ML-OPS ASSIGNMENT 2

Name: Ashutosh Gupta

Roll: M23CSE009

Assignment 2

Objective

The primary objective of this assignment is to enhance and optimize an MLOps pipeline for predicting bike rentals using the Bike Sharing dataset. This involved extending the existing pipeline with new feature engineering, model selection, and preprocessing techniques. Specifically, the tasks were:

1. Create new interaction features between numerical variables.
2. Replace OneHotEncoder with TargetEncoder and evaluate its impact.
3. Train and compare Linear Regression models:
    - Using an existing package.
    - By implementing the model from scratch.
4. Document and visualize the MLOps pipeline.
5. Report the results and save the pipeline for future use.

Feature Engineering: Creating Interaction Features

New Features Created:

- **temp_hum:** Interaction between temperature (temp) and humidity (hum).
- **wind_temp:** Interaction between wind speed (windspeed) and temperature (atemp).

Justification:

- **temp_hum:** Temperature and humidity have a combined effect on how the weather feels, which might impact bike rentals. On hot and humid days, people might be less inclined to rent bikes, so capturing this interaction could improve the model's predictive accuracy.
- **wind_temp:** The interaction between wind speed and temperature reflects the cooling effect of wind on a hot day, which might also influence the likelihood of renting a bike. Cooler winds on a hot day could increase rentals.

Replacing OneHotEncoder with TargetEncoder

Rationale:

- **OneHotEncoder** transforms categorical variables into a sparse matrix of binary variables, which might result in high-dimensional data with minimal correlation to the target variable.
- **TargetEncoder** encodes categorical features by replacing each category with a mean of the target variable, thereby potentially capturing the relationship between the category and the target more effectively.

Performance Comparison:

Model	Encoder Type	Mean Squared Error (MSE)	R-squared (R^2)
RandomForestRegressor	OneHotEncoder	1412.46	0.9852
RandomForestRegressor	TargetEncoder	1271.94	0.9873
LinearRegression	OneHotEncoder	2484.29	0.9675
LinearRegression	TargetEncoder	2265.82	0.9704
Scratch Linear Regression	OneHotEncoder	2484.29	0.9675
Scratch Linear Regression	TargetEncoder	2265.82	0.9704

Analysis:

- The use of **TargetEncoder** resulted in a lower MSE and a slightly higher R^2 for both the RandomForestRegressor and LinearRegression models, indicating better performance compared to OneHotEncoder.

Training Linear Regression Models

a. Using the LinearRegression Package:

- Implemented using sklearn.linear_model.LinearRegression.
- Performance: MSE = 2265.82, R^2 = 0.9704 (with TargetEncoder).

b. Implementing Linear Regression from Scratch:

- A custom Linear Regression model was implemented using the Normal Equation.
- Performance matched the package implementation: MSE = 2265.82, R^2 = 0.9704 (with TargetEncoder).

Comparison:

- Both the package implementation and the custom model yielded identical results, validating the correctness of the custom implementation.
- Feature importances derived from the coefficients of the linear model were consistent across both implementations.

