MLOps
Assignment -2

Enhancing and Optimizing an MLOps Pipeline

भारतीय प्रौद्योगिकी संस्थान जोधपुर

Indian Institute of Technology Jodhpur

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Submitted By:

Shivani Tiwari

M24CSA029

Submitted To:

Dr. Gaurav Harit

Please visit the code for this assignment

(i) Using package
https://colab.research.google.com/drive/16qT6eiOg5ikS8OILxnyHKF6sNunkhAOA?usp=sharing

(ii) From Scratch
https://colab.research.google.com/drive/11JotNTjgra0JpKQf6OBB9zaCwoxQeFMW?usp=sharing

Objective:

The objective of the assignment is to extend and optimize the MLOps pipeline for predicting the number of bike rentals using the Bike Sharing dataset. This involves creating new interaction features between numerical variables to improve model performance, replacing OneHotEncoder with TargetEncoder for categorical variables, and comparing the performance of Linear Regression models trained using a package implementation and a custom implementation from scratch. The evaluation is done using metrics like Mean Squared Error (MSE) and R-squared to assess the impact of these changes.

Dataset : Bike Sharing Dataset

Methodology:

(i) Data Collection

Import the dataset into a Pandas DataFrame.

(ii) Data Preprocessing

- Handling Missing Values: Identify and address any missing data.
- Feature Engineering: Generate new features as needed (e.g., a day/night feature based on the hour). Created three new interaction features temp_hum (temp * hum) , temp_windspeed (temp * windspeed) , hum_windspeed (hum * windspeed)
- Normalization/Standardization: Normalize numerical features such as temp, hum, and windspeed.

For numerical features, the following steps are taken:

- **SimpleImputer:** Fill in missing values with the column's mean.
- **MinMaxScaler:** Scale the values to range between 0 and 1, which can influence regression performance.

For categorical features, the following steps are performed:

- **SimpleImputer:** Fill in missing values with the most frequent value in that column.
- **TargetEncoder:** Convert categorical values into numeric values by encoding them based on the target variable's mean, improving model training.

(iii) <u>Train - Test Split:</u>

Split the data into training and testing sets to evaluate model performance.

(iv) <u>Model Selection</u>:

- **Linear Regression (Package)**: Train a linear regression model using a machine learning library like Scikit-Learn.
- **Linear Regression (Scratch)**: Implement and train a linear regression model from scratch, following the traditional steps of linear regression (e.g., calculating coefficients via the normal equation).

(v) <u>Model Evaluation</u>:

- **Predictions**: Make predictions on the test set using both the package-based and scratch implementations.
- **Performance Metrics**: Evaluate the model performance using metrics such as Mean Squared Error (MSE) and R-squared to compare the accuracy of both approaches.

