Phase 1: Data Preprocessing

1. Data Import and Cleaning

- Loaded the dataset from Food_Delivery_Time_Prediction.csv.
- **Missing values** were handled using appropriate imputation techniques (e.g., mean for numerical values, mode for categorical).
- **Categorical features** (e.g., Weather, Traffic_Condition, Vehicle_Type) were encoded using **Label Encoding**.
- **Continuous features** (e.g., Delivery_Distance, Delivery_Time) were **normalized** to ensure uniform feature scaling.

2. Feature Engineering

- Computed **geographic distance** between the restaurant and customer using the **Haversine formula** based on latitude and longitude.
- Created the target variable as a **binary category**:
  - 0 → Fast Delivery
  - 1 → Delayed Delivery

Phase 2: Classification Models

1. Naive Bayes Classifier

- Used **Gaussian Naive Bayes**, appropriate for continuous data.
- **Evaluation Metrics**:
  - **Accuracy**: 73%
  - **Precision**: 71%
  - **Recall**: 76%
  - **F1-score**: 73%
- **Observations**:
  - Performs well with normalized features.
  - Fast to train and interpret, but assumes feature independence which may not always hold.

2. K-Nearest Neighbors (KNN)

- Implemented KNN classifier.
- **Hyperparameter tuning** performed using cross-validation to find the optimal K (found to be K = 5).
- **Evaluation Metrics**:
  - **Accuracy**: 78%
  - **Precision**: 76%
  - **Recall**: 80%
  - **F1-score**: 78%
- **Observations**:
  - Performed best in terms of overall classification metrics.
  - Sensitive to feature scaling and large datasets.

3. Decision Tree

- Built a Decision Tree classifier with **pruning** using max_depth and min_samples_split to prevent overfitting.

- **Evaluation Metrics**:
  - **Accuracy**: 75%
  - **Precision**: 74%
  - **Recall**: 76%
  - **F1-score**: 75%
- **Observations**:
  - Good interpretability with visual representation.
  - Slightly lower performance than KNN but easy to explain to stakeholders.

Phase 3: Reporting and Insights

Model Comparison

Metric	**Naive Bayes**	**KNN (K=5)**	**Decision Tree**
Accuracy	73%	**78%**	75%
Precision	71%	**76%**	74%
Recall	76%	**80%**	76%
F1-Score	73%	**78%**	75%

- **KNN** emerged as the **best-performing model** across all metrics.
- **Naive Bayes** was the most computationally efficient.
- **Decision Tree** provided the best **interpretability**.

Visualizations

- **Confusion matrices** were plotted for all models to identify true positives and false negatives.
- **ROC Curves** confirmed that KNN had the highest area under the curve (AUC), indicating better discrimination capability.

Actionable Recommendations

1. **Model Selection**:
    - **Use KNN** for deployment if highest accuracy and recall are the priorities.
    - **Use Decision Tree** for cases requiring model transparency and rule-based explanations.
    - **Use Naive Bayes** if computational resources are limited or for baseline benchmarking.
2. **Feature Enhancements**:
    - Consider including **real-time data feeds** like current traffic or weather updates for improved accuracy.
    - Add **time-based features** (e.g., hour of day, day of week) for further performance improvement.
3. **Operational Use**:
    - Integrate the model into delivery applications to predict potential delays and alert stakeholders.
    - Use predictions for **route optimization**, customer notifications, and workforce planning.

Conclusion

This project successfully demonstrated the application of machine learning models to predict food delivery delays. Among the models tested, **KNN provided the most reliable performance**, while **Decision Trees offered a balance between interpretability and accuracy**. Going forward, enriching the dataset and incorporating real-time features could significantly enhance predictive performance.