Data-Driven Recipe Selection

Predicting High-traffic Content For Enhanced User Engagement

Challenge:

Currently, the recipes displayed on the homepage generate high traffic to the rest of the website around 60% of the time, and it’s done through an arbitrary and manual process.

Objective:

Build a Machine Learning Algorithm that can correctly predict high-traffic recipes at least 80% of the time, while minimizing the likelihood of recommending unpopular recipes

Information About the Data

After cleaning and Validation, the Dataset contained **895** different recipes grouped into **10 different categories**.

Each recipe’s traffic generation was classified as one of two options (classes):

‘High’ Class:

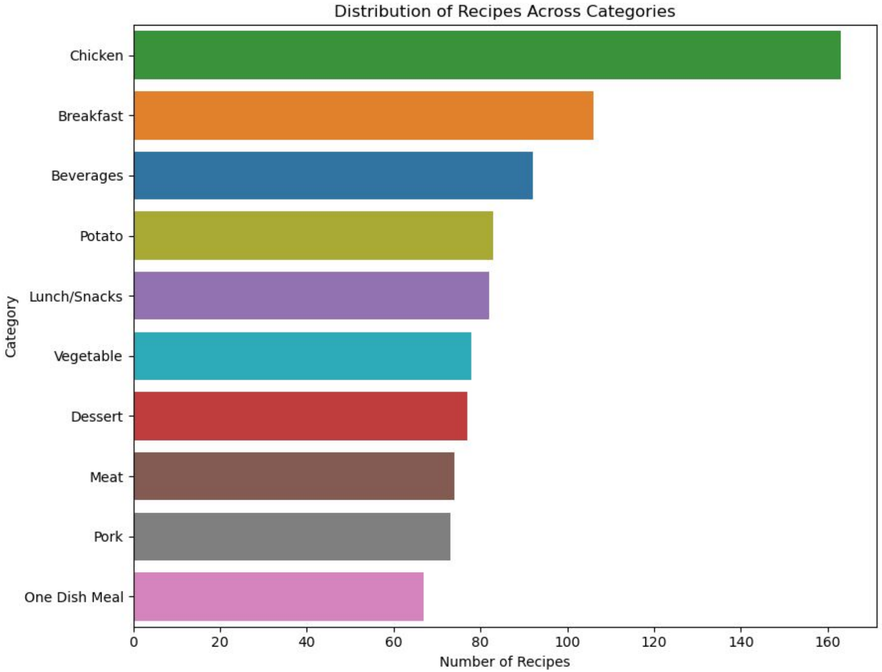
- Recipes that **increased traffic** on the website.

‘Other’ Class:

- Recipes that provided **little to no increase of traffic** on the website.

