Data Scientist Professional Practical Exam Presentation

Dr. Anoushiravan Zahedi

Tasty Bytes

- ***Company Description***
- Tasty Bytes was founded in 2020 in the midst of the Covid Pandemic.
    - Search engine for recipes
    - Monthly subscription -> full healthy, balanced diet, meal plan whatever your budget.
- ***Problem Description***
    - At the moment, the owner chooses their favorite recipe from a selection and displays it on the home page.
    - The company has noticed that traffic to the rest of the website goes up by as much as 40% if they pick a popular recipe.
- **We should:**
    - *Predict which recipes will lead to high traffic?*
    - *Correctly predict high traffic recipes 80% of the time?*

Data Validation

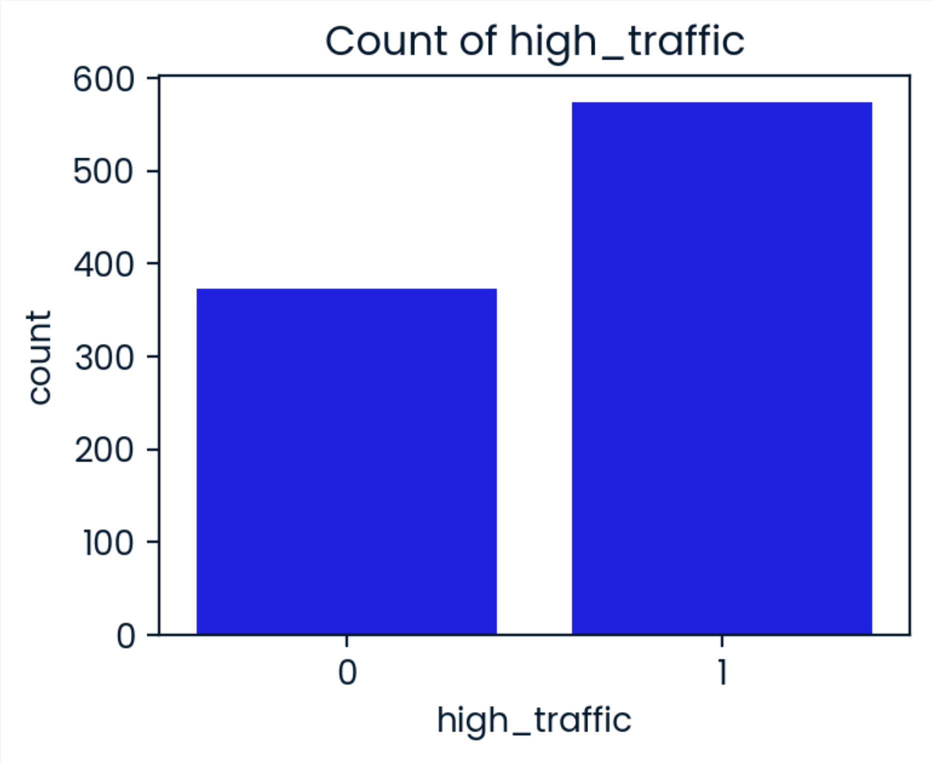
Columns

Column Name	*Details*	*Changes*
recipe	Numeric, unique identifier of recipe	No duplicates: Eliminated for the final modeling
calories	Numeric, number of calories	(1) Validated values are within logical range: i.e., non-negative (2) 52 rows out of 947 (5.5%) had missing values for nutritional columns. Imputed based on servings and category columns
carbohydrate	Numeric, amount of carbohydrates in grams	
sugar	Numeric, amount of sugar in grams	
protein	Numeric, amount of protein in grams	
category	Character, type of recipe. Recipes are listed in one of ten possible groupings: 'Lunch/Snacks', 'Beverages', 'Potato', 'Vegetable', 'Meat', 'Chicken', 'Pork', 'Dessert', 'Breakfast', 'One Dish Meal'.	Corrected typos and inconsistent naming (['Chicken Breast']-> ['Chicken']).
servings	Numeric, number of servings for the recipe	(1) Removed text entries like "as a snack" (2) Converted to a category type.
high-traffic (Target Variable)	Character, if the traffic to the site was high when this recipe was shown, this is marked with "High".	(1) 373 from 947 [39.4%] were NaN -> low traffic (2) Recoded: 0 = low traffic, 1 = high traffic

