**Visualizing High-Dimensional Data**

**Aim:** Generate a parallel coordinates plot for visualizing high-dimensional data.

Let’s use a different dataset to create a parallel coordinates plot with Plotly. For this example, we'll use the **Wine dataset**, which is commonly used in machine learning tasks. The Wine dataset consists of 13 chemical properties of wine and a target column indicating the wine class (type). You can load this dataset from the sklearn library.

### 1. Install the required libraries

pip install plotly seaborn scikit-learn pandas

### 2. Code to generate a parallel coordinates plot for the Wine dataset

import plotly.express as px

import pandas as pd

from sklearn.datasets import load\_wine

# Load the Wine dataset

wine\_data = load\_wine()

# Convert it into a DataFrame

df = pd.DataFrame(wine\_data.data, columns=wine\_data.feature\_names)

# Add the target (wine class) to the DataFrame

df['wine\_class'] = wine\_data.target

# Create the parallel coordinates plot

fig = px.parallel\_coordinates(df,

dimensions=df.columns[:-1], # All features except the wine class

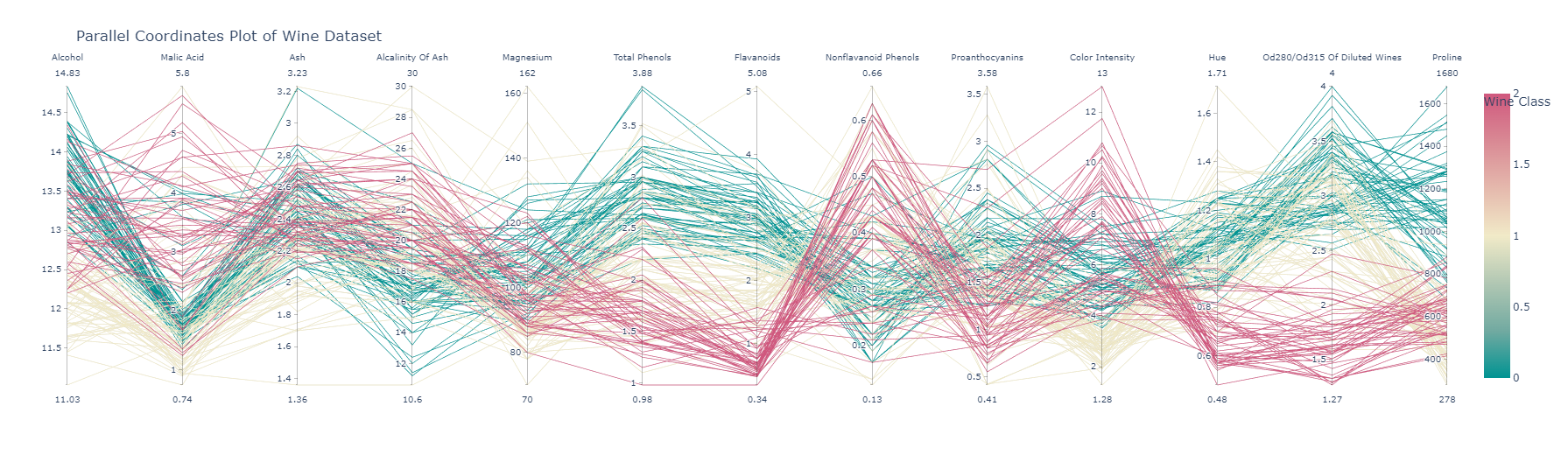
color="wine\_class", # Color by wine class

labels={col: col.replace('\_', ' ').title() for col in df.columns}, # Rename columns

color\_continuous\_scale=px.colors.diverging.Tealrose, # Color scheme

title="Parallel Coordinates Plot of Wine Dataset")

# Show the plot

fig.show()

### Detailed Explanation:

* **Loading the Wine Dataset**: We use load\_wine() from the sklearn.datasets module to load the dataset. It returns the features and target values (wine classes).
* **Converting to DataFrame**: The dataset is converted into a Pandas DataFrame for easier handling, and we assign meaningful column names (feature\_names) to the DataFrame.
* **Target Column (**wine\_class**)**: The target column (wine\_class) is added to the DataFrame. This column represents the class of the wine, which is what we’ll use to color the lines in the plot.
* **Parallel Coordinates Plot**:
  + dimensions=df.columns[:-1]: We exclude the target column (wine\_class) from the features to plot.
  + color="wine\_class": We use the wine\_class column to color the lines, so each wine class (0, 1, 2) is displayed in a different color.
  + labels: We format the column labels to replace underscores with spaces and capitalize each word.
  + color\_continuous\_scale: The Tealrose color scheme is applied to the plot for a diverging color effect.