Exploratory Data Analysis: Recipe Categories

Distribution of Recipes by Category

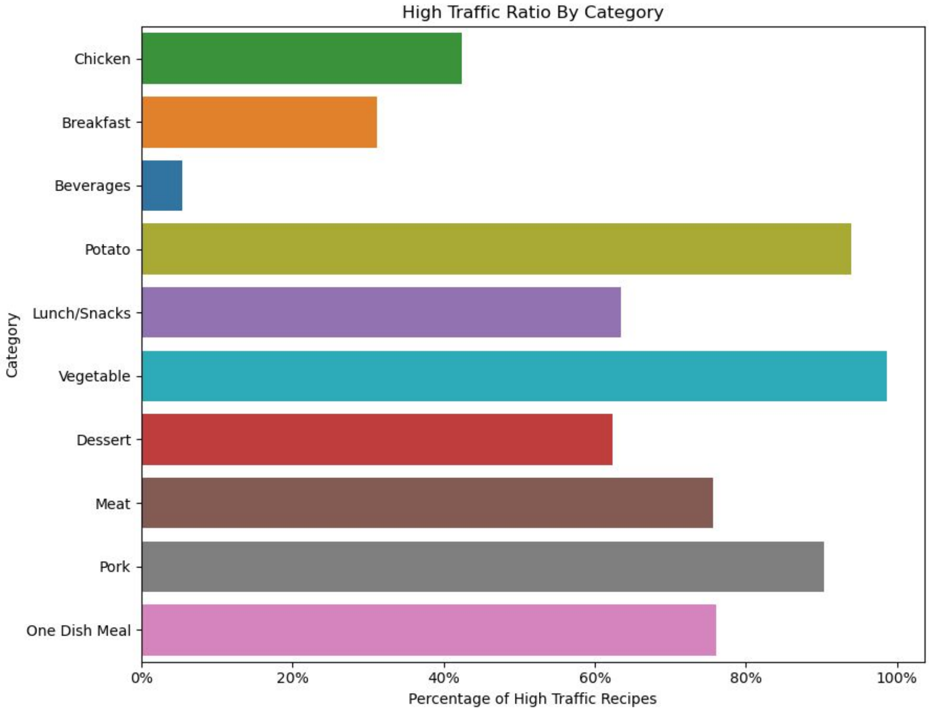


‘Chicken’, ‘Breakfast’ and ‘Beverages’ are the most common recipe categories, representing around **40% of the entire dataset.**

Percentage of High-traffic Recipes by Category

‘Chicken’, ‘Breakfast’ and ‘Beverages’ are the categories that performed worst amongst all.

‘Vegetable’, ‘Potato’ and ‘Pork’ are the **best performing** recipe categories.



Exploratory Data Analysis: Nutritional Features

Discovering Dietary Trends

Nutritional Cluster	Assigned Label	Calories	Carbs	Sugar	Protein
0	**High Protein / High Calorie**	718.73	9.91	8.09	49.1
1	**Low Calorie / Moderate Carb**	237.52	17.62	2.34	6.58
2	**Balanced / High Protein**	110.8	51.56	9.65	37.29
3	**Sweet / Dessert-like**	233.58	26.31	22.97	4.18
4	**High Carb / High Calorie**	762.64	71.36	5.53	22.43

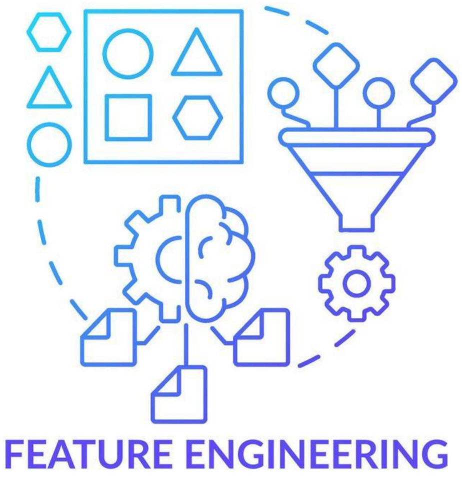
Used K-Means Clustering to **Segment Recipes** by their nutritional profiles.

Allows us to **Uncover Patterns** in traffic ratios. (High-traffic ratios by cluster) Which can be exploited to increase predictive performance.

Can be used by marketing team to **analyze dietary trends** on top of already existing categories.

Interaction Features

Created additional features in order to **capture undiscovered relationships** that can provide deeper insights in recipe popularity, reflect on consumer preferences and enhance predictive power.



Features Created:

1. **High Traffic Ratio by Category**: Highlights which categories are more likely to have high-traffic recipes.
2. **High Traffic Ratio by Cluster**: Shows how different nutritional clusters correlate with traffic patterns
3. **Average Calories by Category**: Provides context about typical calorie ranges within categories.
4. **Nutrition-based interaction features**: aim to capture nutritional density/balance of recipes:
   a. Protein/Calorie Ratio
   b. Sugar/Carb Ratio
   c. Carb/Protein Ratio

Model Development

Logistic Regression, Random Forests and Stacking Classifier.

How are the models trained?

Original Dataset

- Contains the Raw Data after cleaning and validation

VS

Engineered Dataset

Original Dataset + **Engineered** Features.

