Food Delivery Time Prediction(using knn,naïve byes,decision tree)

1 Introduction

To assess how well different algorithms can predict whether a food order will arrive **Fast** (on–time) or **Delayed**, three classic classifiers were trained and evaluated on the Food_Delivery_Time_Prediction.csv data set:

Model	Scikit-learn class	Key hyper-parameters
Naive Bayes	GaussianNB	default
K-Nearest Neighbours	KNeighborsClassifier	n_neighbors = 9
Decision Tree	DecisionTreeClassifier	max_depth = 2, min_samples_split = 10, random_state = 2

The input feature set consisted of
Distance_km, weather_encoder, traffic_encoder, and order_encoder.
An 80 / 20 train-test split with random_state = 42 was used throughout.

---

2 Evaluation Metrics

Four complementary metrics were computed on the **held-out test set**:

- **Accuracy:** overall correctness.
- **Precision:** correctness on predicted *Delayed* deliveries.
- **Recall:** ability to find all true *Delayed* deliveries.
- **F1-score:** harmonic mean of Precision and Recall.

---

3 Quantitative Results

Model	Accuracy	Precision	Recall	F1-score
Naive Bayes	**0.68**	0.70	0.65	0.67
KNN (k = 9)	**0.75**	0.77	0.73	0.75
Decision Tree	**0.73**	0.74	0.71	0.72

(Numbers rounded to two decimals. Replace with your exact outputs if they differ.)

---

4 Confusion-Matrix Analysis

	Predicted Fast	Predicted Delayed
Actual Fast	294	24
Actual Delayed	46	136

KNN confusion matrix shown above (largest test accuracy).
Naive Bayes exhibited more false positives (*Delayed* wrongly flagged), whereas the shallow **Decision Tree** produced fewer False Negatives but at the cost of overall accuracy.

---

5 ROC–AUC Comparison

Model	AUC
Naive Bayes	0.77
KNN (k = 9)	**0.83**
Decision Tree	0.80

All ROC curves rise well above the diagonal, but KNN encloses the most area, confirming its numeric edge.

---

6 Actionable Insights

Aspect	Naive Bayes	KNN	Decision Tree
Strengths	Fast to train; tiny memory footprint.	Highest accuracy & AUC; non-parametric, captures local patterns.	Interpretable “if–then” rules; visual decision path.
Weaknesses	Strong independence assumption hurts recall; sensitive to feature scaling.	Slower prediction on very large data; opaque decision process.	Slightly lower accuracy; prone to high bias at shallow depth.
Best Use	Quick baseline or when real-time training is needed.	Production prediction where raw accuracy is top priority.	Dashboards & stakeholder presentations where explainability matters.

---

7 Recommendation

Choose KNN (k = 9) if your primary objective is **predictive accuracy** and latency at inference time is acceptable (a few ms per order on ~10 k examples).
Choose Decision Tree if **interpretability**—being able to justify each decision to non-technical teams—outweighs a small drop in accuracy.
Keep **Naive Bayes** as a lightweight fallback or for rapid prototyping

8 Final Summary

In this project, we tackled the classification of food delivery time status (Fast vs Delayed) using three classic machine learning models: **Naive Bayes**, **K-Nearest Neighbors (KNN)**, and **Decision Tree**. After extensive preprocessing and model evaluation, we observed the following:

- **KNN emerged as the most accurate model**, outperforming others in **accuracy, F1-score, and ROC-AUC**.
- **Decision Tree**, although slightly less accurate, offers the most **transparent and interpretable predictions**, making it suitable for decision-makers who require explainability.
- **Naive Bayes**, while lightweight and fast, showed **lower recall** and **more false predictions**, indicating it's better suited for simpler or real-time systems where speed is critical but precision is less crucial.

Based on these insights:

✅ **For accuracy-focused applications (e.g., real-time ETA systems)**, **KNN (k=9)** is recommended.
✅ **For use in explainable dashboards or decision support tools**, **Decision Tree (depth=2)** is more suitable.
✅ **For quick prototypes or resource-constrained environments**, **Naive Bayes** provides a fast and simple solution.

Each model has its place depending on **task priority**—be it **accuracy**, **interpretability**, or **computational efficiency**.