### Wine Dataset Description:

* The Wine dataset consists of 178 samples and 13 chemical properties of wine, such as alcohol, malic acid, ash, and magnesium content. The target variable (wine\_class) has three classes representing different wine types.

### Interactive Features:

* You can hover over each line to see the corresponding values for each chemical feature.
* You can interactively filter the data by selecting ranges on any of the feature axes, helping you explore the relationships between the wine features and their classifications.

This code will generate a dynamic parallel coordinates plot to explore high-dimensional data from the Wine dataset.

**Parallel coordinates plot in Titanic Dataset**

Here I have used the **Titanic dataset** to create a parallel coordinates plot. The Titanic dataset contains various features like age, fare, sex, and whether the passenger survived or not. This can provide insights into how different features are related to survival.

### Steps:

1. **Install the required libraries** if not already installed:

pip install plotly seaborn pandas

1. **Create a parallel coordinates plot using the Titanic dataset**:

import plotly.express as px

import seaborn as sns

import pandas as pd

# Load the Titanic dataset from Seaborn

df = sns.load\_dataset("titanic")

# Drop rows with missing values to simplify the visualization

df = df.dropna(subset=['age', 'fare', 'pclass', 'survived', 'sex', 'sibsp', 'parch'])

# Convert categorical columns to numeric values for plotting

df['sex'] = df['sex'].map({'male': 0, 'female': 1})

# Select relevant columns for the parallel coordinates plot

df\_selected = df[['age', 'fare', 'pclass', 'survived', 'sex', 'sibsp', 'parch']]

# Create the parallel coordinates plot

fig = px.parallel\_coordinates(df\_selected,

dimensions=df\_selected.columns, # All selected features

color="survived", # Color by survival

labels={

'age': 'Age',

'fare': 'Fare',

'pclass': 'Passenger Class',

'survived': 'Survived',

'sex': 'Sex (0=Male, 1=Female)',

'sibsp': 'Siblings/Spouses Aboard',

'parch': 'Parents/Children Aboard'

},

color\_continuous\_scale=px.colors.diverging.Tealrose,

title="Parallel Coordinates Plot of Titanic Dataset")

# Show the plot

fig.show()

### Explanation of the Code:

1. **Loading the Dataset**:
   * We use the Titanic dataset, which is readily available through the Seaborn library.
2. **Handling Missing Values**:
   * We drop rows with missing values for simplicity (features like age have missing values). If you'd like, you can handle missing values differently, such as by imputation.
3. **Converting Categorical Columns**:
   * The sex column is categorical, so we map 'male' to 0 and 'female' to 1 for plotting purposes.
4. **Selecting Relevant Columns**:
   * We select the columns we want to visualize in the parallel coordinates plot: age, fare, pclass, survived, sex, sibsp, and parch.
5. **Parallel Coordinates Plot**:
   * dimensions=df\_selected.columns: Specifies the features we want to include in the plot.
   * color="survived": Colors the lines based on the survived column (0 for not survived, 1 for survived).
   * labels: Renames the columns for more readable labels in the plot.
   * color\_continuous\_scale: We use the Tealrose color scheme to show the color gradient from those who didn’t survive to those who did.
6. **Interactive Features**:
   * You can hover over lines to see the values for each feature.
   * You can also filter passengers interactively by selecting ranges on the vertical axes.

### Titanic Dataset Description:

* **Age**: Passenger age.
* **Fare**: The fare paid by the passenger.
* **Pclass**: The passenger class (1st, 2nd, or 3rd class).
* **Survived**: 0 = Did not survive, 1 = Survived.
* **Sex**: 0 = Male, 1 = Female (converted for the plot).
* **SibSp**: Number of siblings or spouses aboard.
* **Parch**: Number of parents or children aboard.

This plot allows you to visually explore the relationships between the features and survival outcomes of the passengers on the Titanic.