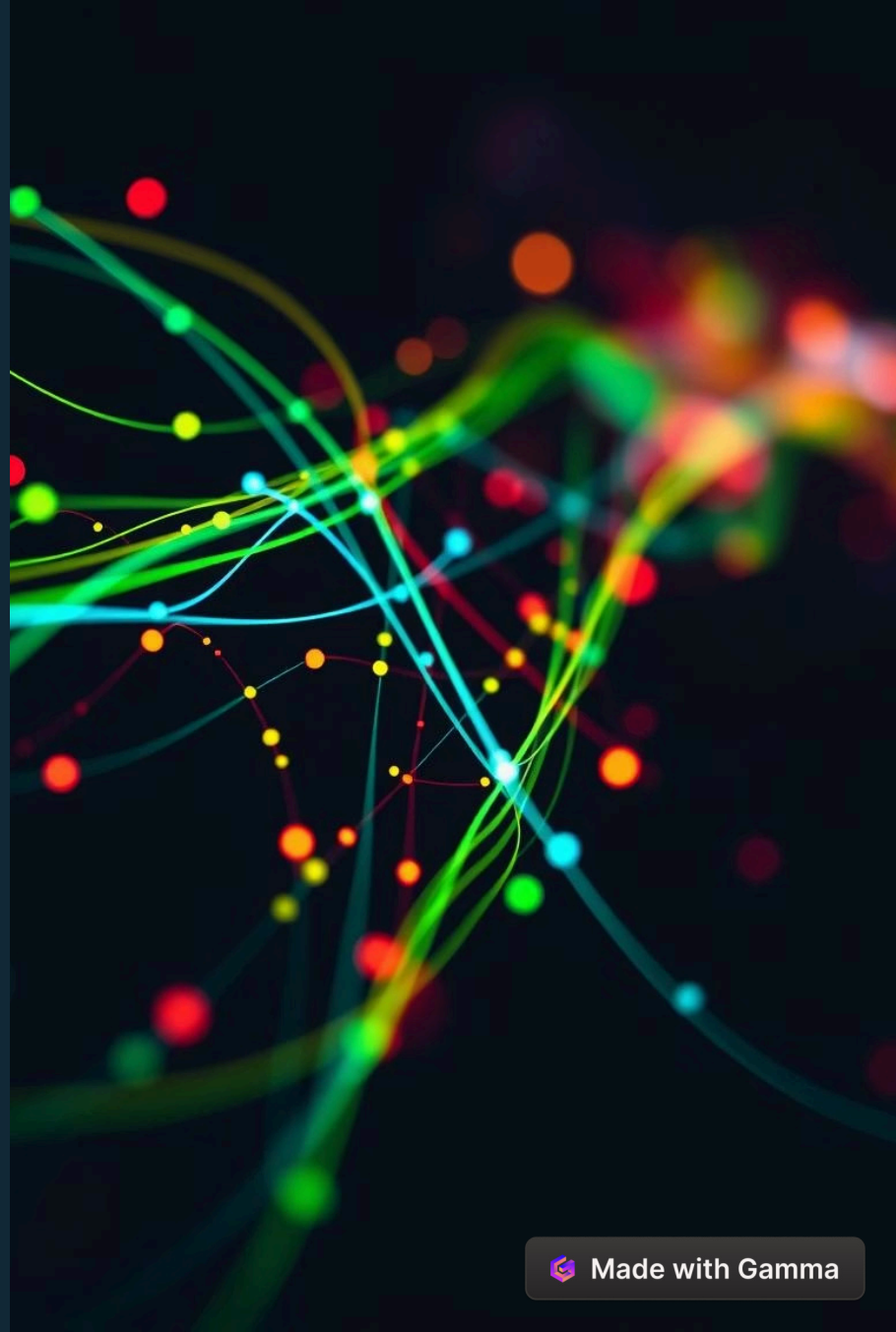
Practical Exam: Predicting Recipe Traffic (Mohamed Nasr)

In this practical exam, we'll dive into the world of recipe traffic prediction . We'll analyze a dataset with valuable information about 947 recipes and leverage data science techniques to forecast which recipes will gain popularity.



Data Presentation

We analyzed a dataset of **947 recipes** provided by the product manager, containing **7 key columns**:

- **Nutritional info**: Calories, carbohydrates, sugars, proteins
- **Category**: One of 10 types (e.g., *Lunch/Snacks*, *Desserts*, *One Dish Meals*)
- **Servings**: Number of people each recipe serves
- **High Traffic Indicator**: Whether the recipe gained high visitor traffic when featured on the homepage

This dataset forms the foundation for building a **predictive model** to identify **popular recipes** and understand **traffic trends**.

