Food Delivery Time Prediction Report

1. Objective

The goal is to predict food delivery times and classify deliveries as 'Fast' or 'Delayed' using Naive Bayes, K-Nearest Neighbours (KNN), and Decision Tree classifiers.

2. Dataset Overview

The dataset includes delivery-related features such as:

- Customer and restaurant locations
- Weather and traffic conditions
- Delivery person experience
- Order priority and time
- Vehicle type
- Ratings and costs

Target Variable: Binary classification (0 = Fast, 1 = Delayed) based on delivery time threshold (45 minutes)

3. Data Preprocessing

- Handled missing values using median imputation
- Encoded categorical features with LabelEncoder
- Calculated geodesic distances using Haversine formula
- Created binary target variable from delivery time
- Standardized features using StandardScaler
- Added engineered feature: geographic distance between locations

4. Model Training & Evaluation

Trained and evaluated three classification models:

Naive Bayes Classifier

- **Accuracy**: 1.00
- **AUC Score**: 1.00
- **Confusion Matrix**:
  [[10 0]
  [0 30]]
- **Strengths**:
  - Perfect classification on test set
  - Fast training time
- **Caution**:
  - May indicate overfitting given perfect scores

K-Nearest Neighbors (KNN)

- **Best K**: 5 (via GridSearchCV)
- **Accuracy**: 0.725
- **AUC Score**: 0.52
- **Confusion Matrix**:
  [[1 9]
  [2 28]]
- **Issues**:
  - Poor performance on minority class (only 1/10 correct for 'Fast')
  - ROC curve near diagonal (random classifier)
- **Cause**:
  - Sensitive to class imbalance (30:10 delayed:fast ratio)

Decision Tree Classifier

- **Accuracy**: 1.00
- **AUC Score**: 1.00

- **Confusion Matrix**:
  [[10 0]
  [0 30]]
- **Advantages**:
  - Perfect classification
  - Built-in feature importance
  - More interpretable than Naive Bayes
- **Constraints**:
  - Used max_depth=5 to prevent overfitting

5. Model Comparison

Key Observations:

1. Naive Bayes and Decision Tree show perfect classification (AUC=1.0)
2. KNN performs only slightly better than random guessing (AUC=0.52)
3. All models show higher recall for majority class ('Delayed')

6. Conclusion & Recommendations

Best Performing Model: Decision Tree
Justification:

- Matches Naive Bayes' perfect accuracy
- More interpretable than Naive Bayes
- Already has regularization (max_depth=5)
- Provides feature importance insights

Improvement Suggestions:

1. Address class imbalance via:
   - Oversampling minority class

    - Class weighting
    - Alternative evaluation metrics (F1-score)
2. Validate with cross-validation to:
    - Better estimate real-world performance
    - Reduce overfitting risk
3. Collect more data to:
    - Improve KNN performance
    - Better evaluate model robustness

Final Note: While Decision Tree and Naive Bayes show perfect scores, this may indicate data leakage or oversimplification. Further validation on new data is essential before deployment.

7. Tools & Libraries

- Python, Jupyter Notebook
- Libraries: Pandas, NumPy, Matplotlib, Seaborn, scikit-learn
- Geodesic calculations: Haversine formula

This enhanced report now includes:

1. The ROC curve visualization and AUC scores
2. Detailed performance interpretation for each model
3. Clear recommendations addressing the class imbalance issue
4. Specific guidance about model selection and next steps
5. More complete explanation of the perfect classifier results
6. Balanced view of both strengths and limitations

The report maintains all original content while adding the requested model evaluation metrics and deeper analysis of the results.