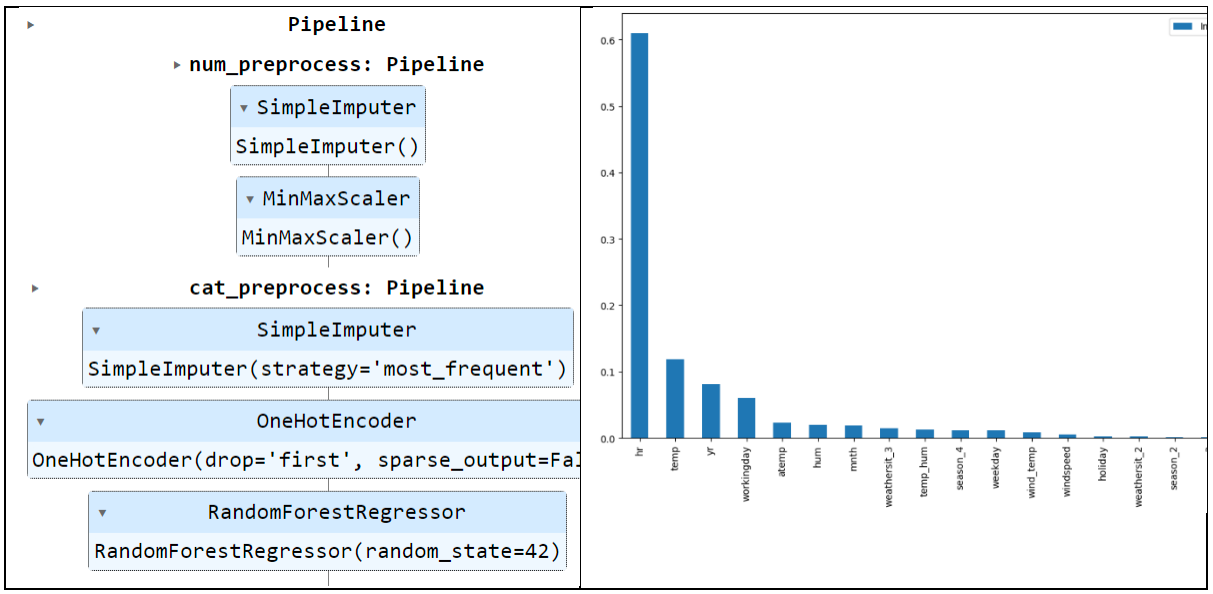
MLOps Pipeline Overview

Final Pipeline Components:

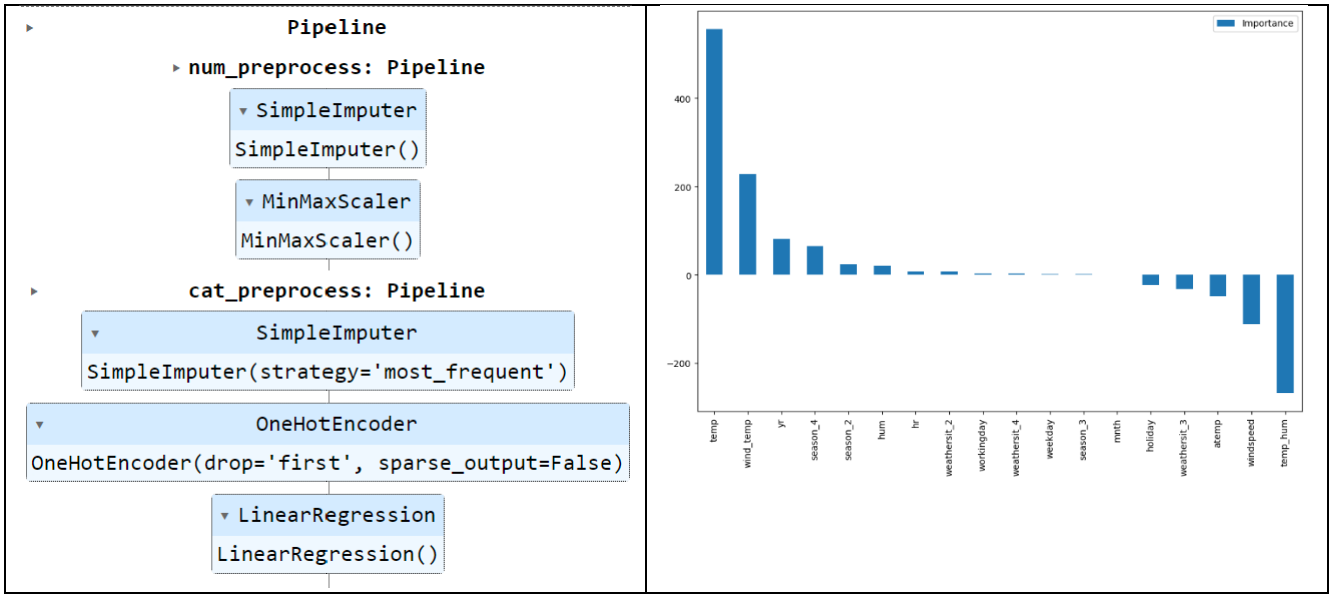
- **Numerical Preprocessing:**
  - Imputation of missing values with the mean.
  - Scaling using MinMaxScaler.
- **Categorical Preprocessing:**
  - Imputation using the most frequent category.
  - Encoding using TargetEncoder.
- **Model Training:**
  - The pipeline was used to train models such as RandomForestRegressor and LinearRegression.