Exploratory analysis

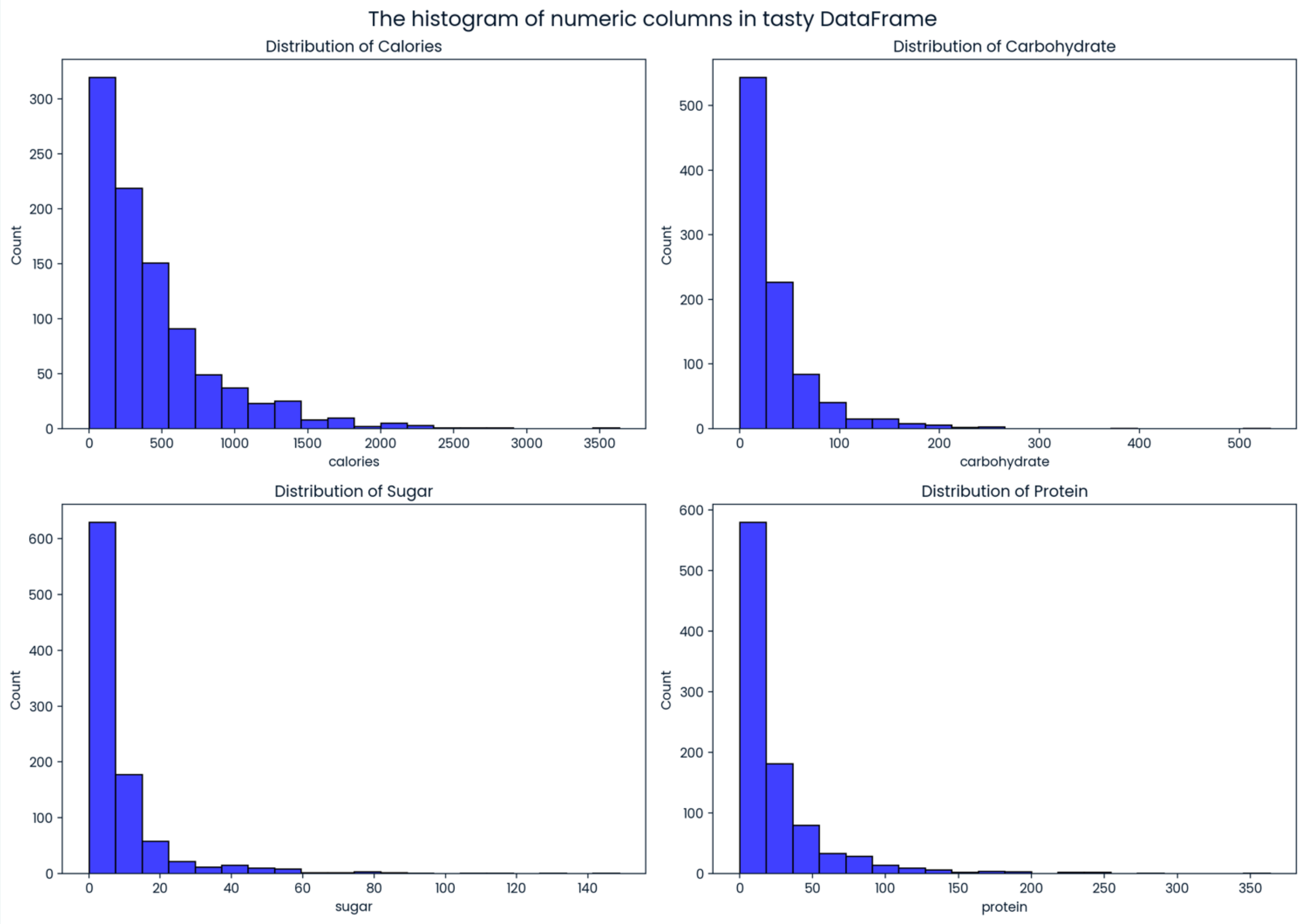
High Traffic Distribution (Count Plot)

- Class imbalance: imblearn.over_sampling.SMOTE()
- The baseline accuracy ~ 60% (majority class)



