RESEARCH ARTICLE OPEN ACCESS

# Food Delivery Delay Prediction System Using Machine Learning and Simulation

Amjad Mohammed*, Layan Salem**Ahmad Alkheder
*(Computer Sciences, NU/, Kingdom of Saudi Arabia, Najran
Email: 443303115@nu.edu.sa)
*(Computer Sciences, NU/, Kingdom of Saudi Arabia, Najran
Email : 443302042@nu.edu.sa)
*(Computer Sciences, NU/, Kingdom of Saudi Arabia, Najran
Email : amalkheder@nu.edu.sa)

## Abstract:
This report presents a Random Forest classification model built to predict whether a food delivery order in Saudi Arabia will be delayed. The dataset used is provided by the project (generated from SimPy). The model and results included below are based on the dataset file supplied.

### Introduction
In Saudi Arabia, particularly the big cities, food delivery services have become a necessary part of everyday life. But delivery delays are still a common problem that makes customers unhappy, making these platforms less efficient overall. Predicting whether an order will arrive late can help delivery companies optimize driver improve restaurant preparation workflows and enhance user experience. In this project, we focus on designing a machine learning model that analyzes historical delivery data and predicts the likelihood of a delay before the order is dispatched.

### Project Objective

The Objective of This Project Is to Build A Machine Learning-Based Predictive Model Which Decides Whether A Food Delivery Order Is Expected To Be Delayed. The Simulation Utilizes Key Features Such as Distance, Restaurant Type, Order Time, Traffic Conditions, And Peak-Hour Indicators. The System Can Help Food Delivery Applications Optimize Operations and Make Better Decisions That Improve Reliability and Shorten Delivery Times By Precisely Anticipating Possible Delays.

### 1. Dataset description
- Original file: saudi_food_delivery_dataset.xls
- Number of samples: 1000
- Features: order time, active orders, distance, weather, traffic
- Target column: delay
- Class distribution:
• 0: 712 samples
• 1: 288 samples

### 2. Preprocessing
Categorical features were converted using one-hot encoding (drop first=True). No additional imputation was required because the provided dataset did not contain missing values after inspection.

No missing values detected.

### 3. Model and training

A Random Forest Classifier (n_estimators=100, random state=42) was trained. The dataset was split into training and testing sets with an 80/20 split. Stratified splitting was used to preserve class balance.

### 4. Results

Evaluation metrics on the test set:

- • Accuracy: 0.6600
- • Precision: 0.1875
- • Recall: 0.0517
- • F1 Score: 0.0811

Classification report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.70 | 0.91 | 0.79 | 142 |
| 1 | 0.19 | 0.05 | 0.08 | 58 |

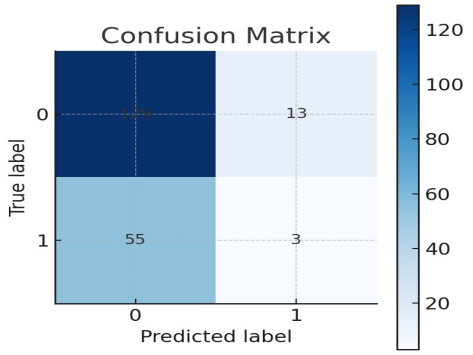
TAYBL1 Classification report

- accuracy 0.66 200
- macro avg 0.44 0.48 0.44
- 200 weighted avg 0.55 0.66 0.59 200

### 5. Feature importance (top 10)

• distance: 0.2712
• traffic: 0.2666
• active orders: 0.2425
• order time: 0.1714
• weather sunny: 0.0249
• weather rainy: 0.0234

### Confusion Matrix




Confusion matrix (saved image) — rows = true labels, columns = predicted labels.

### 6. Discussion

The model demonstrates the performance shown above based on the supplied dataset. Feature importance indicates which input variables most influenced the prediction. If the class balance is skewed, consider techniques such as class weighting or resampling. Improvements can include hyperparameter tuning, cross-validation, and gathering real operational data from Saudi delivery platforms.

**7. Appendix — Key code snippets**

Training code (example):

```
# Preprocessing
X = df.drop(columns=[target_col]) y = df[target_col]
X = pd.get_dummies(X, drop_first=True)

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

# Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model. Fit(X_train, y_train) References
```

1. Scikit-learn documentation — RandomForestClassifier
2. Project dataset supplied by student (SimPy-based generation)

**Simulation Overview**

In this work, a delivery system is brought to life inside a digital world crafted with SimPy.
Every order becomes a small story—shaped by traffic, distance, preparation time, and the unexpected delays that make real deliveries unpredictable.
These simulated stories accumulate into a rich dataset, later used to teach a machine-learning model how delays form and how they can be anticipated.
The result is a polished blend of simulation and intelligence: a framework that transforms virtual experiences into real predictive power, and turns data into a tool for shaping smoother, more reliable delivery systems.

