05.2025

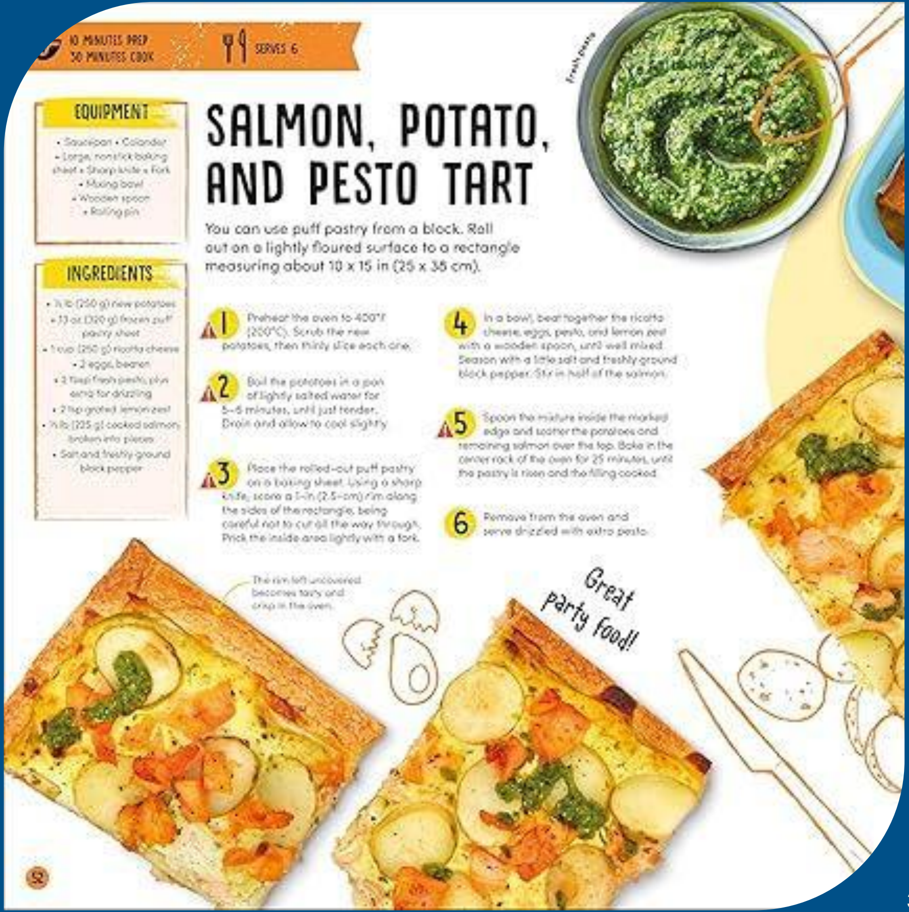
Recipe Site Traffic Prediction

Author: Son Hai Le

1. Project Overview
2. Method
3. Key Findings
4. Summary & Recommendations

1. Project Overview

- Background
- Business Goals



1. Project Overview

→ **Background**

- Homepage recipes can boost total site traffic by up to 40% if popular.

- Recipe currently selected based on personal preference.

- Increased traffic = more subscriptions → high business impact.