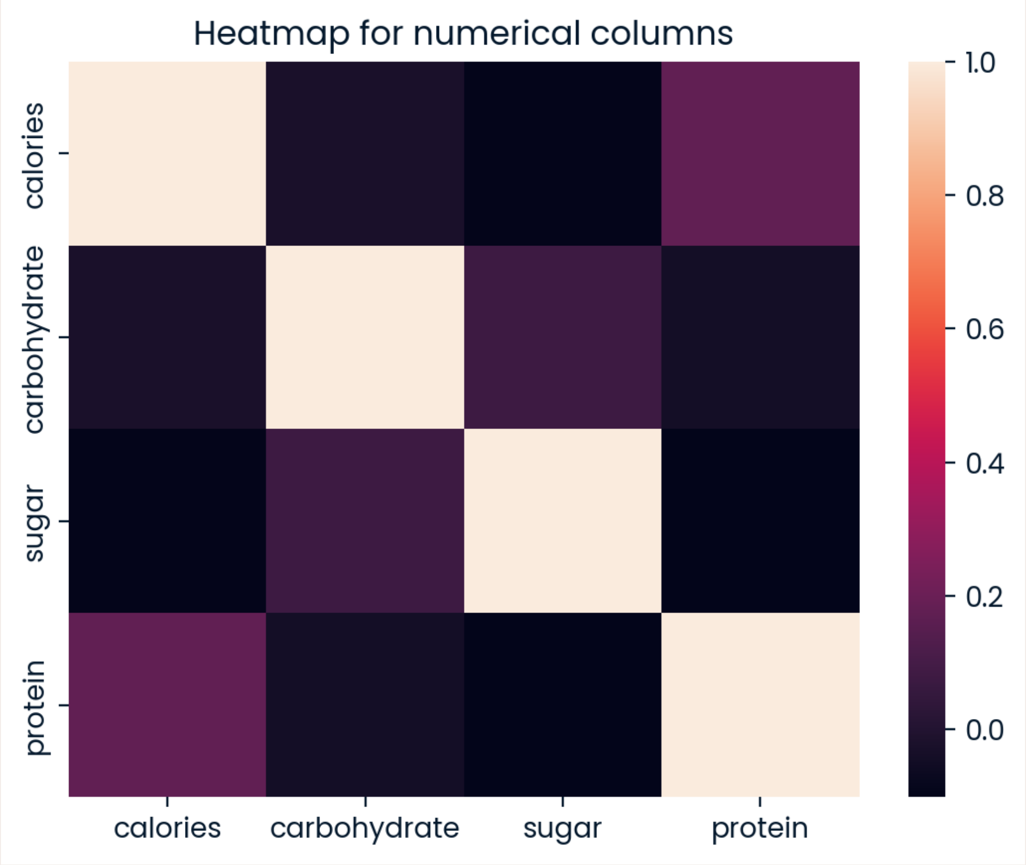
Numerical Feature Distributions (Histograms)

- Nutritional features show right-skewed distributions: PowerTransformer()
- Different scales (calories in hundreds vs. protein in tens): StandardScaler()



Correlation Heatmap

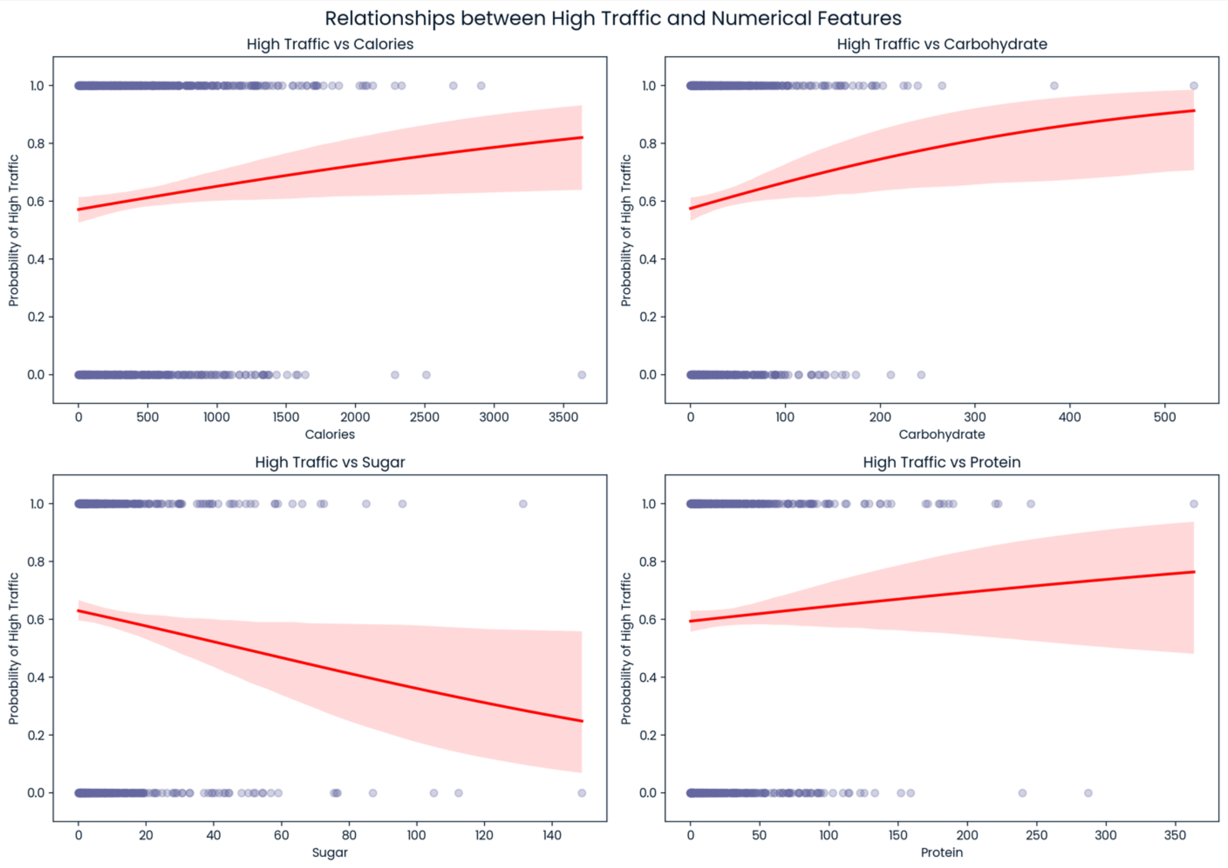
- Heatmap helps identify redundant features.
- Weak correlations between features, implying no multicollinearity.