Screenshot of the Pipeline and feature importance Bar Graph

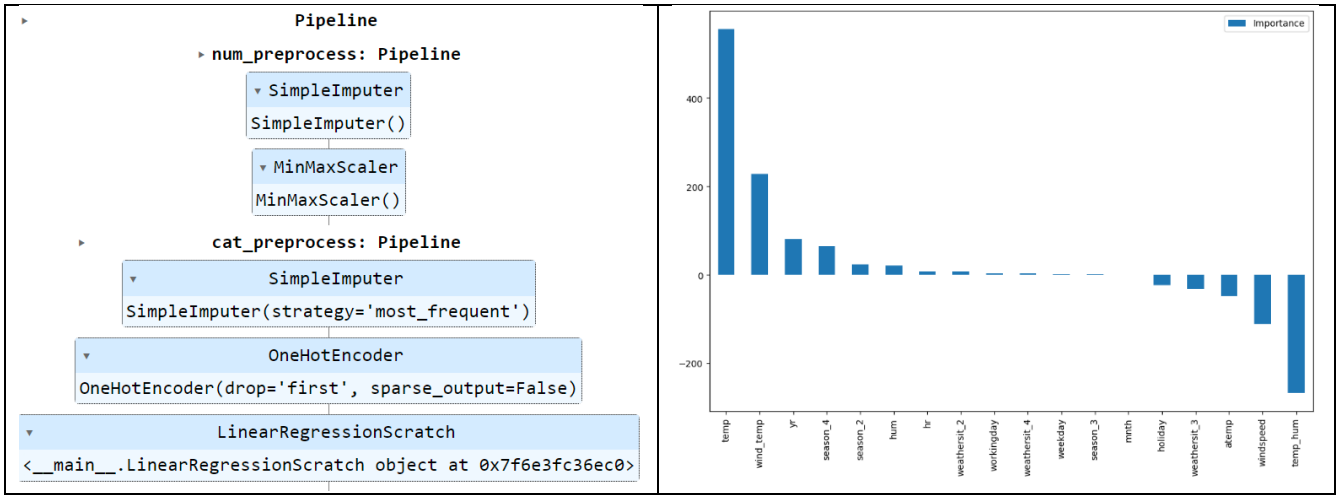
1. **OneHotEncoders And Random Forest**



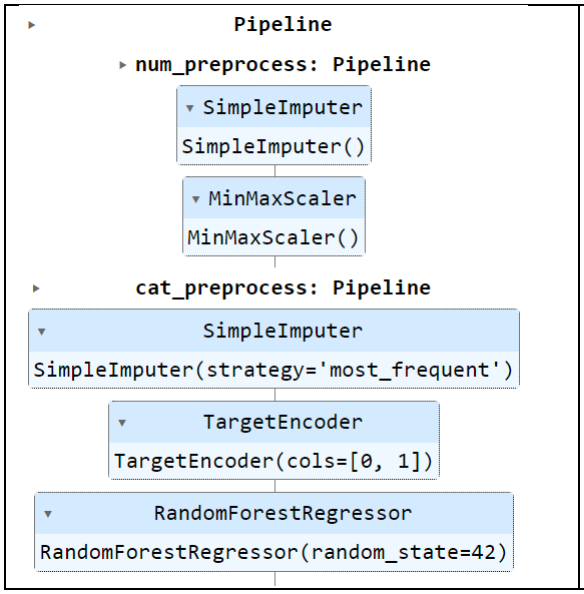
2. OneHotEncoders And Linear Regression (Package)