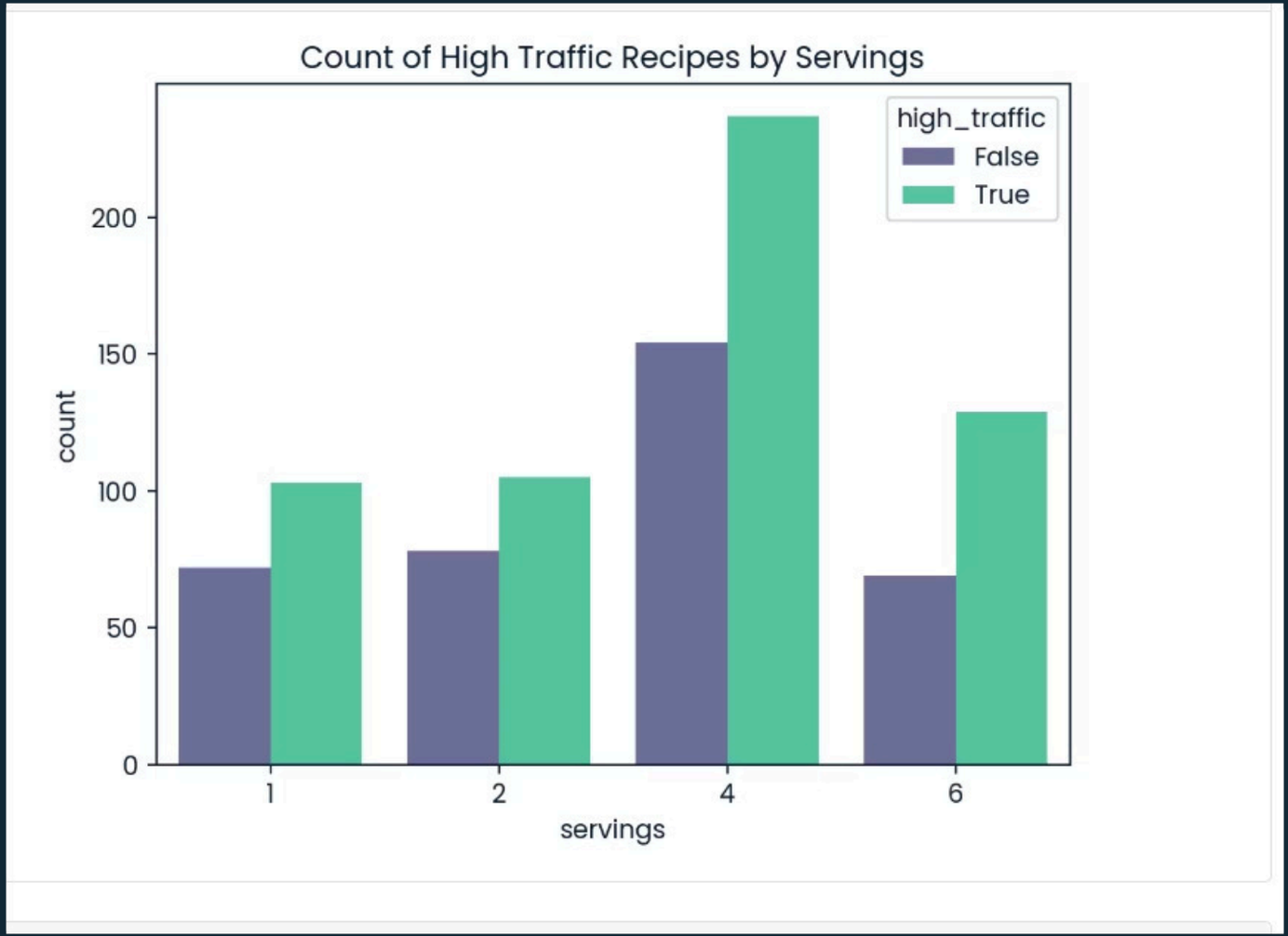
Data Validation

In our analysis of the recipe popularity prediction model, we performed several key steps to ensure data integrity:

1. **Dataset Review**: We verified the dataset of 947 recipes, confirming no duplicates and evaluating the integrity of each column.
2. **Servings Column**: Adjusted to maintain its numeric data type, ensuring flexibility during model development.
3. **High Traffic Column**: Converted from an object type to boolean, representing "High" and "Low" traffic based on the presence or absence of values.
4. **Category Column**: Reclassified an inconsistent "Chicken Breast" entry under "Chicken" to align with predefined categories like 'Lunch/Snacks', 'Beverages', and others.
5. **Missing Values**: Addressed 52 missing values (18.21% of the data) in the nutritional content columns by filling them with category and serving size-specific averages.

These corrections ensure that the dataset is now complete and ready for exploratory analysis, setting the foundation for accurate predictive modeling of recipe popularity.

Exploratory Data Analysis: Visualising Relationships and Patterns



The bar chart compares the count of recipes categorized by the number of servings (1, 2, 4, and 6) and whether they are considered "high traffic" (True or False).

For all serving sizes shown, the count of high-traffic recipes (True, represented by the teal bars) is notably higher than the count of non-high-traffic recipes (False, represented by the purple bars).

The most significant difference in counts between high and non-high traffic recipes occurs for recipes with 4 servings, indicating this serving size has the highest proportion of popular recipes in this dataset.

Model Development

In the development of our predictive model, we followed these steps:

1. **Data Splitting**: We selected the relevant features (calories, carbohydrates, sugar, protein, servings, and category) and high_traffic as the target variable.
2. **Train-Test Split**: Using `train_test_split`, we divided the data into training and testing sets for model evaluation.
3. **Model Development**: We created baseline and comparison models, training them on the data and making predictions.
4. **Model Evaluation**: We evaluated the performance of several models, with a focus on **Logistic Regression** and **Decision Tree**.

Model Evaluation - Logistic Regression & Decision Tree

- The **Logistic Regression** model performed well with higher precision and accuracy (0.80 and 0.76).
- The **Decision Tree** showed good accuracy but had a lower precision (0.72) compared to Logistic Regression.

Business Metrics

The **Logistic Regression** model emerged as the most effective for predicting high-traffic recipes, achieving **80% accuracy** and prioritizing **precision** to minimize false negatives. It performed well in both **training** and **testing**, with a **High Traffic Conversion Rate** of **4.09** for training and **4.0** for testing, making it reliable for deployment.

In contrast, the **Decision Tree** model showed signs of **overfitting**, with a **KPI of 230.0**, poor generalization, and a **High Traffic Conversion Rate** of **459.0** for training and **2.48** for testing, which made it unsuitable for deployment.

Thus, **Logistic Regression** was chosen for its **robustness** and **business relevance**, while the **Decision Tree** was discarded due to its overfitting issues.