Numerical Features vs. High Traffic

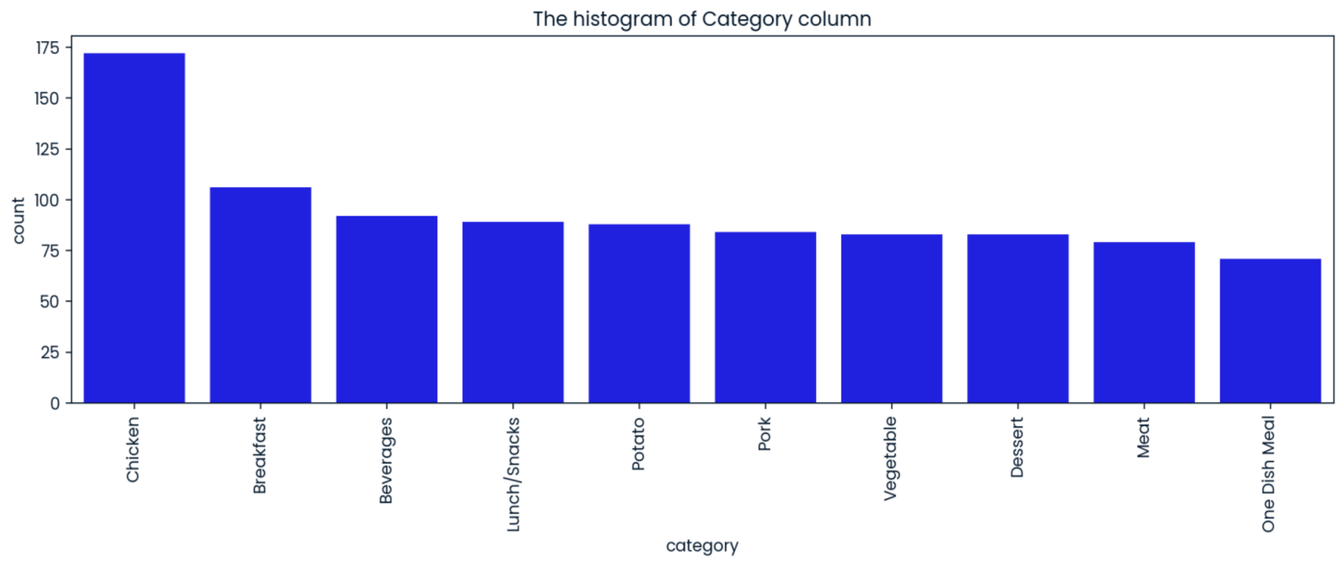
- seaborn.regplot shows how nutritional features (calories, protein, sugar, carbohydrates) predict whether recipes are high- or low-traffic.

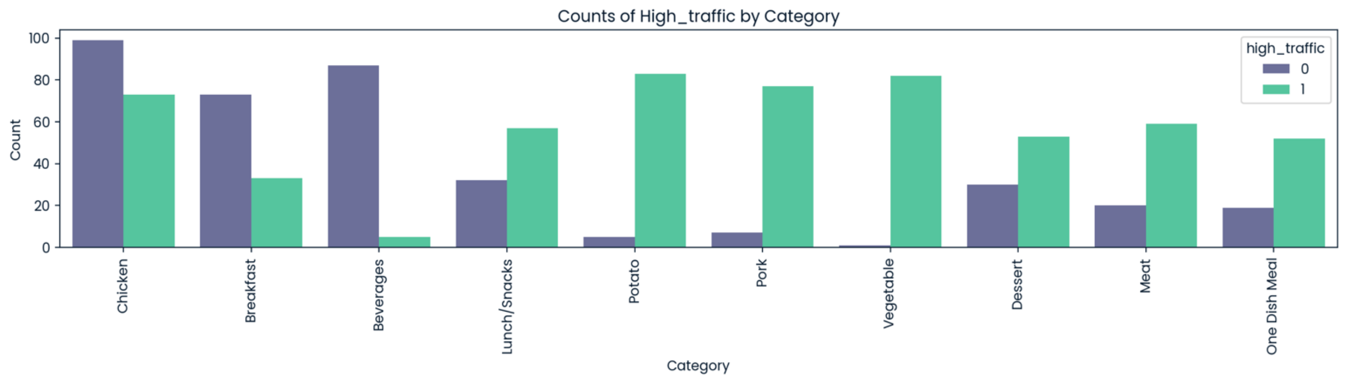
- High-traffic recipes tend to have higher carbohydrate, calorie, and protein content and lower sugar.



