Report on Data Cleaning and Preprocessing for Adult Income Dataset

Introduction

In this report, we will discuss the data cleaning and preprocessing steps performed on the "Adult Income" dataset. The dataset contains information about individuals, including demographic and employment-related data, and is often used for predicting whether an individual earns more or less than $50,000 per year.

Data Overview

Before diving into the data preprocessing steps, let's provide an overview of the dataset:

The dataset was read from a CSV file, and the initial structure was inspected using the head() and tail() functions.

The column names were assigned to the dataset to make it more readable and interpretable.

Data Cleaning

Handling Missing Values

One of the initial data cleaning steps was to address missing values in the dataset. Missing values were represented as " ?" in the dataset. These were replaced with NaN (Not-a-Number) values using the NumPy library. The specific code used for this purpose was as follows:

python

Copy code

```
for i in df.columns:
    for j in range(len(df)):
        if df[i][j] == " ?":
            df[i][j] = np.nan
```

After this step, we checked for missing values using df.isna().sum() to confirm that there were no more missing values in the dataset.

Removing Rows with Missing Values

To maintain data quality and avoid potential issues in downstream analysis, rows containing missing values were removed from the dataset using the dropna() function. The axis=0 argument indicates that rows with missing values should be removed, and the inplace=True argument ensures that the changes are made directly to the DataFrame.

python

Copy code

```
df.dropna(axis=0, inplace=True)
```

After removing rows with missing values, the index of the DataFrame was reset for consistency using reset_index(drop=True).

Data Encoding

Label Encoding for Categorical Variables

Categorical variables in the dataset were encoded into numerical values using the LabelEncoder from the scikit-learn library. The categorical columns that underwent label encoding included:

workclass

education

marital-status

occupation

relationship

race

sex

native-country

target (the target variable)

For each categorical column, the fit_transform method was used to map the categorical values to numerical labels. The inverse_transform method can be used to convert the numerical labels back to their original categorical values if needed.

Data Description

To gain insights into the dataset's numerical features, we generated summary statistics using df.describe(). This provided basic statistics such as mean, standard deviation, minimum, and maximum values for numerical columns.

Data Export

After completing the data cleaning and preprocessing steps, the cleaned dataset was saved to a new CSV file named "cleaned_data.csv" using df.to_csv("cleaned_data.csv").

Conclusion

In this report, we have outlined the key data cleaning and preprocessing steps performed on the "Adult Income" dataset. These steps were essential to ensure that the data is ready for further analysis and machine learning tasks. The dataset is now free of missing values and has categorical variables en encoded as numerical values, making it suitable for modeling and predictive analysis.