→ **Business Goals**

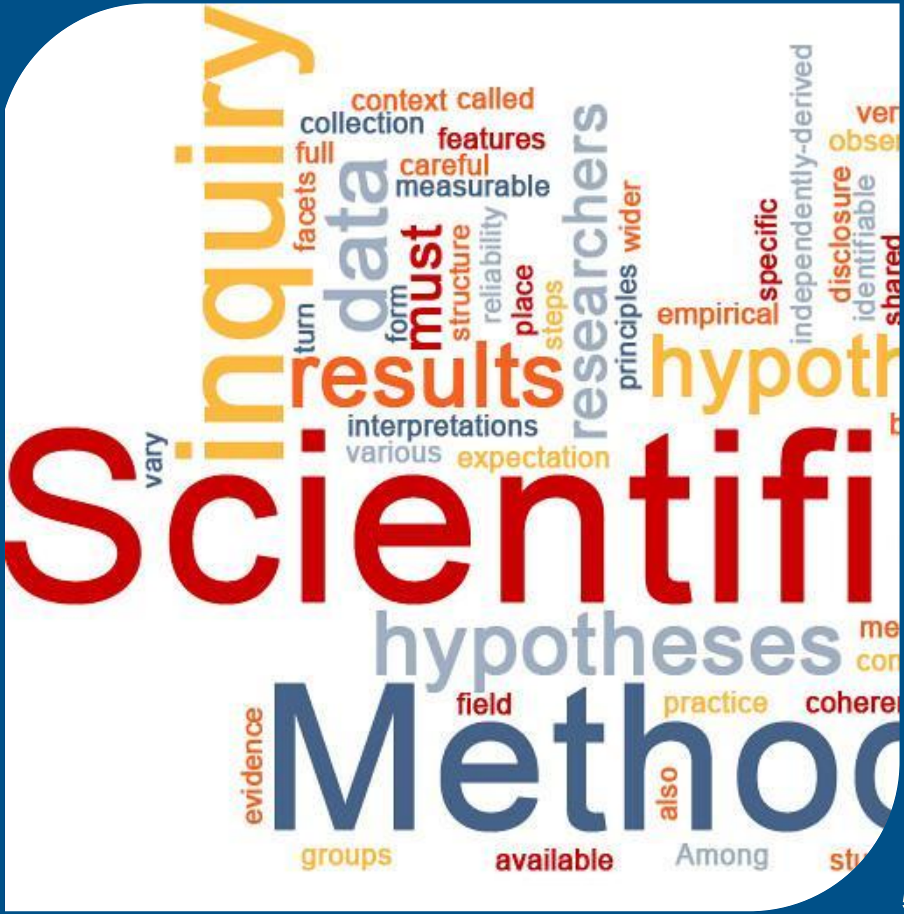
- Predict which recipes will lead to high traffic.

- Achieve 80+% accuracy in identifying popular recipes.

- Provide data-driven recommendations for homepage selection

2. Method

- Procedure
- Metrics Selection



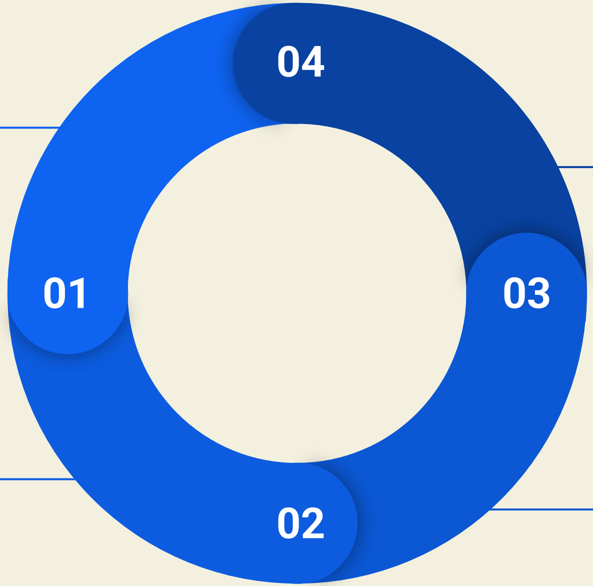
Procedure

Data Validation

1. Validation
2. Cleaning

Exploratory Data Analysis (EDA)

1. Single variable analysis
2. Multiple variable analysis



Model Evaluation

1. Model evaluations on a specific metric

Model Development

1. Data Preprocessing
2. Models Construction and Comparison

Metrics Selection

🎯 **Accuracy = (TP + TN) / (TP + TN + FP + FN)**

- Overall correctness of the model.
- Can be misleading if popular vs. unpopular recipes are imbalanced.

✅ **Precision = TP / (TP + FP)**

- Of all recipes predicted *popular*, how many actually were?
- High precision = fewer wrongly promoted recipes (i.e., False Positive).

📢 **Recall = TP / (TP + FN)**

- Of all truly *popular* recipes, how many did we catch?
- High recall = fewer missed opportunities.