Evaluation Framework

Primary Metric:

- **F1-Score** for the **“High”** Class: Balances the trade-off between precision (avoiding false positives) and recall (avoiding false negatives)

Supporting Metrics:

- **Precision** for the **"High"** Class: Measures how many of the model’s predicted positives are actually correct.
- **Recall** for the **"High"** Class: Measures the model’s ability to detect true positives.
- **ROC AUC Score**: Provides a single value to evaluate the model’s ability to distinguish between classes across all classification thresholds.
- **Precision** for **“Other”** Class: minimize the risk of falsely excluding potential high-traffic recipes.

Goal: correctly predict high-traffic recipes at least **80% of the time**, while minimizing the likelihood of recommending unpopular recipes

Thresholds:

- **F1-Score**, **Precision** and **Recall** for **‘High’** Class **≥ 0.8**

- **ROC AUC** score: **≥ 0.8**

- **Precision** for **‘Other’** Class **≥ 0.65**

Logistic Regression VS Random Forest

Performance:

Logistic Regression provides **balanced performance** across both classes, achieving better precision and recall for the “Other” class compared to Random Forest.

Random Forest shows a slight **advantage in recall for high-traffic recipes**, but its poor recall for the “Other” class makes it less suitable for avoiding unpopular recommendations.

Logistic Regression’s higher ROC AUC score (0.84 vs. 0.82) demonstrates **stronger** overall discriminatory power between high- and low-traffic recipes.

Caveats:

Both models performed marginally better with the Engineered Dataset, however, **overreliance on a single feature** (High-traffic ratio by Category) raised serious **concerns about overfitting**.

Class imbalance led to favoring the majority (‘High’) class, which resulted in **sub-optimal performance** when identifying the minority (‘Other’) class.

Stacking Classifier

Metric	Performance Values	Comparison to Best-performing Model
Precision (High)	0.82	+ 0.02
Precision (Other)	0.72	0.00
Recall (High)	0.80	- 0.02
Recall (Other)	0.74	+ 0.05
F1-Score (High)	0.81	0.00
F1-Score (Other)	0.73	+ 0.02
Accuracy	0.78	+ 0.01
ROC AUC Score	0.84	0.00

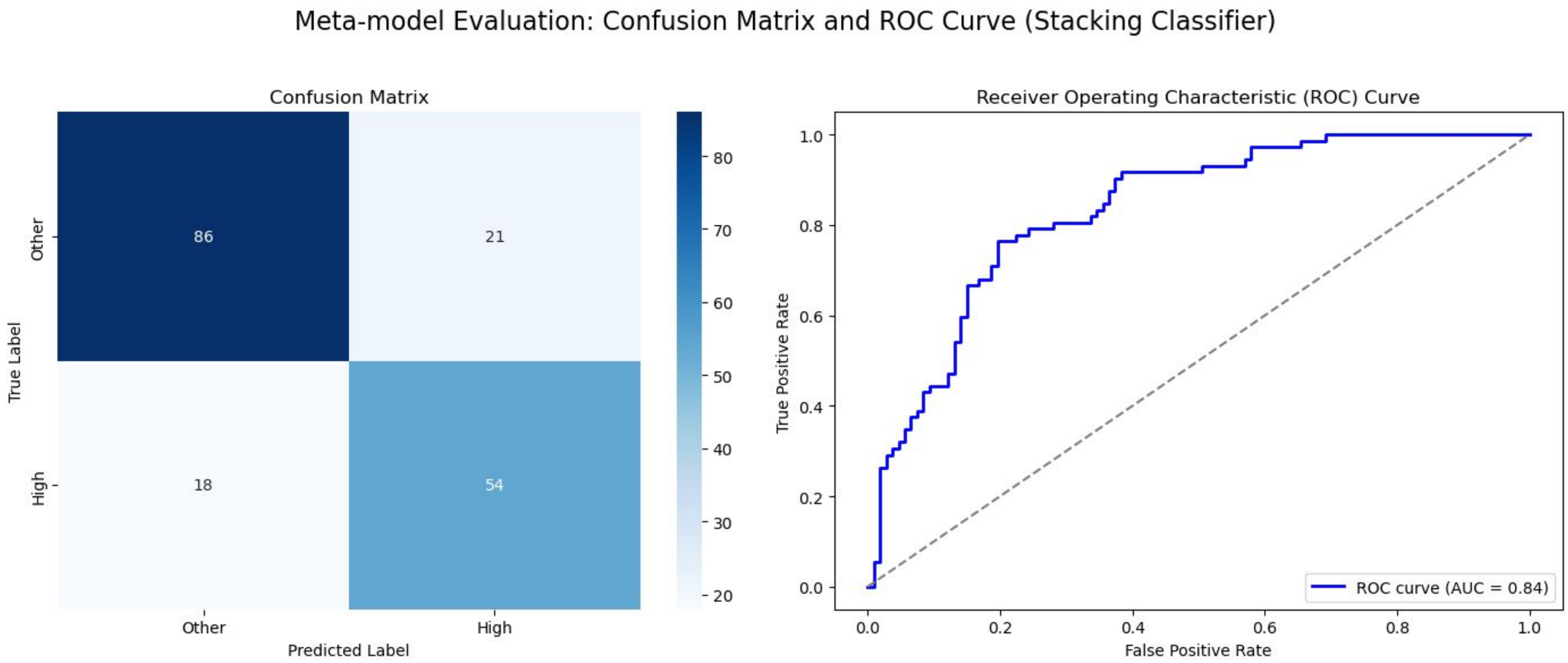
Stacking Classifier combines the **strengths** of both the **Logistic Regression** and the **Random Forests** models, working as the base predictors. The meta-model (final predictor) was the Logistic Regression.