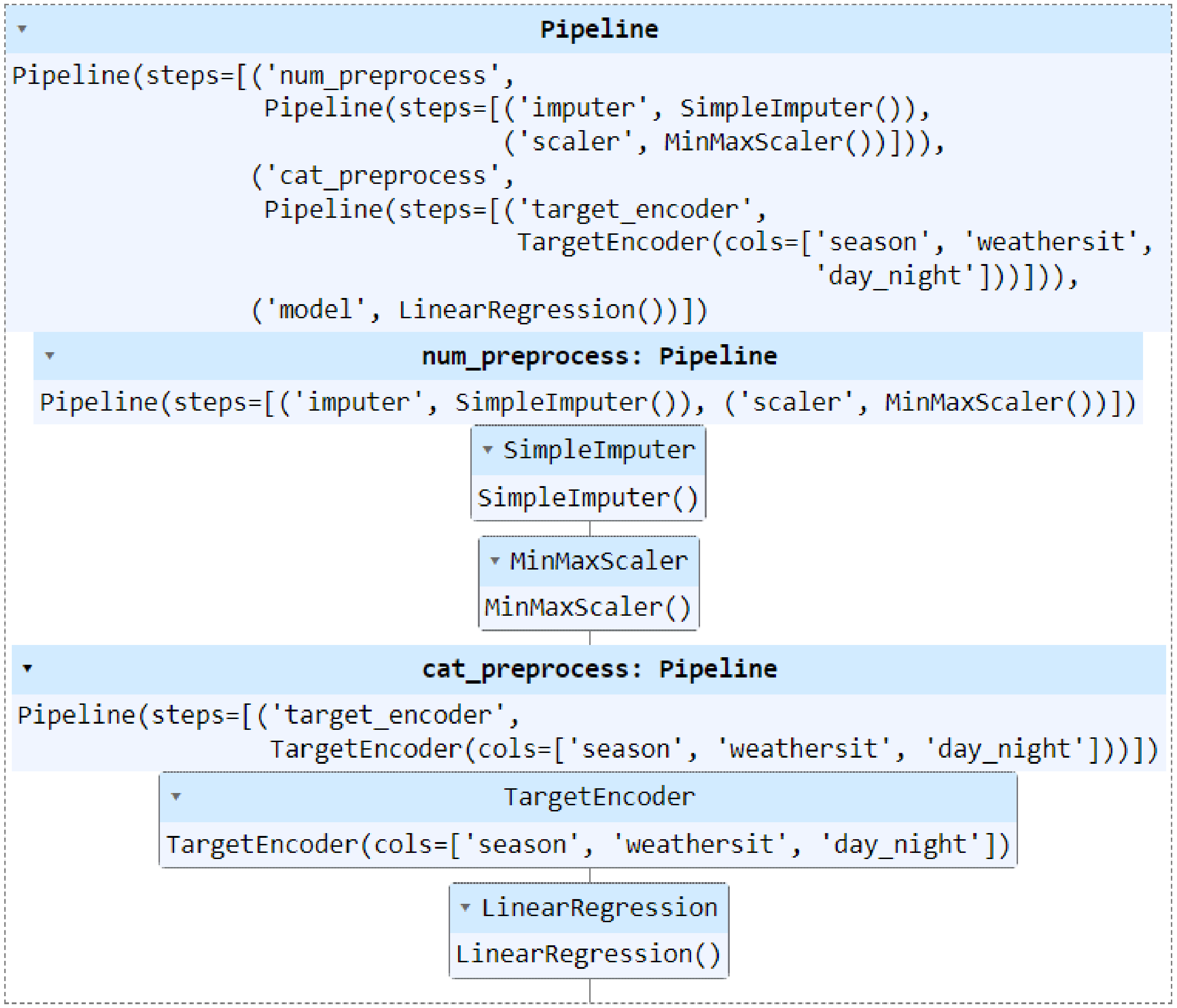
(vi) <u>Pipeline Automation:</u>

- Create an end-to-end ML pipeline that includes data preprocessing, feature engineering, model training, and prediction steps, ensuring consistency and repeatability in the process.

Results:

1. After implementing Linear Regression using a standard package, along with introducing three new interaction features between numerical variables and utilizing Target Encoding for categorical variables, the model achieved a Mean Squared Error (MSE) of 19,399.43 and an R-squared (R²) value of 0.3874.

Pipeline Output Using Package



```
Pipeline(steps=[('num_preprocess',
                 Pipeline(steps=[('imputer', SimpleImputer()),
                                 ('scaler', MinMaxScaler())])),
                ('cat_preprocess',
                 Pipeline(steps=[('target_encoder',
                                  TargetEncoder(cols=['season', 'weathersit',
                                                      'day_night']))])),
                ('model', LinearRegression())])
```

```
Pipeline(steps=[('imputer', SimpleImputer()), ('scaler', MinMaxScaler())])
```

```
SimpleImputer()
```

```
MinMaxScaler()
```