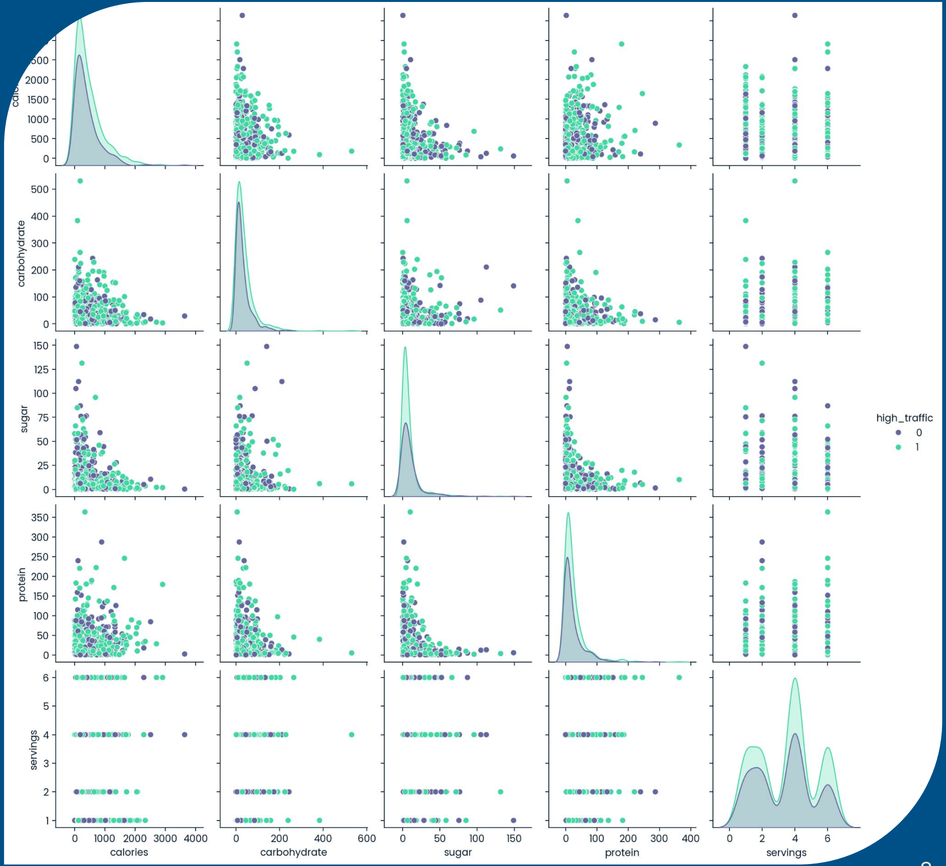
⚖️ **F1-Score = 2 * (precision * recall) / (precision + recall)**

- Balance between precision and recall.
- Useful when having imbalance data.

→ Accuracy and precision: for models comparison
→ F1-score: for hyperparameter tuning

3. Results

- Data Validation
- Exploratory Data Analysis (EDA)
- Data Preprocessing
- Models Construction and Comparison
- Key Findings

