Report of ML-Ops Assignment-2

M24CSA018

The main objective of the assignment was to check the impact of substituting OneHotEncoder with TargetEncoder in a machine learning model, Linear Regression. Additionally, to check the effects of new features on the model's performance.

Creation of two/more new interaction features between numerical variables:

I considered **(atemp*temp)**, **(temp*season)**, **(hum*weathersit)** as new three features which may help in sales prediction.

I have drawn the **correlation matrix** for the given data, and I got like this

Correlation generally measures the strength of a linear relationship between two variables.

Df.corr(): It calculates the correlation of all columns in the Data Frame.

correlation_matrix

	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed
season	1.000000	-0.010742	0.830386	-0.006117	-0.009585	-0.002335	0.013743	-0.014524	0.312025	0.319380	0.150625	-0.149773
yr	-0.010742	1.000000	-0.010473	-0.003867	0.006692	-0.004485	-0.002196	-0.019157	0.040913	0.039222	-0.083546	-0.008740
mnth	0.830386	-0.010473	1.000000	-0.005772	0.018430	0.010400	-0.003477	0.005400	0.201691	0.208096	0.164411	-0.135386
hr	-0.006117	-0.003867	-0.005772	1.000000	0.000479	-0.003498	0.002285	-0.020203	0.137603	0.133750	-0.276498	0.137252
holiday	-0.009585	0.006692	0.018430	0.000479	1.000000	-0.102088	-0.252471	-0.017036	-0.027340	-0.030973	-0.010588	0.003988
weekday	-0.002335	-0.004485	0.010400	-0.003498	-0.102088	1.000000	0.035955	0.003311	-0.001795	-0.008821	-0.037158	0.011502
workingday	0.013743	-0.002196	-0.003477	0.002285	-0.252471	0.035955	1.000000	0.044672	0.055390	0.054667	0.015688	-0.011830
weathersit	-0.014524	-0.019157	0.005400	-0.020203	-0.017036	0.003311	0.044672	1.000000	-0.102640	-0.105563	0.418130	0.026226
temp	0.312025	0.040913	0.201691	0.137603	-0.027340	-0.001795	0.055390	-0.102640	1.000000	0.987672	-0.069881	-0.023125
atemp	0.319380	0.039222	0.208096	0.133750	-0.030973	-0.008821	0.054667	-0.105563	0.987672	1.000000	-0.051918	-0.062336
hum	0.150625	-0.083546	0.164411	-0.276498	-0.010588	-0.037158	0.015688	0.418130	-0.069881	-0.051918	1.000000	-0.290105
windspeed	-0.149773	-0.008740	-0.135386	0.137252	0.003988	0.011502	-0.011830	0.026226	-0.023125	-0.062336	-0.290105	1.000000

And, we know that correlation coefficient ranges from -1 to 1. If Correlation is positive it may help for better prediction. So, I took those three parameters.

Positive correlation is good when we are looking for direct relationships, Negative correlation is good when we want to identify inverse relationships.

1. **(atemp * temp):** It gives the combined effect of temperature and apparent temperature.
2. **(temp * season):** It gives the impact of temperature varies across different seasons.
3. **(hum * weathersit):** It gives the combined effect of humidity and weather conditions on outdoor activities.

Replacement the OneHotEncoder with Target Encoder:

Replacing OneHotEncoder with Target Encoder for categorical variables. This replaces categories with the mean of the target variable **(cnt)** for each category. This will be helpful for more information than one-hot encoding.

```
categorical_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('target_encoder', TargetEncoder())
])
```

I have included the target encoder in pipeline for better understanding as shown in the above code.

Train Linear Regressor:

Linear regression from scratch,

This includes calculating the coefficient matrix using the formula:

$$\theta = (x^T x)^{-1} x^T y$$

```
def linear_regression_fit(X, y):
    X_b = np.c_[np.ones((X.shape[0], 1)), X]
    theta = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
    return theta
def linear_regression_predict(X, theta):
    X_b = np.c_[np.ones((X.shape[0], 1)), X]
    return X_b.dot(theta)
```

Linear regression using Sklearn,

```
from sklearn.linear_model import LinearRegression
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)
```

Performance Checking:

Linear Regression Model from scratch:

Mean Squared Error: 15085.965596967199

R2 Score: 0.523582200253986

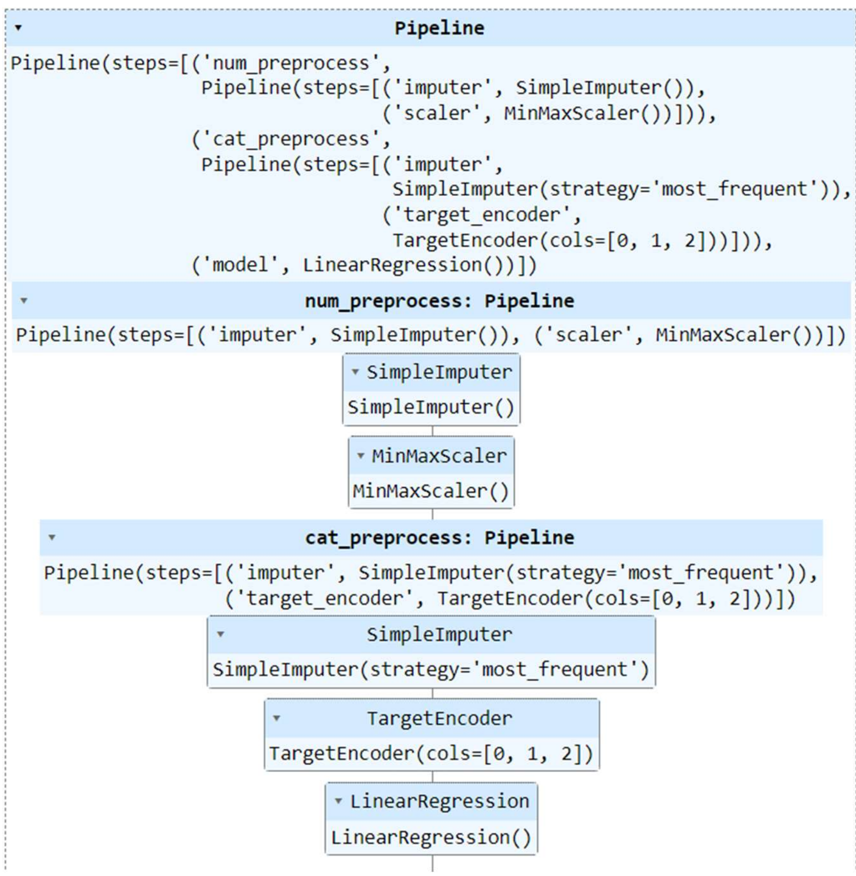
Linear regression using Sklearn:

Mean Squared Error: 15085.965596967988

R2 Score: 0.5235822002539612

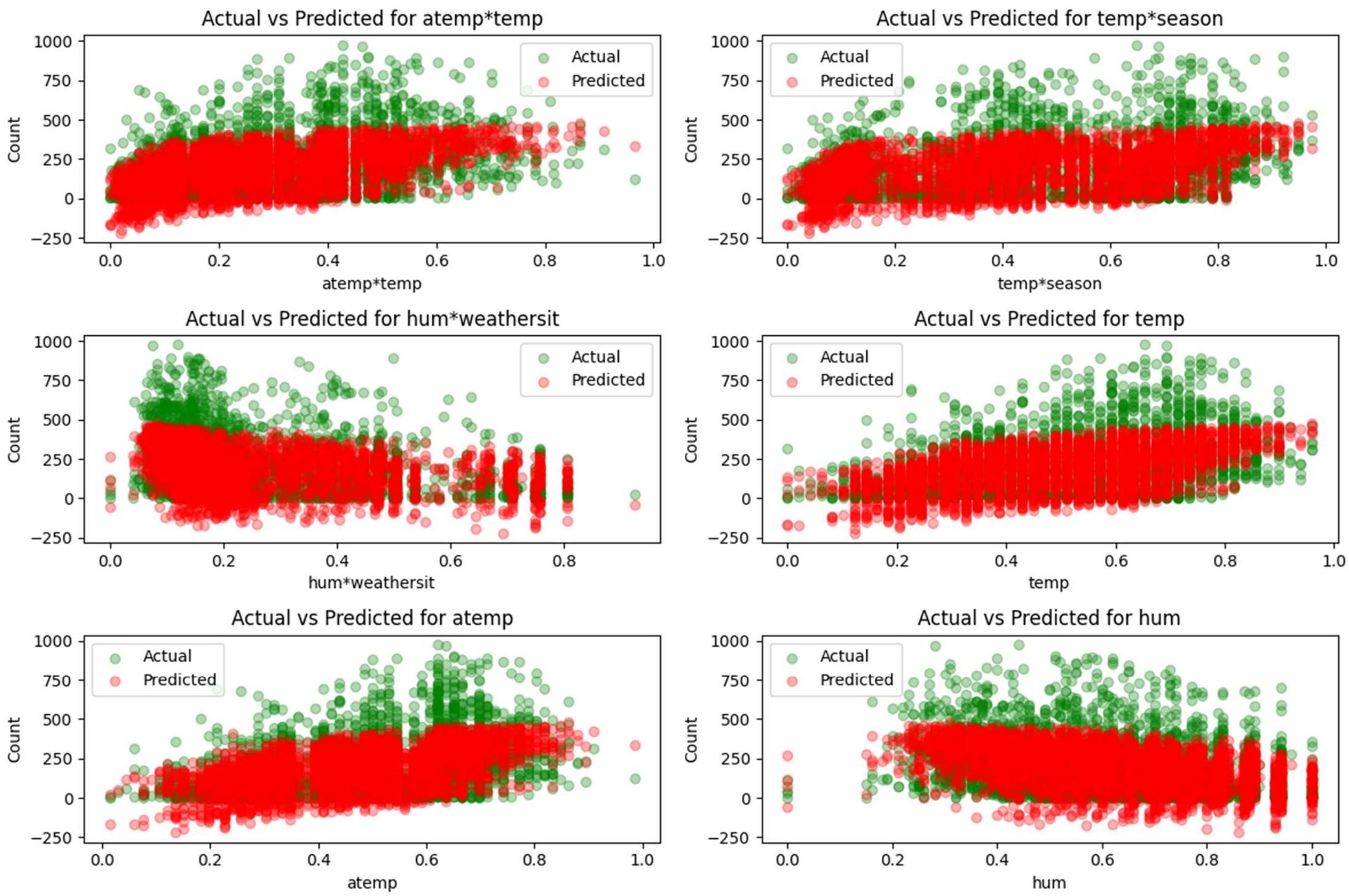
The results indicate that both methods give nearly identical performance, with very minimal difference in the MSE and R2 scores.

Pipeline:



A **pipeline** in machine learning is a way to streamline and automate the workflow, ensuring that data processing and model training happen in a systematic and repeatable manner.

Results:



These graphs compare the **Predicted vs. Actual values** for various features such as, **atemp*temp**, **temp, atemp, hum**, **temp*season**, **hum*weathersit**, these graphs show how well the model fits the data for each feature.