Machine Learning Model On Food Delivery Time Prediction

By – Rahul Nair

Problem Statement

1. Predict the time in minutes to deliver Swiggy orders from origin to destination

2. Project is a regression problem that has input features about:
   - Rider
   - Delivery vehicle
   - Weather conditions
   - Traffic
   - Location of restaurant
   - Location of Delivery

Stakeholders

Swiggy

Delivery Agents / Riders

Restaurants

Customers

Business Use Case

Swiggy
- Improve Delivery Efficiency
- Enhance Customer Satisfaction
- Optimize Operational Costs

Rider
- Plan pickups and drops
- Can manage multiple orders
- Avoid Risky Driving

Restaurant
- Prioritization of Orders
- Can manage staff for in house orders vs home deliveries

Customer
- Experience of on-time delivery
- No anxiety of order arrival

Data Preparation

1. Clean the data for missing values, duplicates and other inconsistencies
2. Conduct univariate analysis to identify the features which have the highest correlation with target variable

Train Baseline Model

1. Build a baseline linear regression model to check for the performance
2. Build a baseline random forest model to compare with the linear regression model

Further Models & Model Evaluation

1. Build random forest, KNN, GB and LGBM models and evaluate the same
2. Use Grid Search CV to identify the best hyperparameters and the best model
3. Inference based on the best model

Base line Models with missing data filled

1 Datapoints

Description	Count
Data	45,593
Cleaned Data	45,502
Missing values	7,438
Train Data	36,401
Test Data	9,101

2 Data Cleaning and EDA

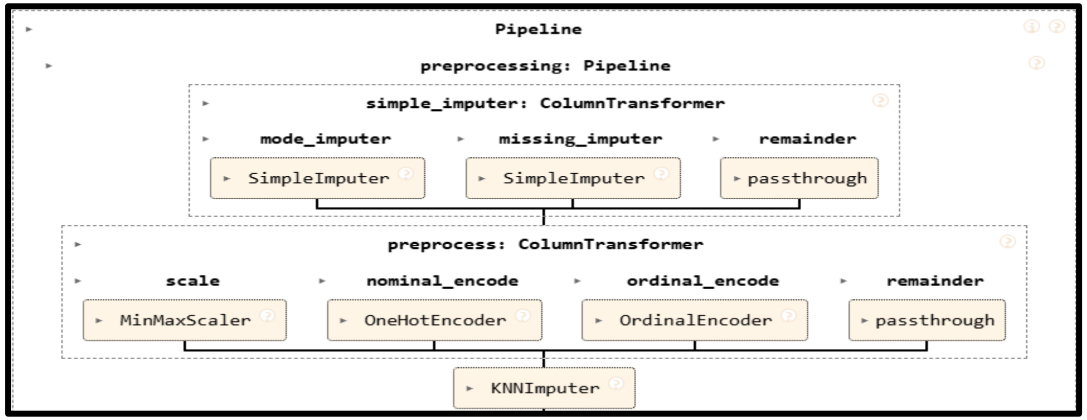
Data Cleaning

- Remove duplicate values
- Remove columns with data inconsistency

EDA

- Conduct Anova and Chi square test to identify the relationship between the variables