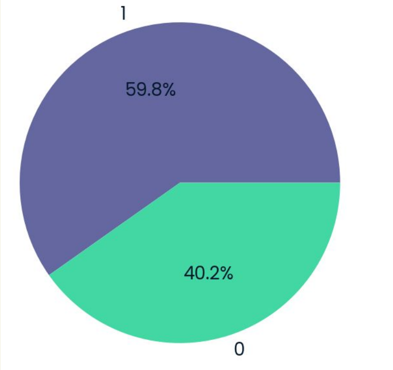


Data Validation

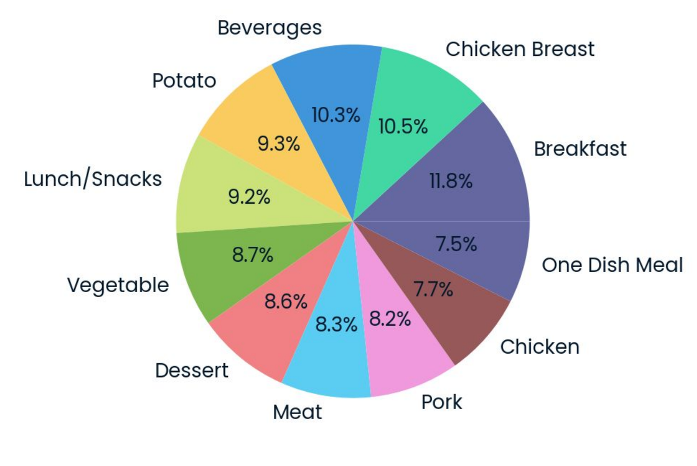
Column	Non-Null Count	Dtype	Clean steps
recipe	947	int64	Remove
calories	895	float64	Remove rows w/ nulls (5.5%)
carbohydrate	895	float64	
sugar	895	float64	
protein	895	float64	
category	947	object	-
servings	947	object	Clean, convert to `int`
high_traffic	574	object	Replace nulls by 0, `High` by 1

→ Before cleaning: 947 rows; After cleaning: 895 rows (5.5%↓)

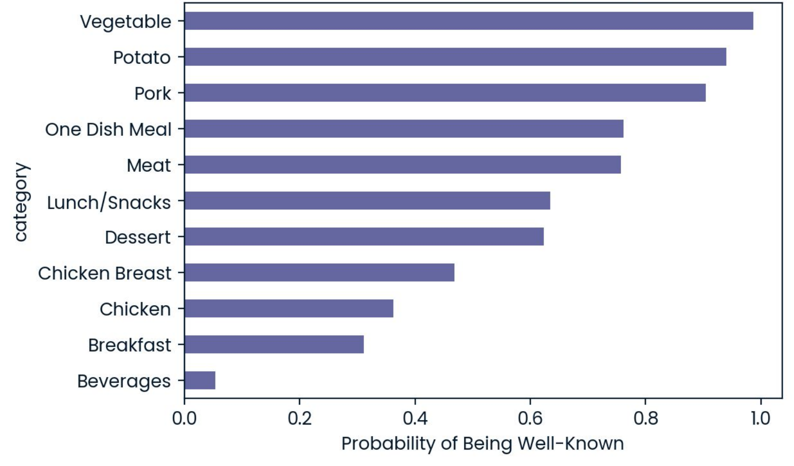
EDA [1/3]



High traffic (1) /non-high traffic (0) recipe proportions



Food categories proportions



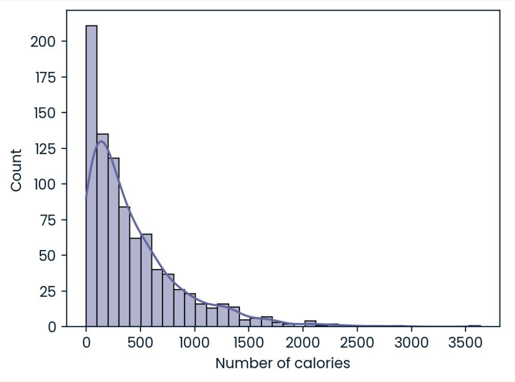
Probability of each category to be well-known

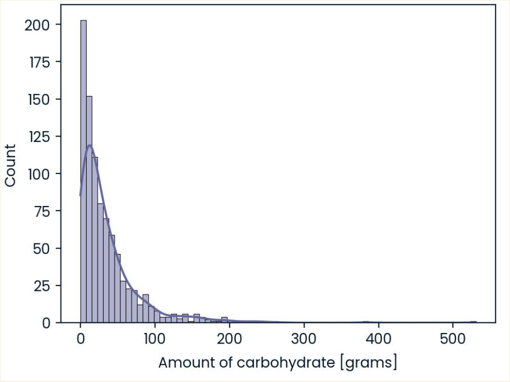
→ Target classes: balanced, might not need additional techniques

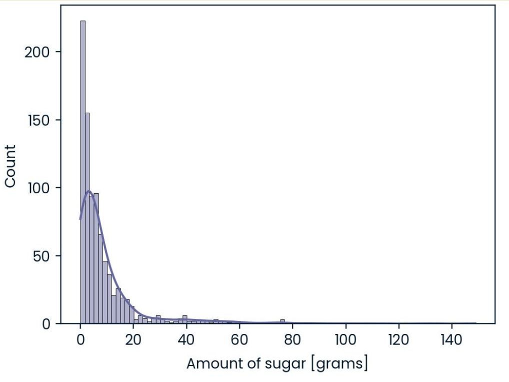
→ `Category` feature: balanced between each categories, not need combine rare ones