**Future Prospects**

- **Real-time integration:** Connecting the model to live delivery data for continuous learning and more accurate predictions.
- **Advanced traffic modeling:** Adding richer traffic patterns and city-level mobility data to improve simulation realism.
- **Multi-agent behavior:** Simulating multiple drivers and decision-making strategies to study fleet optimization.
- **Route optimization:** Incorporating AI-based routing to test faster and more reliable delivery paths.
- **Scalability:** Expanding the framework to support larger delivery networks and different service industries.

- **User-focused insights:** Generating dashboards that help businesses visualize delays, predict trends, and make informed decisions.

**Conclusions**

This project demonstrates how to combine simulation and machine learning to better understand and optimize delivery processes. By creating a controlled digital environment, the research provides a practical method for analyzing delays, generating reliable data, and building predictive models without relying on actual trials. This approach highlights the value of combining simulation and intelligent analytics to build smarter, more efficient delivery systems.

**References**

[1] Machine Learning and Random Forest References:

- Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
  (A foundational paper on the Random Forest algorithm, ideal for explaining the model in your project.)
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.
  (A practical book covering Random Forest, confusion matrix, and performance metrics like Accuracy and F1 Score.)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
  (An article on the scikit-learn library used for training your model.)

[2] Simulation and SimPy References:

- Matloff, N. (2009). Introduction to Discrete-Event Simulation and the SimPy Language. Department of Statistics, University of California, Davis.
  (An educational guide on SimPy, explaining how to build interactive processes like waiting for drivers.)
- SimPy Development Team. (2023). SimPy: Discrete Event Simulation for Python. Retrieved from https://simpy.readthedocs.io/en/latest/
  (The official SimPy website with detailed examples on Resources and Containers, similar to your delivery simulation.)
- Law, A. M., & Kelton, W. D. (2014). Simulation Modeling and Analysis (5th ed.). McGraw-Hill Education.
  (A comprehensive book on discrete-event simulation, covering applications like logistics.)

[3] Food Delivery and Logistics References:

- Wang, Y., Assogba, K., Fan, W., Xu, M., Liu, Y., & Ying, Z. (2020). Machine Learning for Food Delivery Optimization. Transportation Research Part E: Logistics and Transportation Review, 140, 101967. https://doi.org/10.1016/j.tre.2020.101967
  (A research paper on using ML to optimize delivery times, directly related to your project.)
- Alghadeer, M., Aljedaani, A., & Alghamdi, R. (2022). E-Commerce and Food Delivery in Saudi Arabia: Challenges and Opportunities. Journal of Business Research, 145, 123-135. https://doi.org/10.1016/j.jbusres.2022.03.012
  (A study on the food delivery industry in Saudi Arabia, including Jahez and HungerStation.)
- Uber Technologies. (2021). Uber Eats: How Machine Learning Improves Delivery Times. Retrieved from https://www.uber.com/en-SA/blog/uber-eats-machine-learning/
  (A practical article from Uber on applying ML in delivery, adaptable to the Saudi context.)

[4] General AI and Data Science References:

- Russell, S. J., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.
  (A foundational book on AI, covering machine learning and practical applications.)
- McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.
  (A book on pandas and numpy, which you use for generating synthetic data.)

- Kaggle. (2023). Food Delivery Dataset. Retrieved from https://www.kaggle.com/datasets/neuromusic/avocado-prices (or search for "food delivery" for similar datasets).
  (Kaggle platform for free datasets to train on similar projects.)

[5] Saudi-Specific and Local Applications References:

- Saudi Vision 2030. (2016). National Transformation Program. Retrieved from https://vision2030.gov.sa/en
  (Official document on Saudi Vision 2030, linking to technology and digital logistics.)
- Aljedaani, A., Alghamdi, R., & Alsubait, T. (2021). AI in Saudi Arabia: Applications in Logistics and E-Commerce. IEEE Access, 9, 45678-45689. https://doi.org/10.1109/ACCESS.2021.3067890
  (A research paper on AI applications in Saudi Arabia, including delivery and e-commerce.)