3 Preprocessing Steps



4 Machine Learning Models & Metrics

Models used

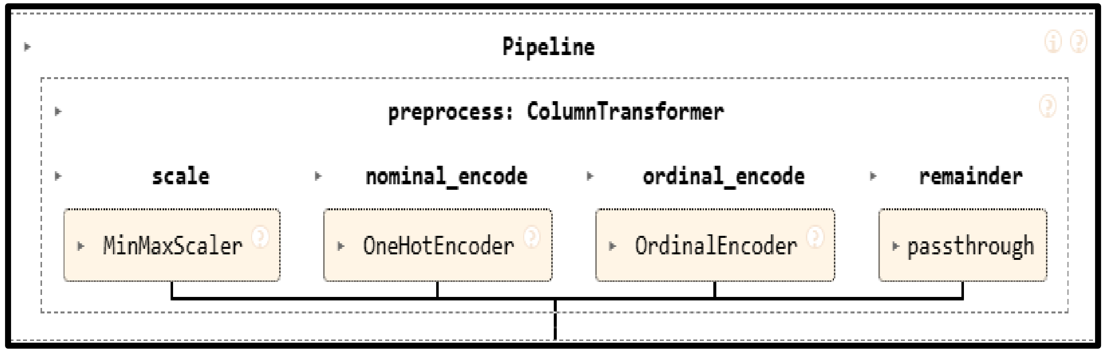
- Linear Regression
- Random Forest

Metrics used –

- Mean Absolute Error
- R2 score

Base line Models with missing data removed

3 Preprocessing Steps



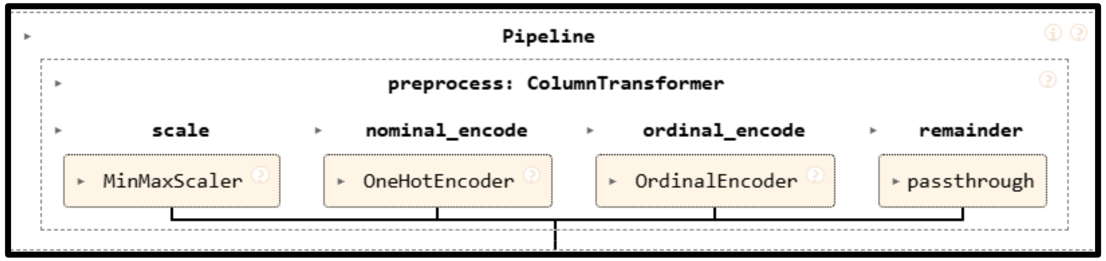
4 Machine Learning Models & Metrics

Models used
- Linear Regression
- Random Forest

Metrics used –
- Mean Absolute Error
- R2 score

Ensemble Model with missing data removed

3 Preprocessing Steps



4 Parameters of Grid Search CV

- CV=5
- Scoring – neg_mean_absolute_error

5 Models used in Grid Search CV

Model Name	Hyperparameters in Grid Search CV
Random Forest	1. n_estimators - 10, 100, 200 2. max_depth - 2, 20
XGBoost	1. n_estimators - 10, 100, 200 2. max_depth - 2, 20 3. learning_rate - 0.1, 0.5
LGBM	1. n_estimators - 10, 100, 200 2. max_depth - 2, 20 2. 3. learning_rate – 0.1, 0.5
KNN	1. n_neighbours - 1, 25 2. weights – 'uniform', 'distance'
Averaging Ensemble Model	**XGBoost – n_estimators – 100, max_depth – 20, learning rate – 0.1** **LGBM - n_estimators – 200, max_depth – 20, learning rate – 0.1**

Model Performance

	Model Name	MAE		R2 score	
		Train	Test	Train	Test
Baseline Model	Linear Regression (Missing values dropped)	4.67 min	4.73 min	0.60	0.60
	Linear Regression (Missing values filled)	4.82 min	4.85 min	0.58	0.58
	Random Forest (Missing values dropped)	1.15 min	3.13 min	0.98	0.83
	Random Forest (Missing values filled)	1.22 min	3.28 min	0.97	0.80
Grid Search CV	Random Forest	1.29 min	3.12 min	0.97	0.83
	LGBM	**2.82 min**	**3.06 min**	**0.86**	**0.84**
	XGBoost	**1.53 min**	**3.10 min**	**0.96**	**0.83**
	KNN	0 min	4.26 min	1.00	0.66
Ensemble model	**Averaging of XGBoost & LGBM**	**2.15 min**	**3.05 min**	**0.92**	**0.84**