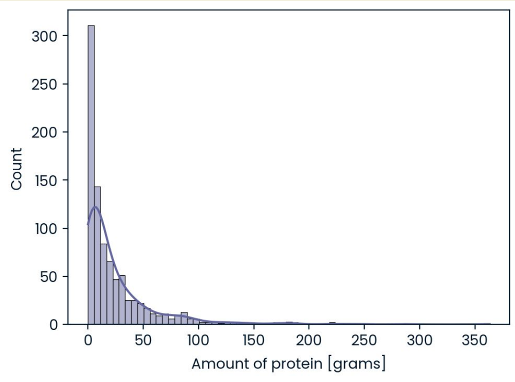
→ `Vegetable`, `Potato` and `Pork`: have highest chances to be popular

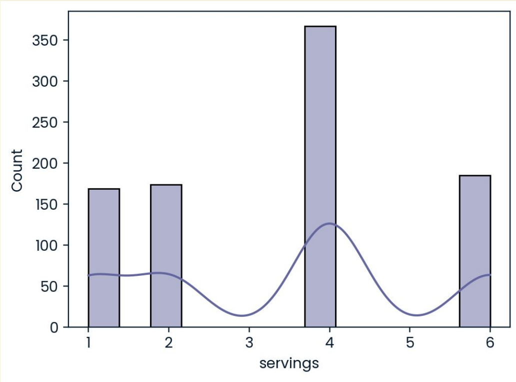
EDA [2/3]







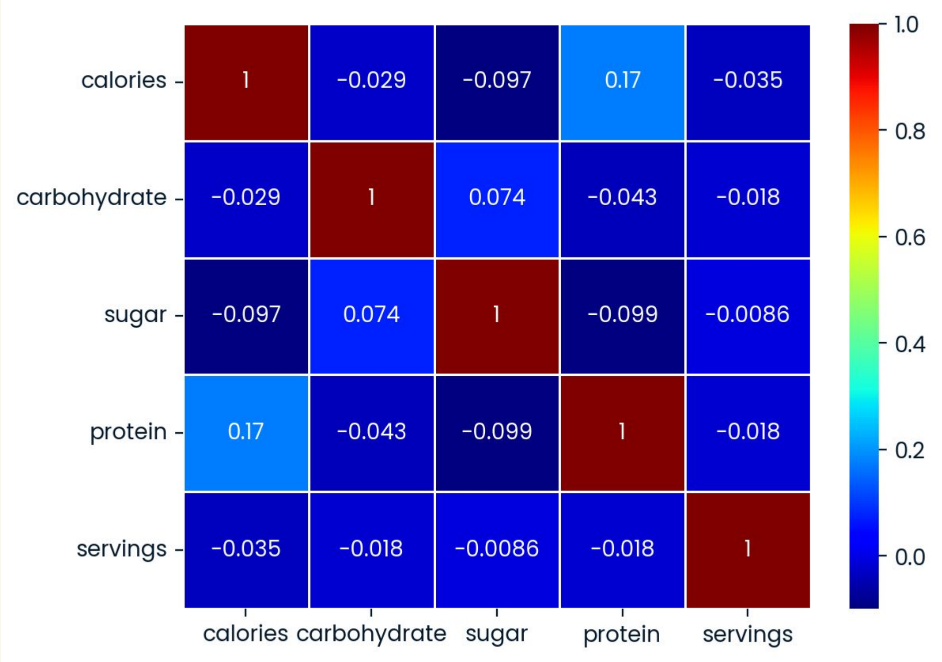




Distributions of different numeric input features

→ All features except `servings`: have long tail distributions
→ Will need a suitable transformation before modeling