Categorical Feature Distributions and Categorical Features vs. High Traffic

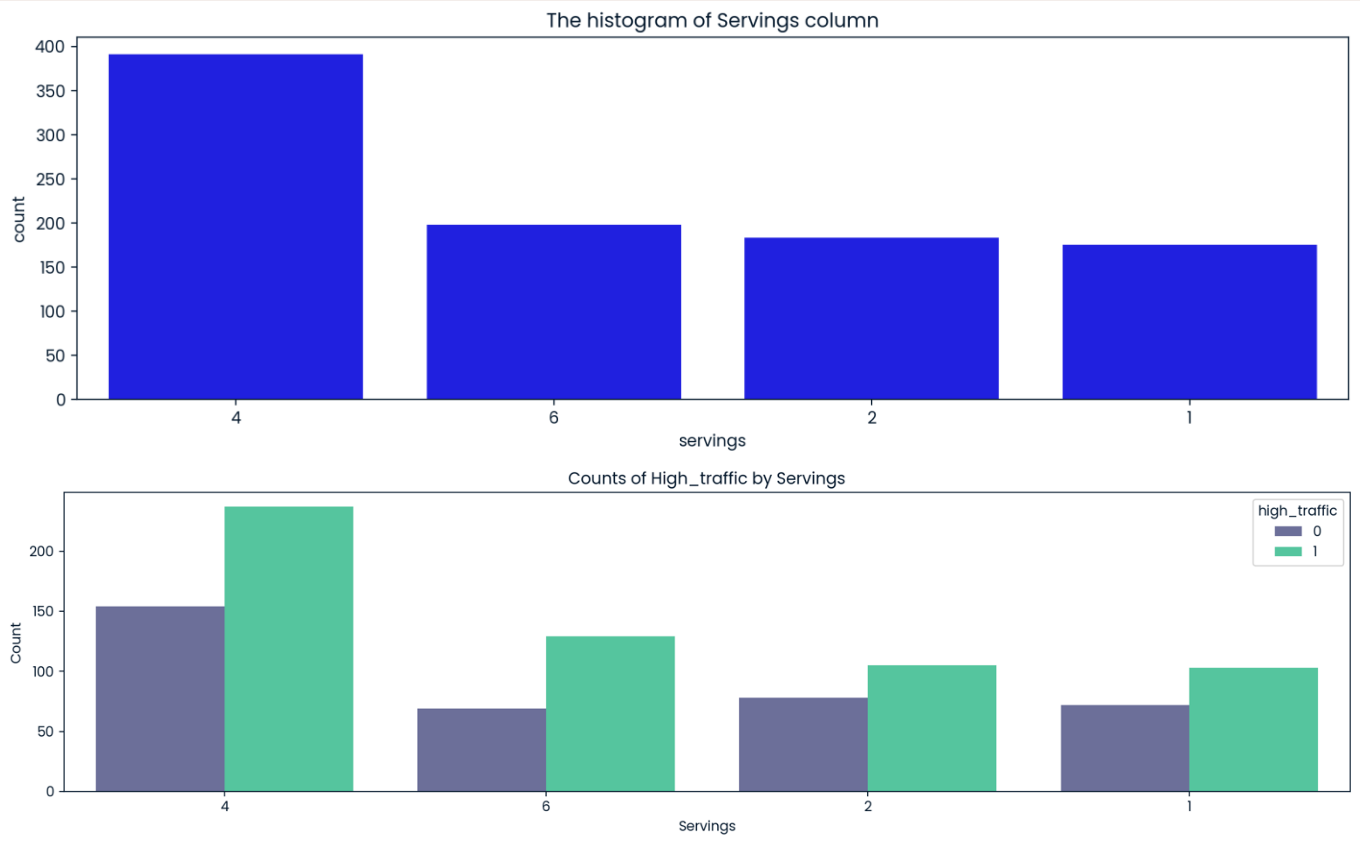
- Specific categories (e.g., Potato, Pork, and Vegetable) have higher proportions of high-traffic recipes.