Trained on **Original Data**

Adjusted Class weights to **reduce class imbalance.**

Best performance overall, and is expected to perform well with unseen data.

Stacking Classifier Model Evaluation (Original Data)



Key Recommendations and Monitoring Plan

Recommendations

1.- Deploy Stacking Classifier

- Best-performing model overall, meeting the requirements previously set.

2.- Address Class imbalance

- Try Oversampling, undersampling or cost-sensitive learning.

3.- Scale and Validate the Model

- Test on new / external data to validate performance and ensure robustness

4.- Continuous Monitoring and Feedback

- Implement monitoring system to track performance.
- Stay up-to-date with seasonal trends.
- Incorporate user feedback to refine model.

5.- Strategic Model Integration

- Expand model to enhance recommendations for subscribers.
- Integrate prediction system into marketing campaigns.

Monitoring Plan

Monitoring Process:

- **Daily Metrics Review:** Track **recall** for **high-traffic recipes** daily to ensure robust performance.
- **Recipe Performance Tracking**: Analyze **user engagement** (e.g. clicks, time spent on website, new subscriptions) for recipes displayed at the homepage to better quantify the effect of a high-traffic recipe and improve the model.
- **Periodic Model Reevaluation:** Perform **monthly checks** to ensure models predictions remain as intended.

Metrics to Follow:

- Recall for “High” Class.
- Precision for “High” Class.
- F1-Score for “High” Class.
- Precision for “Other” Class

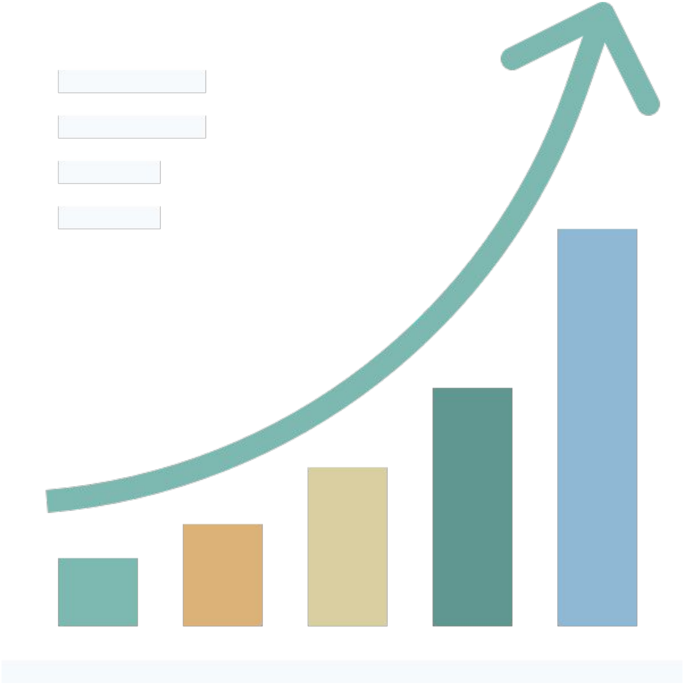
Metric Thresholds

Good Performance:

- **Prediction Accuracy:** Between 70%-80%
- **Precision, Recall and F1-Score for "High" Class**: between 0.7-0.8.
- **Precision for "Other" Class:** Above 68%

Optimal Performance:

- **Prediction Accuracy:** Above 80%
- **Precision, Recall and F1-Score for "High" Class:** Above 0.8.
- **Precision for "Other" Class:** Above 68%



How Are The Metrics Calculated?

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP)\ +\ False\ Positives\ (FP)}$$

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP)\ +\ False\ Negatives\ (FN)}$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

		Predicted Values	
		Positive	Negative
True Values	Positive	**True Positive**	**False Negative**
	Negative	**False Positive**	**True Negative**

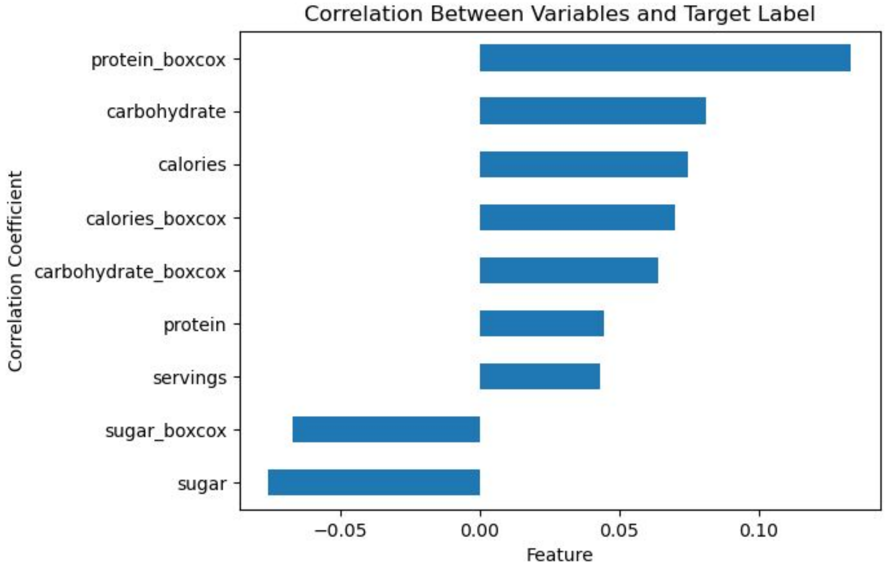
Thank you!

Support slides

Project Overview and Business Goals

- **Challenge**:
  - Recipe selection relies on a manual and subjective process, leading to inconsistent traffic outcomes.
- **Opportunity**:
  - Replace manual selection (60% accuracy) with a predictive model that can enhance traffic on website, leading to an increase in new subscriptions.
- **Goal**:
  - Develop a data-driven model to predict which high-traffic recipes, maximizing user engagement and increasing subscriptions.

Nutritional Features and Correlation with Target

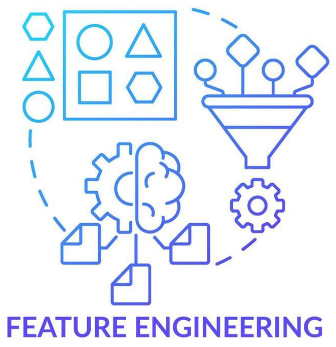


Box-cox transformations were applied to **reduce skewness** and **normalize** the nutritional **features.**

There is Little to no **linear** relationship between nutritional features and target label, even after transformations.

Feature Engineering

Other Interaction Features



Purpose:

- **Capture relationships** between nutritional components.