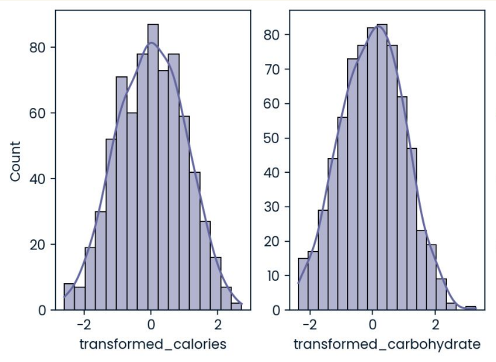
EDA [3/3]

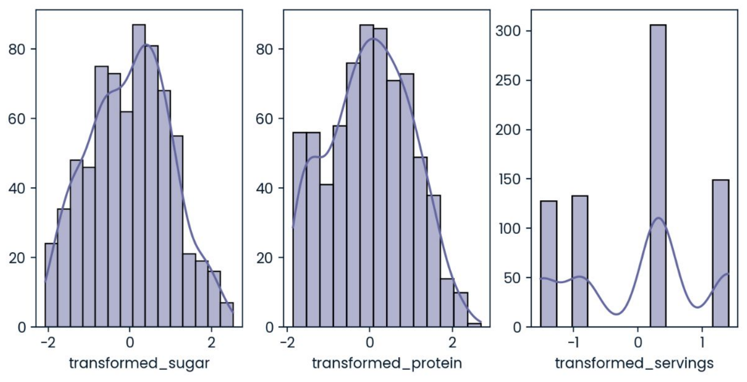


→ No strong correlations between numeric input features
→ Keep all these features for modelling

Data Preprocessing

<table>
<tr><th>Column</th><th>Step 1</th><th>Step 2</th></tr>
<tr><td>calories</td><td rowspan="5">Power Transform</td><td rowspan="6">PCA</td></tr>
<tr><td>carbohydrate</td></tr>
<tr><td>sugar</td></tr>
<tr><td>protein</td></tr>
<tr><td>servings</td></tr>
<tr><td>category</td><td>One hot encoder</td></tr>
</table>





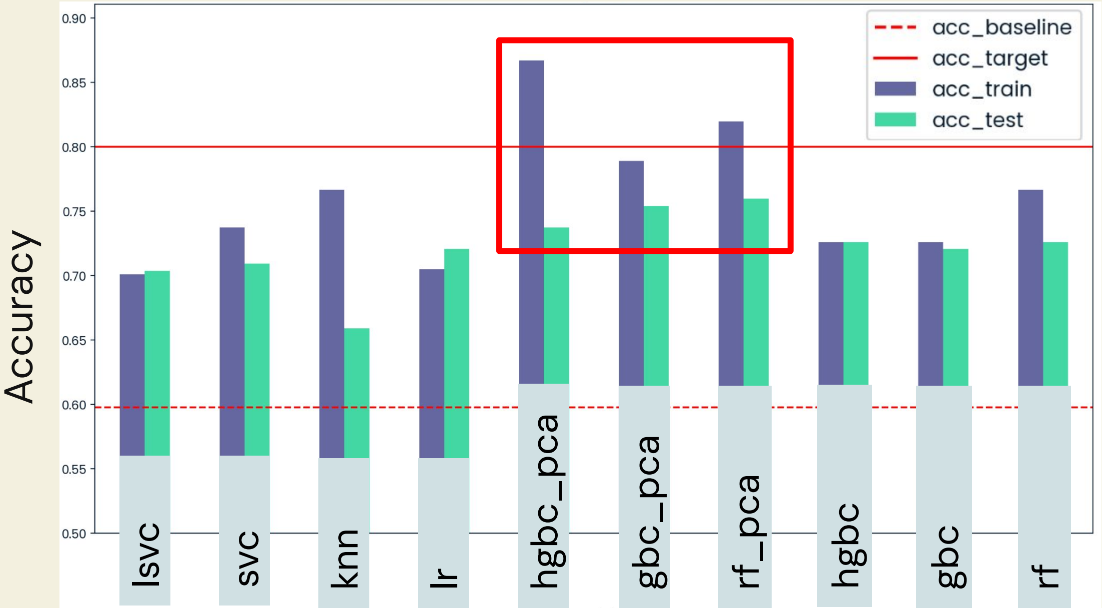
- Power transform → Standardization
- One hot encoder → categorical variables into a binary format
- PCA → Dimension reduction