Categorical Feature Distributions and Categorical Features vs. High Traffic

- Serving sizes for 4-6 people might be more popular than individual servings



Model Development

- ***Problem Type:***
  - *This is a binary classification problem.*
- ***Preprocessing Strategy:***
  - *Train-test split (80/20) with stratification to maintain class distribution*
  - *SMOTE to address class imbalance by generating synthetic minority class examples*
  - *Numerical features:*
    - *Median imputation (already implemented during data validation) +*
    - *PowerTransformer with yeo-johnson method +*
    - *StandardScaler normalization*
  - *Categorical features: One-hot encoding*
- ***Hyperparameter tuning:***
  - *GridSearchCV (for simpler models) and RandomizedSearchCV (for the most complicated model) with 5-fold cross-validation*

Model Selection Rationale

- **Baseline Model: Logistic Regression**
  - (**LR**): It is well-suited for binary classification problems
  - (**LR**): It is computationally efficient and fast to train
  - (**LR**): It works well when there is a roughly linear relationship between features and the log-odds of the target

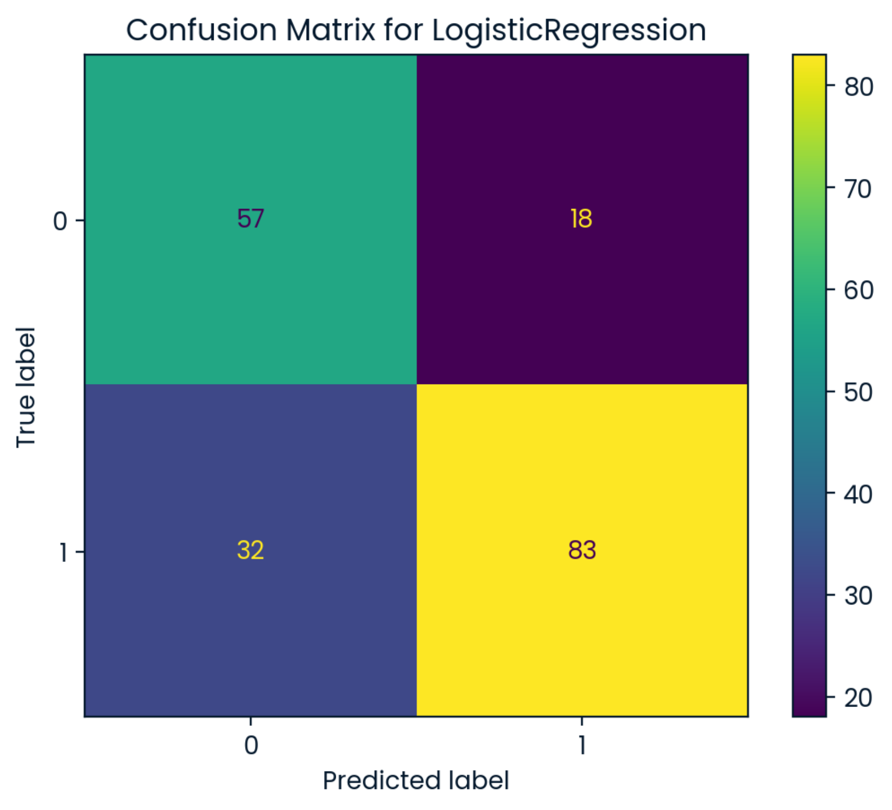
- **Comparison Model 1: Support Vector Classifier (SVC)**
  - **(SVC):** It can capture non-linear relationships through kernel functions (RBF kernel)
  - **(SVC):** It works well in high-dimensional spaces (important after one-hot encoding categorical features)

- **Comparison Model 2: Gradient Boosting (XGBoost)**
  - **(XGBoost):** It automatically captures feature interactions and non-linear relationships
  - **(XGBoost):** It handles missing values and outliers well