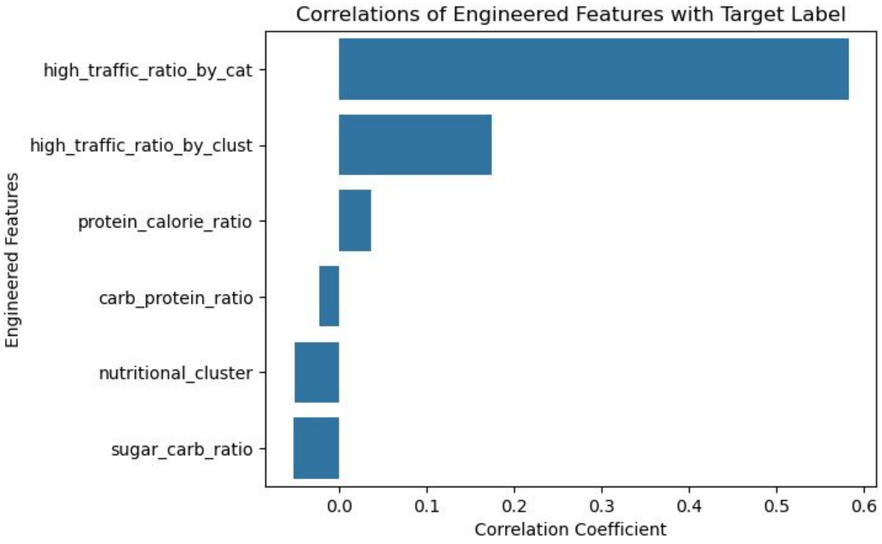
Value:

- Provide **deeper insights** into recipe popularity.
- Reflect consumer **preferences**.
- Enhance **predictive power**.

Engineered Features:

- High-Traffic by category Ratio
- High-Traffic by cluster Ratio
- Protein-Calorie Ratio
- Sugar-Carbohydrate Ratio
- Carbohydrate-Protein Ratio

Correlation Between New Features and Target



- **High-Traffic Ratio by Category** has the **strongest linear relationship** with the target label. (Correlation coefficient = ~0.6)
- Despite other engineered features exhibiting weak linear relationships, they can still **capture nonlinear patterns** that advanced models can exploit.

