performing the preprocessing techniques, to have the data which improves model performance.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

```
import pandas as pd

import numpy as np

# Load the dataset

from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

df['species'] = iris.target

# Define bin edges and labels

def discretize_feature(df, feature_name, bins, labels):

    df[feature_name] = pd.cut(df[feature_name], bins=bins, labels=labels, include_lowest=True)

# Discretizing Sepal Length

df['sepal length (cm)'] = pd.cut(df['sepal length (cm)'], bins=[4, 5, 6, 7], labels=['Short', 'Medium', 'Long'])

# Discretizing Sepal Width

df['sepal width (cm)'] = pd.cut(df['sepal width (cm)'], bins=[2, 3, 4, 5], labels=['Narrow', 'Medium', 'Wide'])

# Discretizing Petal Length

df['petal length (cm)'] = pd.cut(df['petal length (cm)'], bins=[0, 2, 5, 7], labels=['Short', 'Medium', 'Long'])

# Discretizing Petal Width

df['petal width (cm)'] = pd.cut(df['petal width (cm)'], bins=[0, 1, 2, 3], labels=['Narrow', 'Medium', 'Wide'])

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

# Encode categorical features

df_encoded = pd.get_dummies(df, columns=['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)'])

# Separate features and target

X = df_encoded.drop('species', axis=1)

y = df_encoded['species']

# Normalize features

scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)
```