Model Evaluation and Selection

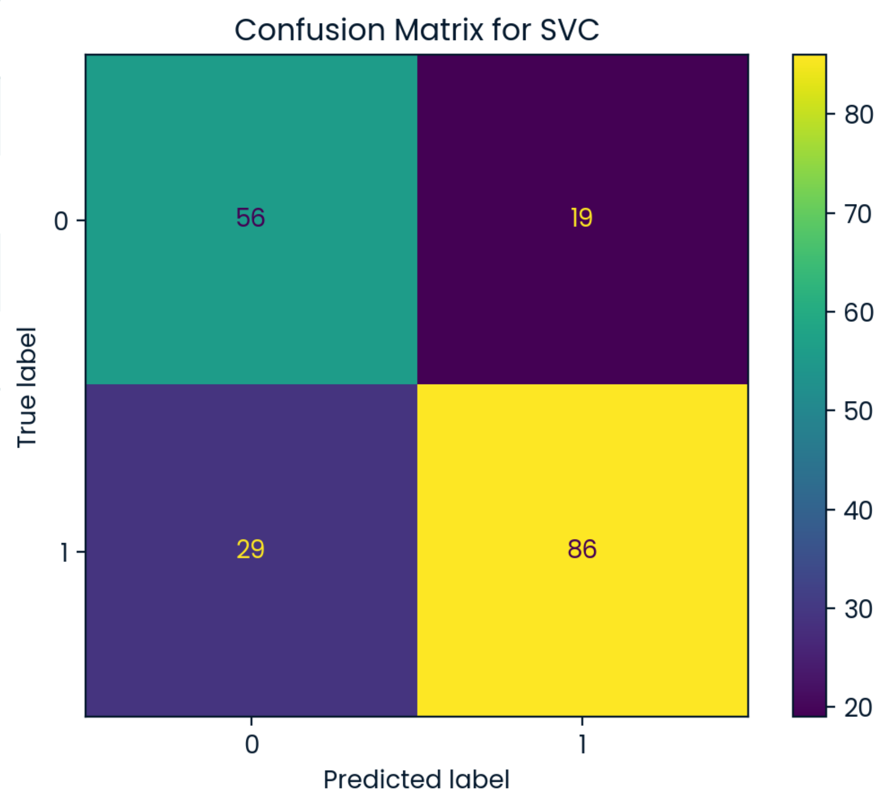
• Logistic Regression

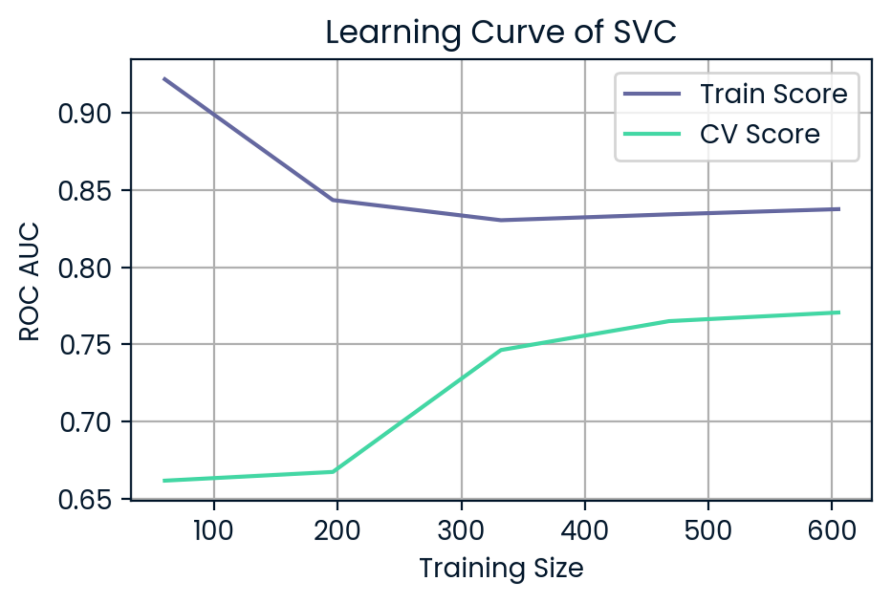
Metric	Value
Accuracy	73.68%
Precision	82.18%
Recall	72.17%
F1 Score	76.85%



• Support Vector Classifier

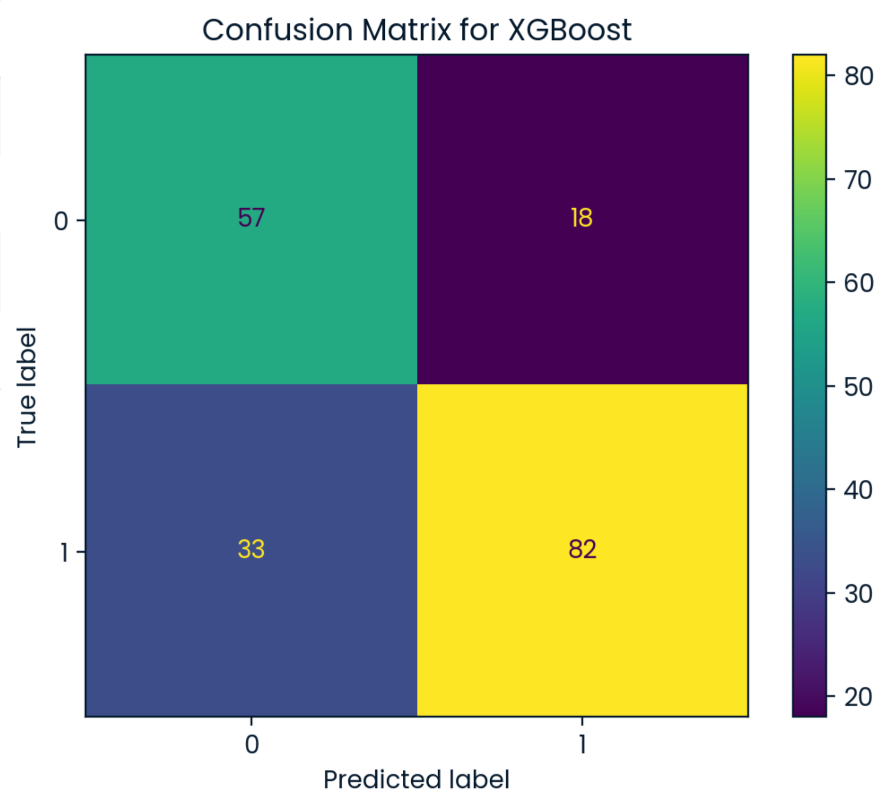
Metric	Value
Accuracy	74.74%
Precision	81.90%
Recall	74.78%
F1 Score	78.18%