Logistic Regression Model Evaluation

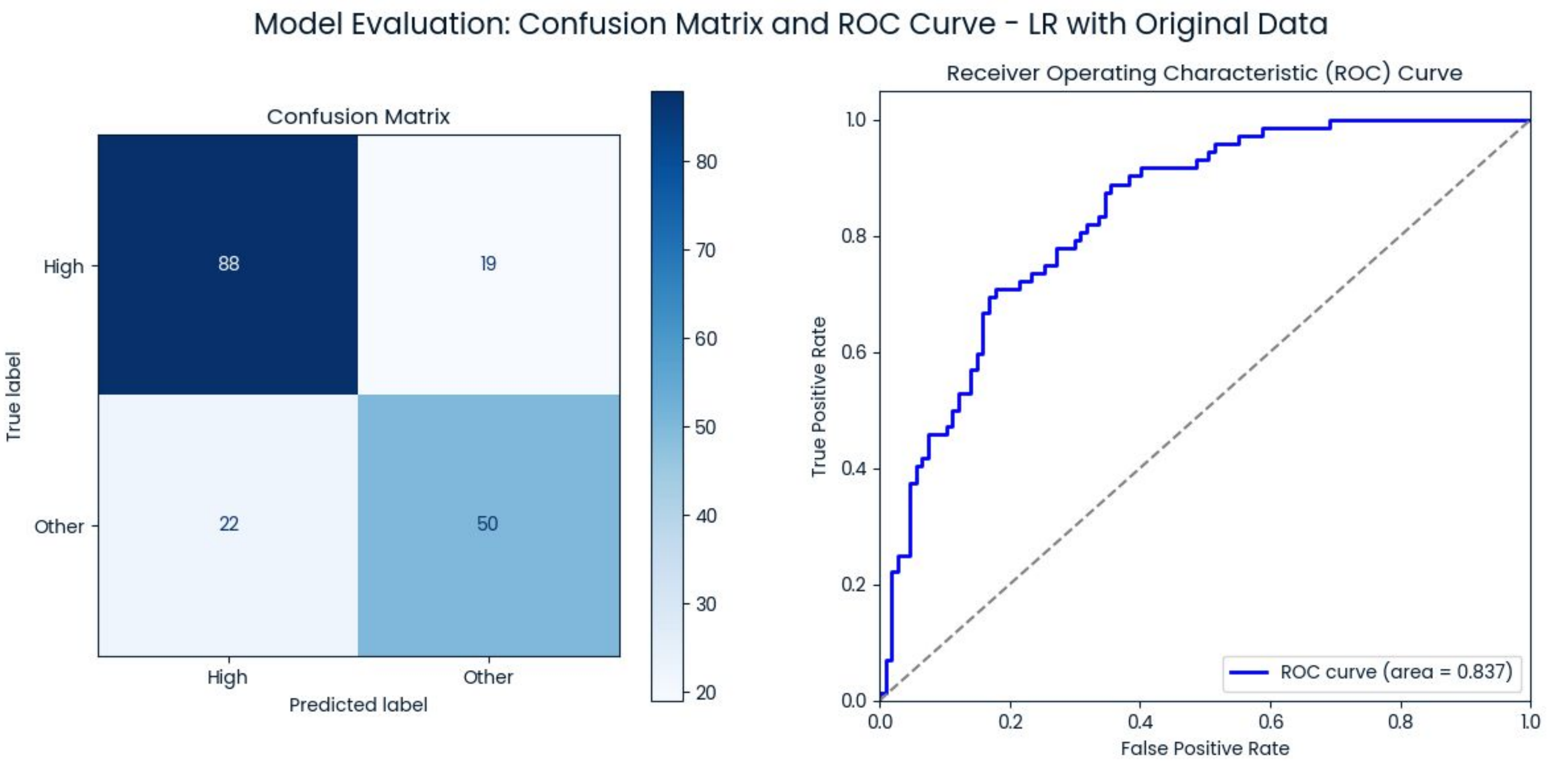
Metric	Original Data	Engineered Data
Precision (High)	0.80	0.81 + 0.01
Precision (Other)	0.72	0.73 + 0.01
Recall (High)	0.82	0.82 + 0.00
Recall (Other)	0.69	0.71 + 0.02
F1-Score (High)	0.81	0.81 + 0.00
F1-Score (Other)	0.71	0.72 + 0.01
Accuracy	0.77	0.78 + 0.01
ROC AUC Score	0.84	0.84 + 0.00

- Slightly **better performance** for the model trained on the **Engineered Data**. However, over reliance on a single feature (High-traffic ratio by category), which raised concerns of **overfitting**.

- Model Trained on the **Original Data** provided **better reliability** and has the potential of working **best with unseen data**.

- **Lower Performance overall for "Other"** Class raised concerns of **class imbalance.**

Logistic Regression Model Evaluation (original Data)



Random Forest Model Evaluation

Metric	Original Data	Engineered Data
Precision (High)	0.73	0.76 **+ 0.03**
Precision (Other)	0.69	0.71 **+ 0.02**
Recall (High)	0.84	0.83 **- 0.01**
Recall (Other)	0.53	0.61 **+ 0.08**
F1-Score (High)	0.78	0.79 **+ 0.01**
F1-Score (Other)	0.6	0.66 **+ 0.06**
Accuracy	0.72	0.74 **+ 0.02**
ROC AUC Score	0.82	0.81 **- 0.02**

- **Better Performance** on the model Trained on **Engineered Data**. Considerable increase in Recall and F1-Score for “Other” Class.
- Lower performance on “Other” class (compared to the “High” class) continues to raise concerns about **class imbalance**. Despite of bootstrapping efforts.
- Similar to the Logistic Regression model, **over reliance on “High-traffic ratio by category” (18.7% importance)** indicates the **potential of performing poorly on unseen data.**

Random Forest Model Evaluation (Original Data)