Models Construction

Model Type	Model	With PCA
Naive	Baseline	**-**
Linear	`LinearSVC`	Yes
	`LogisticRegression`	
Non-linear	`SVC`	
	`KNeighborsClassifier`	
Ensemble	`RandomForestClassifier`	Yes and No
	`GradientBoostingClassifier`	
	`HistGradientBoostingClassifier`	

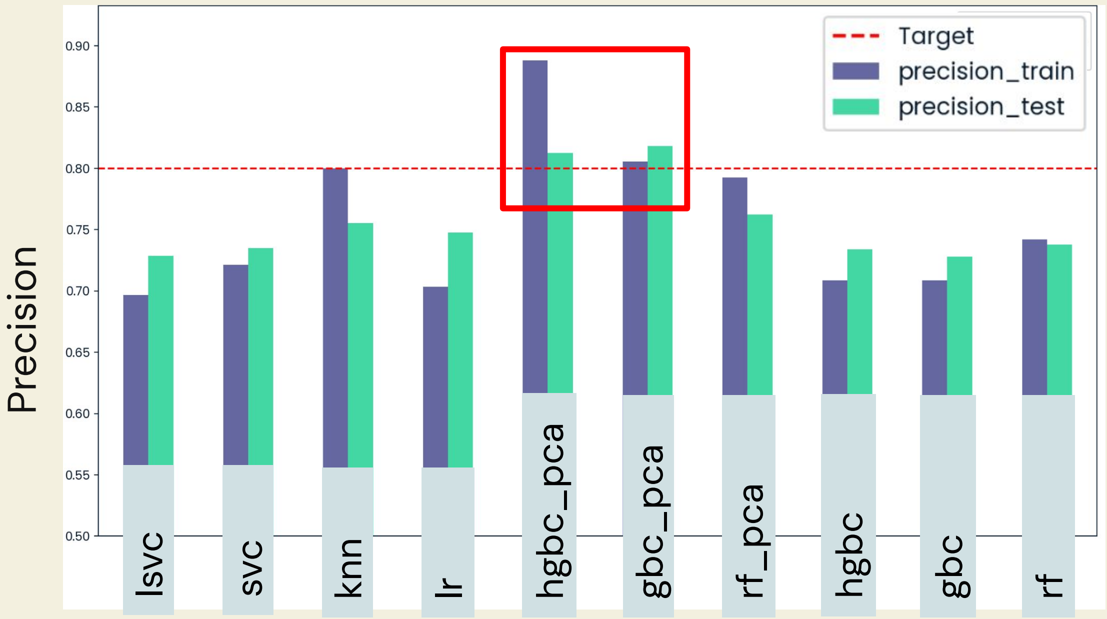
→ Apply Random Search for Models' hyperparameters and Number of PCA's components

Models Comparison



→ All models: better than the baseline one
→ Ensemble models w/ PCA: the best ones
→ No models achieve 80% accuracy on the test dataset

Models Comparison



→ Gradient boosting and Histogram gradient boosting w/ PCA: achieve 80% accuracy on the number of positive predictions

4. Key Findings

4. Key Findings

- Top performers: `rf_pca`, `gbc_pca`, and `hgbc_pca` approach or exceed the target
- Overfitting: `rf_pca` and `hgbc_pca` show train-test gaps, suggesting overfitting.
- Underperformers: `knn`, `lsvc`, and `lr` perform just above baseline

→ PCA-enhanced ensemble models generalize best

→ `gbc_pca` stands out as the most balanced and robust across train and test sets.

5. Recommendations

5. Recommendations

✅ **Deploy** `gbc_pca (i.e., gradient boosting classifier enhanced w/ PCA)`

- Meets/exceeds 0.80 precision target
- Low overfitting (small train-test gap)
- Strong and stable test accuracy

🔧 **Next Steps**

- **Threshold Tuning**: Use precision-recall curve to reduce false positives
- **Post-Prediction Check**: Add business rules (e.g., ingredient trends, seasonality)
- **Monitoring**: Regular evaluation & retraining as user behavior shifts
- **PCA**: Keep in pipeline — it's boosting precision, but monitor over time