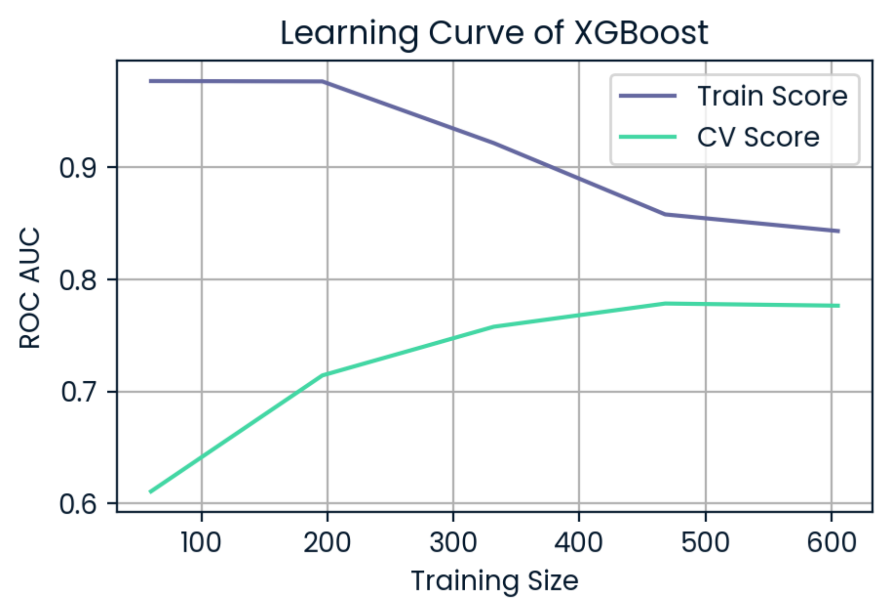




- # Gradient Boosting Classifier

Metric	Value
Accuracy	73.68%
Precision	82.18%
Recall	72.17%
F1 Score	76.85%





Best Model Selection

Based on test-set performance, the model with the highest F1 score is selected.

The models will be ranked in terms of complexity as follows:

1. Logistic Regression Model
2. Support Vector Classifier
3. Gradient Boosting Classifier

Support Vector Classifier

Metric	Value
Accuracy	74.74%
Precision	81.90%
Recall	74.78%
F1 Score	78.18%

Business Metrics

Cost-Benefit Analysis

- True Positive (Correct high-traffic prediction): Business promotes the recipe and gains traffic/revenue.
- False Positive (Incorrectly predict high traffic): Business wastes resources promoting a low-traffic recipe.
- True Negative (Correct low-traffic prediction): Business correctly avoids promoting low-traffic recipe.
- False Negative (Miss a high-traffic recipe): Business misses opportunity to promote popular recipe.

```
ROI = (TP × $100) + (FP × -$20) + (TN × $0) + (FN × -$80)
```

- **Logistic Regression:**
  - ROI = (83 × $100) + (18 × -$20) + (32 × -$80) + (57 × $0)= $8,300 - $360 - $2,560 + $0= $5,380
- **SVC:**
  - ROI = (86 × $100) + (19 × -$20) + (29 × -$80) + (56 × $0)= $8,600 - $380 - $2,320 + $0= $5,900
  - Compared with Logistic Regression, SVC yields a *higher ROI*, providing **better business value**.
- **Gradient Boosting Classifier (XGBoost):**
  - ROI = (82 × $100) + (18 × -$20) + (33 × -$80) + (57 × $0)= $8,400 - $360 - $2,640 + $0= $5,200
  - XGBoost has a lower ROI than both SVC and Logistic Regression.

Final Summary and Recommendations

- After a comprehensive evaluation, the **Support Vector Classifier (SVC)** achieved the best performance with:
- This model successfully predicts recipe traffic with 74.74% accuracy, significantly outperforming the baseline accuracy of 60% (majority class prediction). Further, the model satisfies the task assignment with 81.90% precision, meaning it can predict a high-traffic recipe above the required threshold of 80%.
- **Business Recommendations:**

1. Content Strategy
2. Resource Allocation
3. Content Optimization
4. Continuous Improvement

Thank you

Dr. Anoushiravan Zahedi

anoushiravanzahedi@gmail.com