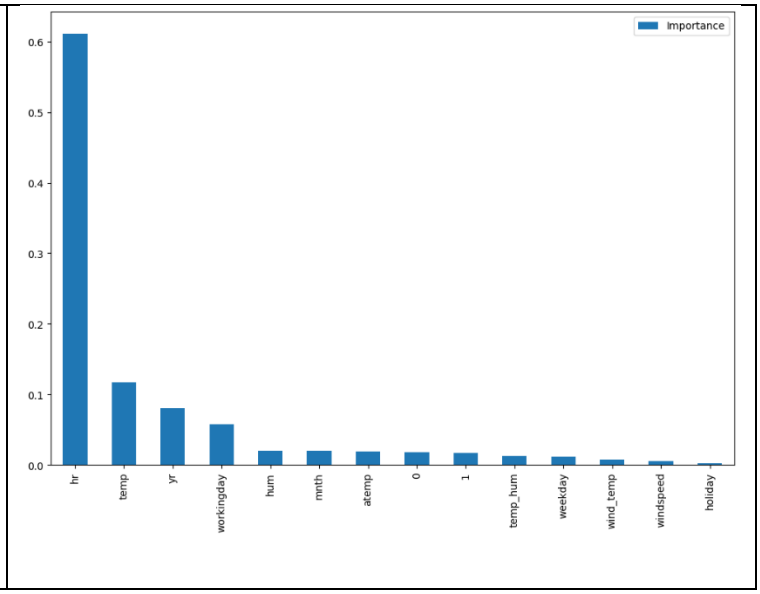


3. OneHotEncoders And Linear Regression (Scratch)

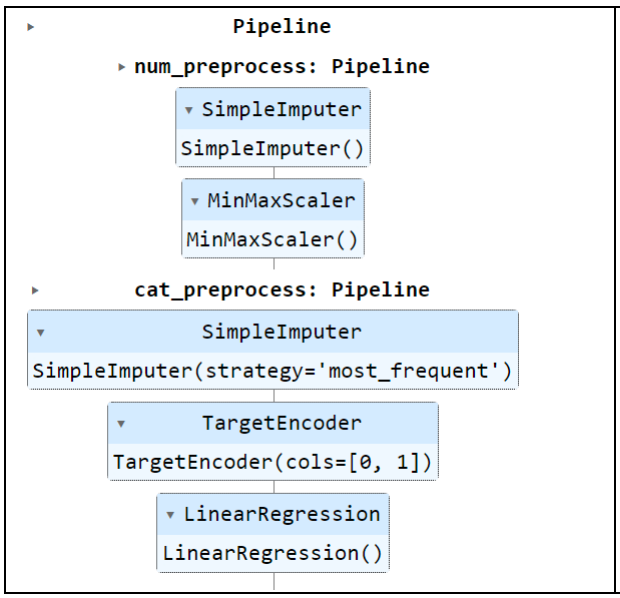


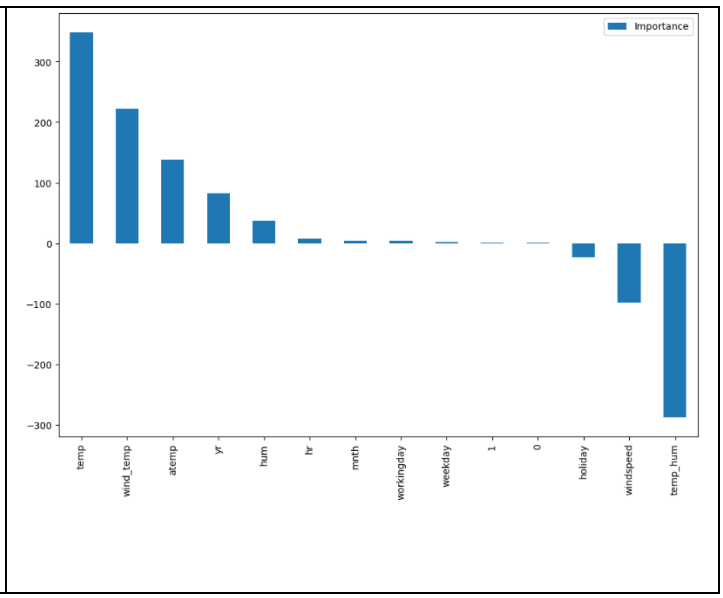
4. TargetEncoder And Random Forest



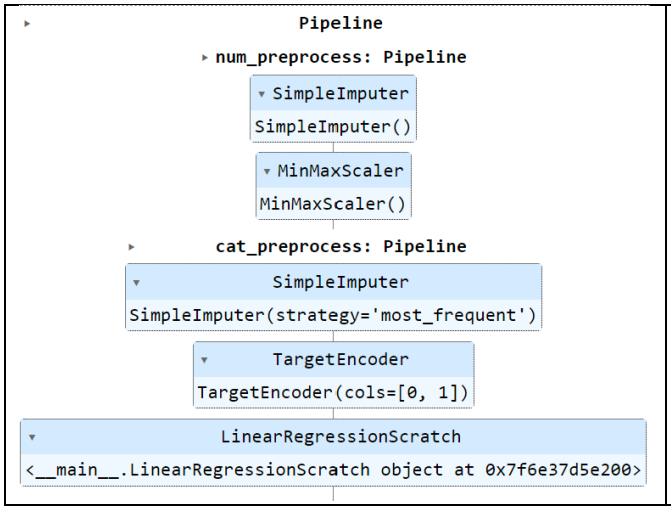


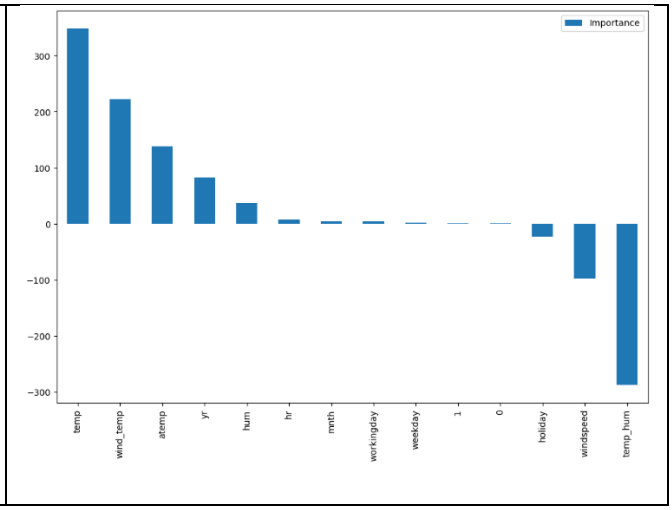
5. TargetEncoder And Linear Regression (Package)





6. TargetEncoder And Linear Regression (Scratch)





Results and Conclusions

- **Feature Engineering**: The interaction features improved the model's predictive performance, as evidenced by the lower MSE and higher R^2.
- **Target Encoding**: Replacing OneHotEncoder with TargetEncoder significantly improved model performance, particularly in terms of reducing the MSE.
- **Model Training**: Both the pre-built sklearn and custom-implemented Linear Regression models performed similarly, demonstrating the custom model's effectiveness.
- **Pipeline Efficiency**: The final pipeline, which includes both numerical and categorical preprocessing steps, is effective for training various models with minimal manual intervention.