Machine Learning Pipeline Report

1. Introduction

This report details the machine learning pipeline implemented to analyze food delivery times. The study involves data preprocessing, feature engineering, model training, and evaluation using a structured approach with the scikit-learn library.

2. Approach

2.1 Data Collection

The dataset used is Food_delivery_Times.csv, which contains various factors affecting delivery times, such as weather conditions, traffic levels, and vehicle type.

2.2 Data Preprocessing

- **Handling Missing Values:** The dataset initially contained missing values, which were removed using dropna().
- **Feature Selection:**
  - The input features (**X**) were selected from columns 1 to 8, covering factors like weather, traffic levels, and time of day.
  - The target variable (**y**) was extracted from the last column, representing delivery time.
- **Train-Test Split:** The data was divided into training (70%) and testing (30%) sets with train_test_split().

2.3 Feature Transformation

A ColumnTransformer was implemented to apply:

- **Ordinal Encoding:** Applied to categorical features such as Weather, Traffic_Level, Time_of_Day, and Vehicle_Type.
- **Feature Scaling:** Used MinMaxScaler to normalize numerical features.

2.4 Pipe-Line creation

- **Created a pipe line after applying column transformation.**

pipe_line = make_pipeline(trf2,trf3,trf4)

2.5 Model Training

- **Linear Regression:** A basic regression model was trained to predict food delivery times.
- The model was fitted using training data and evaluated on test data.
- Used pipe_line.fit(X_train,y_train)

2.6 Model Evaluation

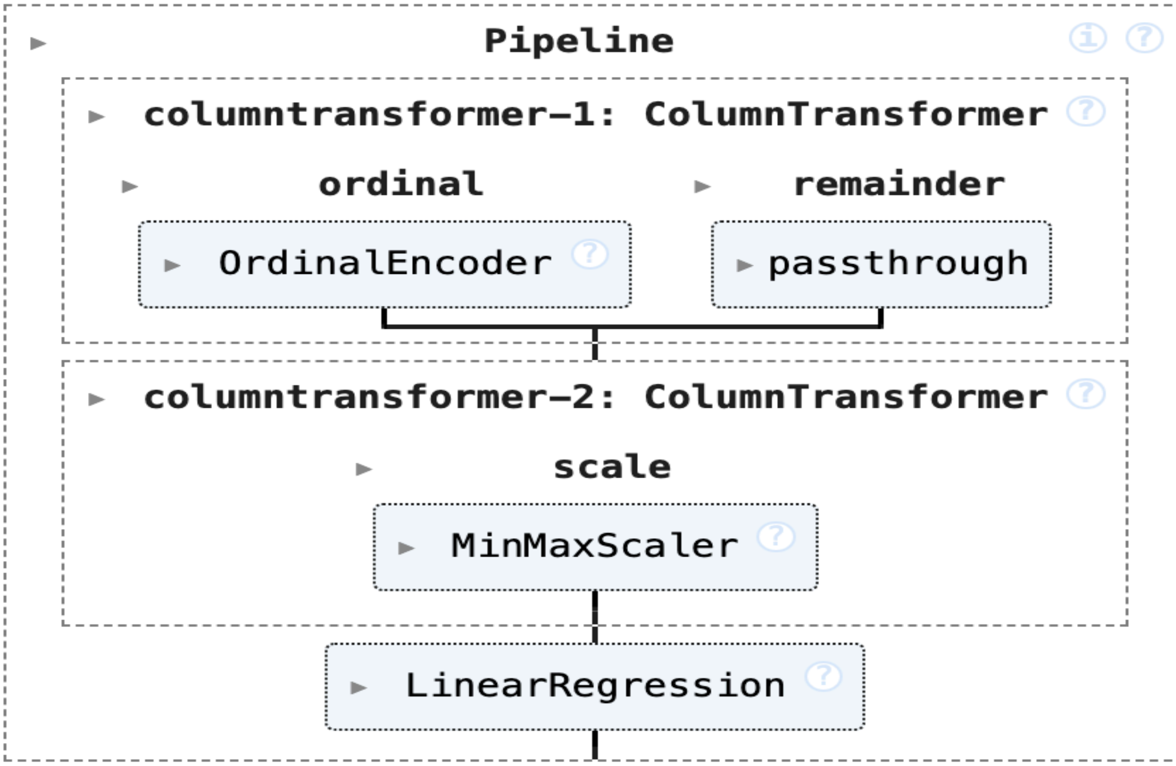
- The performance was assessed using the **R^2 score**, which measures how well the model explains variance in the target variable.
- Used pipe_line.predict(X_test)

3. Challenges

1. **Handling Missing Values:** Dropping missing values reduced the dataset size and might have caused information loss. Imputation could be explored as an alternative.
2. **Categorical Encoding:** Using **ordinal encoding** might not be ideal if categorical variables lack inherent order (e.g., weather conditions). **One-hot encoding** could be more effective.
3. **Model Choice:** Only **Linear Regression** was implemented, which might not be optimal for capturing complex relationships.
4. **Feature Engineering:** Further exploration, such as interaction terms or polynomial features, could improve model accuracy.

4. Results & Findings

- The **R^2 score** provides an indication of model performance.
- The impact of traffic level, weather, and vehicle type on delivery time could be further analyzed using feature importance techniques.
- Additional hyperparameter tuning and cross-validation might improve accuracy.



Conclusion

This project successfully built a basic machine learning pipeline for predicting food delivery times. While the model provides a starting point, improvements in feature engineering, model selection, and evaluation metrics could lead to better predictive performance.