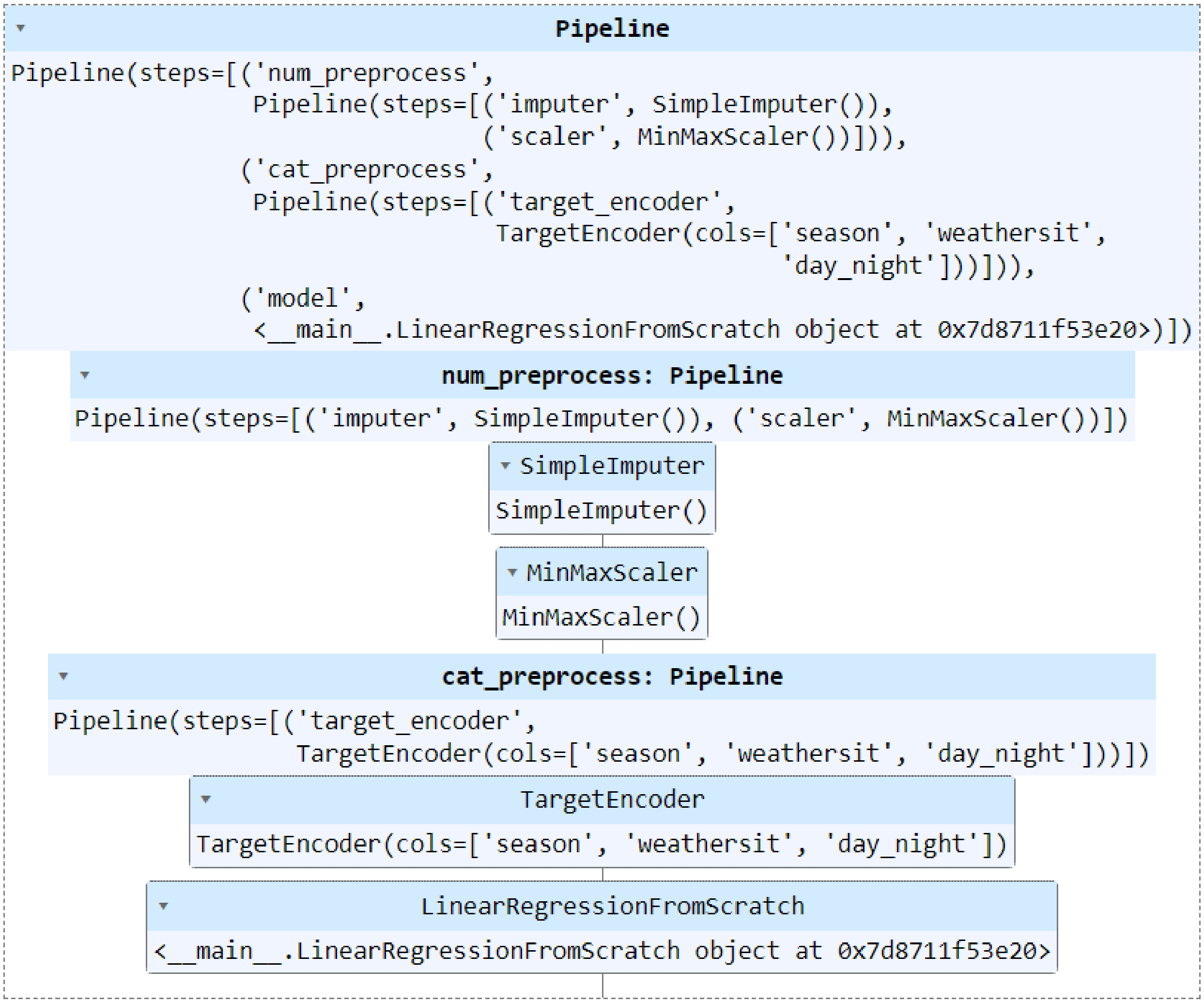
```
Pipeline(steps=[('target_encoder',
                 TargetEncoder(cols=['season', 'weathersit', 'day_night']))])
```

```
TargetEncoder(cols=['season', 'weathersit', 'day_night'])
```

```
LinearRegression()
```

2. After implementing Linear Regression from scratch, the model produced the same results, Mean Squared Error (MSE) of 19,399.33 and an R-squared (R²) value of 0.3874.

Pipeline Output without using Package



```
▾ Pipeline
Pipeline(steps=[('num_preprocess',
                 Pipeline(steps=[('imputer', SimpleImputer()),
                                 ('scaler', MinMaxScaler())])),
                ('cat_preprocess',
                 Pipeline(steps=[('target_encoder',
                                  TargetEncoder(cols=['season', 'weathersit',
                                                      'day_night']))])),
                ('model',
                 <__main__.LinearRegressionFromScratch object at 0x7d8711f53e20>)])

▾ num_preprocess: Pipeline
Pipeline(steps=[('imputer', SimpleImputer()), ('scaler', MinMaxScaler())])

▾ SimpleImputer
SimpleImputer()

▾ MinMaxScaler
MinMaxScaler()

▾ cat_preprocess: Pipeline
Pipeline(steps=[('target_encoder',
                 TargetEncoder(cols=['season', 'weathersit', 'day_night']))])

▾ TargetEncoder
TargetEncoder(cols=['season', 'weathersit', 'day_night'])

▾ LinearRegressionFromScratch
<__main__.LinearRegressionFromScratch object at 0x7d8711f53e20>
```

Observations:

(i) The interaction features temp * hum, temp * windspeed, and hum * windspeed were selected to capture potential relationships between these variables that might influence bike rental behavior.

- **temp_hum**: This feature captures the combined effect of temperature and humidity, as both are likely to impact comfort levels for outdoor activities like biking. Higher humidity at warmer temperatures could deter bike rentals, while moderate combinations might encourage them.
- **temp_windspeed**: This interaction helps account for how wind speed might affect bike rentals differently at various temperatures. For example, high wind speeds could be more tolerable in cooler temperatures but could become a deterrent when it's hot.
- **hum_windspeed:** This feature captures the interaction between humidity and wind speed, both of which affect perceived comfort. High humidity combined with high wind speed might negatively impact bike rentals, making this feature important for the model to learn complex patterns.

(ii) After replacing the OneHotEncoder with TargetEncoder for categorical variables, the model's performance decreased, as indicated by a higher Mean Squared Error (MSE) compared to using OneHotEncoder. This suggests that OneHotEncoder, which transforms categorical variables into binary dummy variables, was more effective in capturing the categorical feature interactions and their relationship with the target variable, leading to improved model performance.