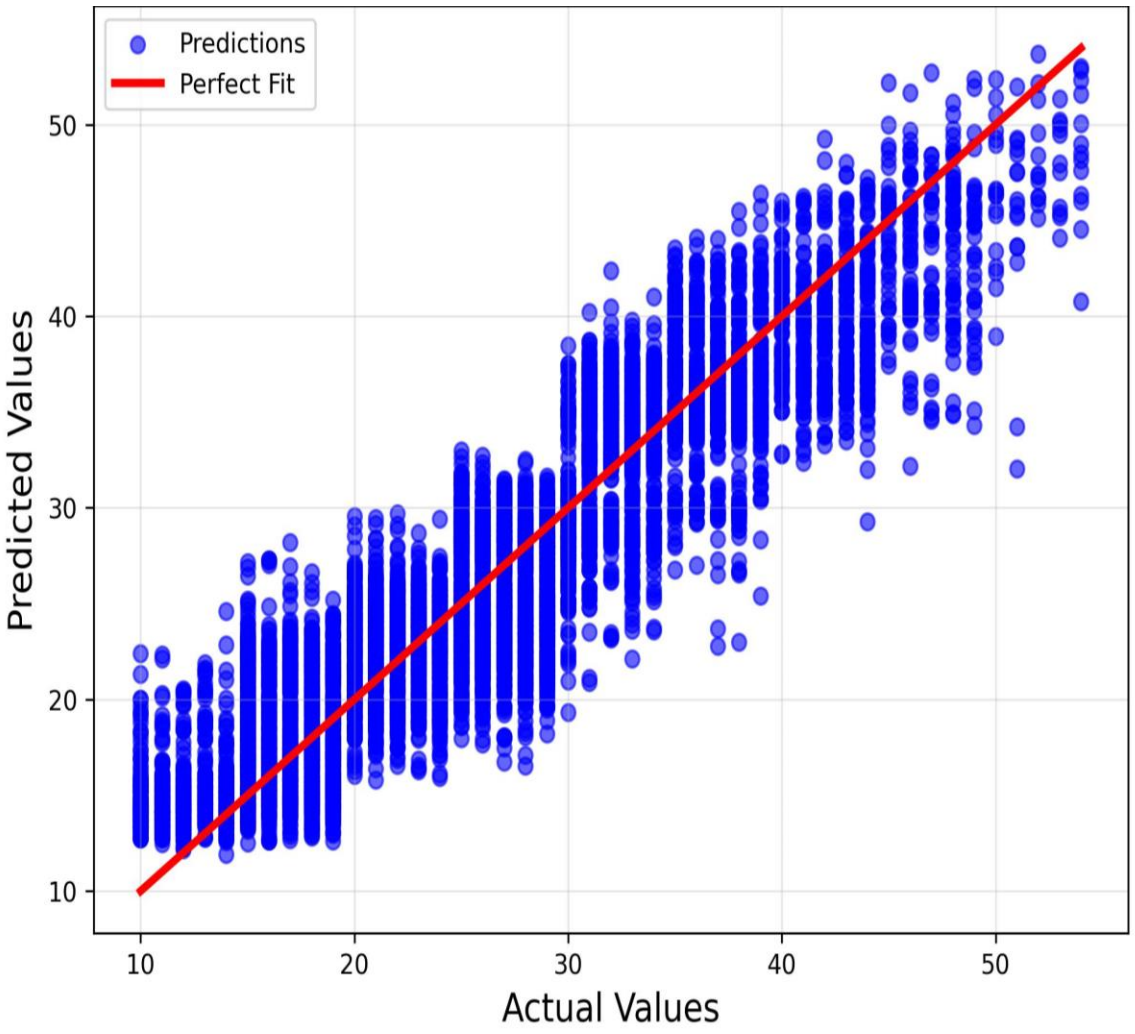
Hyperparameters of LGBM

- Learning_Rate – 0.1
- Max_Depth – 20
- N_Estimators – 200

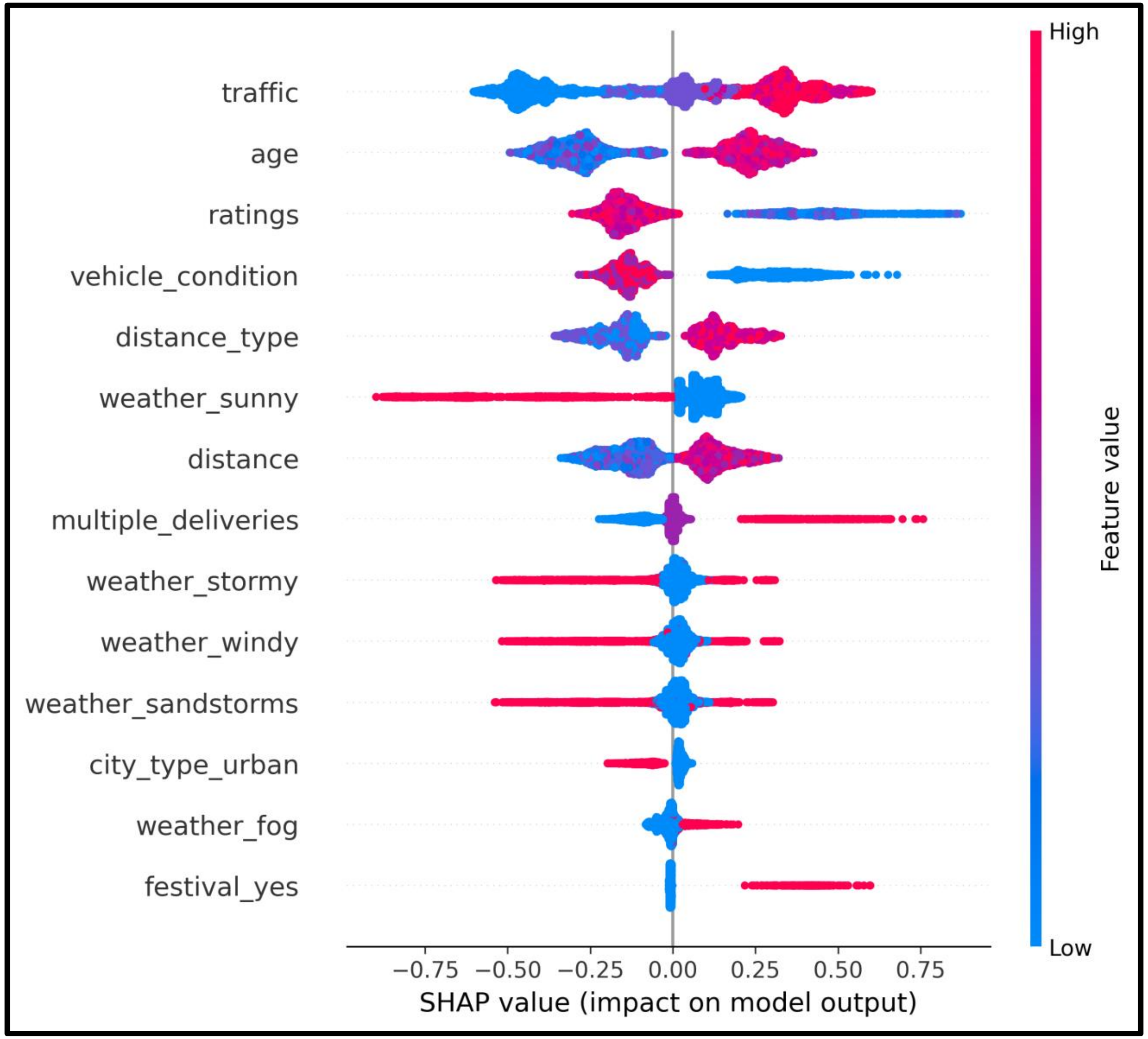
Hyperparameters of XGBoost

- Learning_Rate – 0.1
- Max_Depth – 20
- N_Estimators – 100

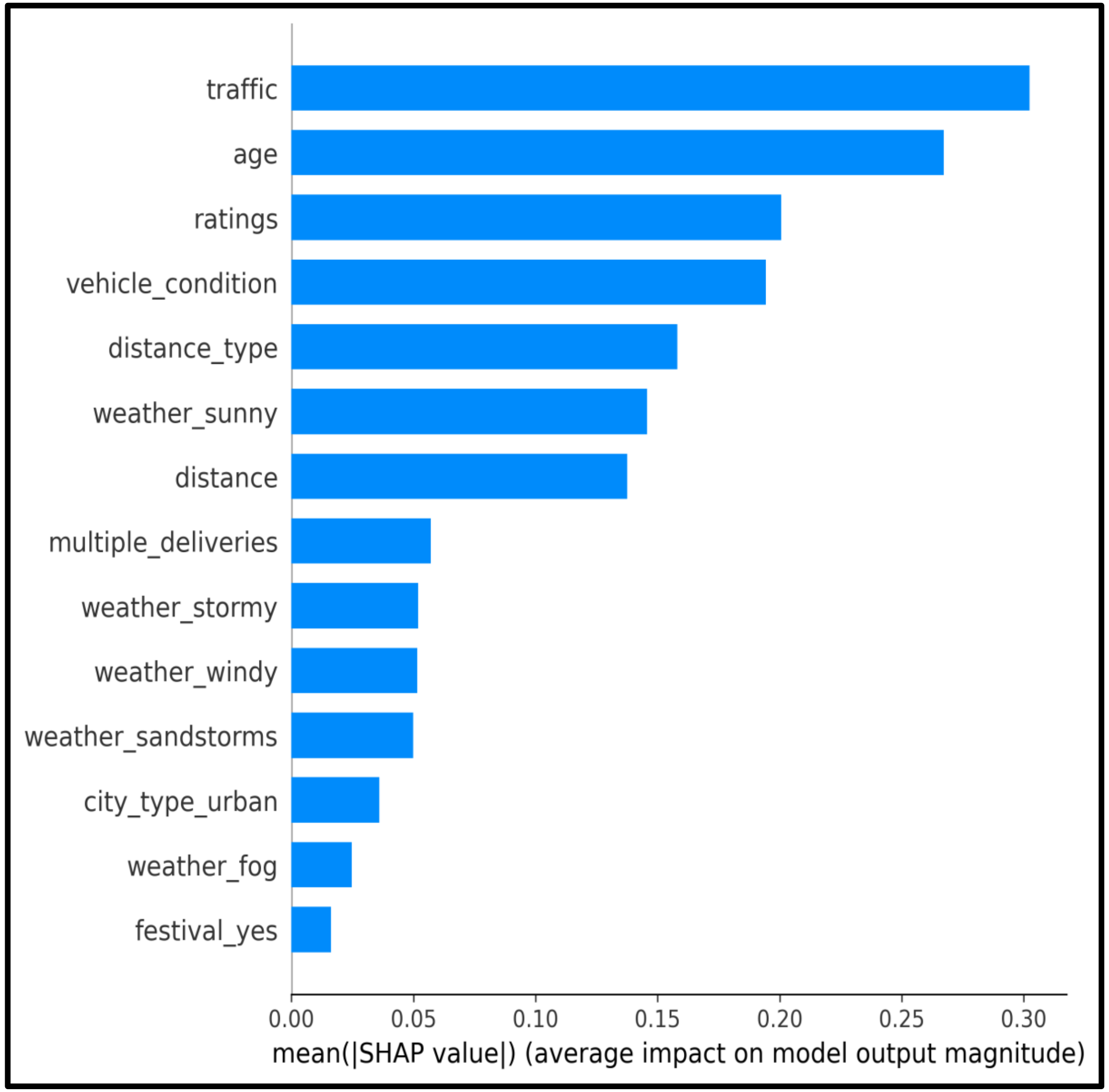
Actual vs Predicted values



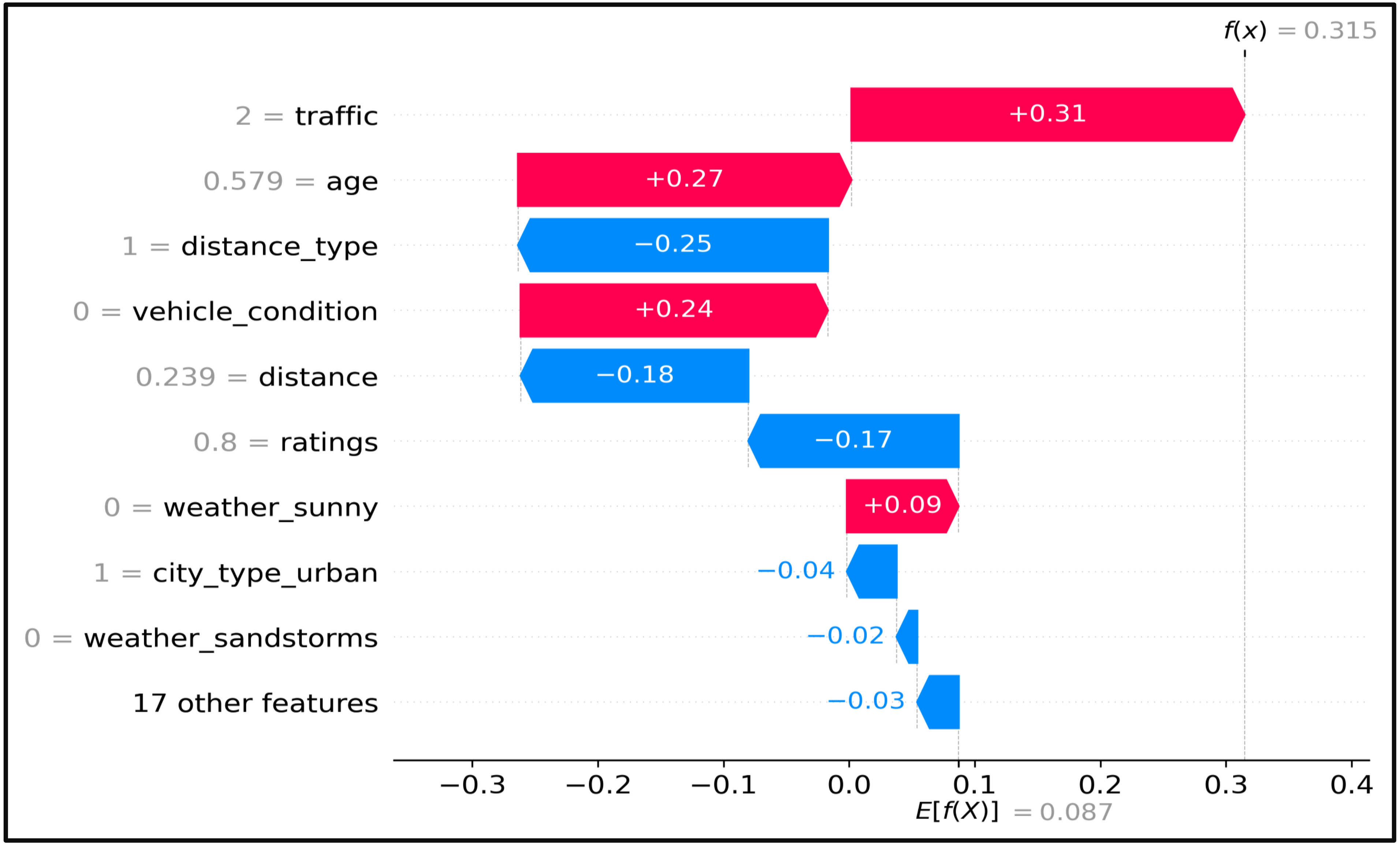
Shapley Summary Plot of Features



Feature Importance – Bar Plot



Impact on Features on Final Output for 1 instance



Original Values

E(f(x)) – 26.33 minutes

f(x) – 28.52 minutes

REFERENCES

1. **Code & Dataset Link**
   https://github.com/rahulnair2402/IIT-Roorkee-Capstone-Project---Food-Delivery-Prediction/tree/Project-branch
2. **Dataset source**
   https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset?select=train.csv
3. Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow - Aurelien Geron
4. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794)
5. LightGBM: A Highly Efficient Gradient Boosting Decision Tree
6. Ensemble Methods: Foundations and Algorithms

COMMENTS ON NEXT STEPS

1. **Optuna for Hyperparameter Tuning:**
   - Use Optuna to optimize the hyperparameter space more efficiently than Grid Search CV
   - Leverage techniques like Bayesian Optimization and early pruning of unpromising trials
2. **Deployment and Monitoring:**
   - Use frameworks like Flask, FastAPI, or Django to expose the model as an API.
   - Implement monitoring tools to track prediction accuracy and latency after deployment