# Data Science Project Report: Transport Demand Prediction for Mobiticket

**Author:** Jenkinson W

## 1. Executive Summary

This project was initiated to address a key business challenge for Mobiticket: predicting transport demand to optimize operations and revenue. By analyzing historical ticket data for 14 routes terminating in Nairobi, we developed a machine learning model to forecast the number of seats sold for any given ride.

Our analysis revealed that demand is heavily influenced by the **vehicle's capacity**, the **time of day**, and the **day of the week**, with significant travel peaks during early mornings and on Fridays and Sundays.

A **Random Forest Regressor** model was trained and evaluated, demonstrating solid predictive power. It explains **61% of the variance** in ticket sales, with a Mean Absolute Error (MAE) of approximately **3.2 seats**. The model's key takeaway is that operational factors (vehicle size, departure time) are the most significant drivers of sales.

**Key recommendations** include optimizing vehicle allocation based on predicted demand for specific times and routes, implementing dynamic pricing strategies for peak hours, and focusing marketing efforts on less popular travel times. This predictive model provides a foundational tool for data-driven decision-making to enhance Mobiticket's efficiency and profitability.

## 2. Business Understanding

### 2.1. Business Problem

Mobiticket, a transport service provider, sought to accurately forecast the number of seats sold for each of its rides. An inability to predict demand leads to two primary business problems:

- **Lost Revenue:** Under-predicting demand means not having enough capacity, leaving potential customers behind.
- **Increased Costs:** Over-predicting demand results in sending out vehicles with many empty seats, leading to inefficient fuel consumption and operational costs per passenger.

The goal of this project is to build a regression model that predicts the number of tickets sold for a specific route on a specific date and time, enabling Mobitoken to optimize resource allocation.

### 2.2. Context

The dataset covers 14 routes originating from towns northwest of Nairobi (e.g., Kisii, Homa Bay) and ending in Nairobi. These are long-haul journeys (8-9 hours) that are significantly impacted by Nairobi's traffic, which can add another 2-3 hours to the trip. Passenger behavior is likely influenced by a desire to avoid peak traffic hours, as well as by regular weekly work and travel patterns.

---

## 3. Exploratory Data Analysis (EDA) & Key Insights

The initial dataset contained records for each individual ticket sold. This was aggregated to create a dataset where each row represents a unique ride, with the target variable being `tickets_sold`.

**Key Visual Insights:**

- **Vehicle Type is Crucial:** A primary driver of sales is the vehicle type. Buses, with a `max_capacity` of 49, naturally sell far more tickets than shuttles, which have a capacity of 11. This was confirmed by the box plot analysis.
- **Demand is Time-Sensitive:** Ticket sales are not uniform throughout the day. The vast majority of departures occur in the morning (5 AM - 11 AM), with a smaller peak in the evening. This strongly suggests passengers schedule their travel to avoid arriving in Nairobi during peak afternoon traffic.
- **Weekly Travel Rhythms:** There is a clear weekly pattern in demand. Sales are highest on **Fridays** and **Sundays**, corresponding to typical weekend travel to and from Nairobi for work, school, or personal reasons.
- **Bimodal Sales Distribution:** The distribution of tickets sold per ride showed two peaks. This indicates that many trips are either nearly empty or almost full, suggesting a need for better demand matching.

---

## 4. Data Preparation

To prepare the data for modeling, the following steps were performed:

1. **Aggregation:** The dataset was grouped by `ride_id` to calculate the total `tickets_sold` for each unique ride.
2. **Feature Engineering:**

    - `travel_date` was used to extract `day_of_week`, `day_of_month`, `month`, and `year`.
    - `travel_time` was converted into `hour_of_day`.
3. **Categorical Encoding:** Non-numeric features like `travel_from`, `car_type`, and `day_of_week` were converted into numerical format using Label Encoding.
4. **Feature Selection:** The `ride_id` and `travel_to` columns (constant value 'Nairobi') were dropped as they are not predictive features.

---

## 5. Modeling

A **Random Forest Regressor** was selected as the modeling algorithm. This choice was made due to its robustness, ability to handle non-linear relationships, and its inherent resistance to overfitting.

**Snippet code**

**Define Features & Target:**

`X` = `df_agg` (excluding `tickets_sold`, `ride_id`, `travel_to`)

```
y = df_agg['tickets_sold']
```

**Split Data:**

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

**Train Model:**

```
model = RandomForestRegressor(n_estimators=100, random_state=42, n_jobs=-1)
```

```
model.fit(X_train, y_train)
```

**Predict:**

```
y_pred = model.predict(X_test)
```

**Evaluate Model:**

```
mae = mean_absolute_error(y_test, y_pred)
```

```
mse = mean_squared_error(y_test, y_pred)
```

```
r2 = r2_score(y_test, y_pred)
```

**Print Results:**

```
print(f"Mean Absolute Error (MAE): {mae}")

print(f"Mean Squared Error (MSE): {mse}")

print(f"R-squared: {r2}")
```

- **Features (X):** `travel_from`, `car_type`, `max_capacity`, `day_of_week`, `month`, `year`, `day_of_month`, `hour_of_day`.
- **Target (y):** `tickets_sold`.

The data was split into an 80% training set and a 20% testing set. The model was trained on the training data using 100 decision trees.

---

## 6. Evaluation

The model's performance was evaluated on the unseen test data with the following results:

- **R-squared (R²): 0.61**
  - *Interpretation:* The model successfully explains 61% of the variability in the number of tickets sold. This is a good performance, indicating a strong relationship between the features and the target.
- **Mean Absolute Error (MAE): 3.22**
  - *Interpretation:* On average, the model's prediction for the number of seats sold is off by approximately 3 seats.
- **Mean Squared Error (MSE): 20.57**

### Feature Importance

The model identified the following features as most important for prediction:

1. **`max_capacity`**: Overwhelmingly the most important factor.
2. **`hour_of_day`**: The second most influential predictor.
3. **`day_of_month`** & **`travel_from`**: Also significant contributors.

This confirms that operational decisions (which vehicle to use and when) are paramount in determining sales.

---

## 7. Recommendations & Actionable Insights

Based on the model and EDA, we recommend the following:

1. **Implement Dynamic Vehicle Allocation:**
   - **Action:** Use the model's predictions to assign buses (high capacity) to rides with high forecasted demand (e.g., Friday morning departures) and shuttles (low capacity) to rides with low predicted demand.
   - **Impact:** Reduce costs from running near-empty buses and prevent lost revenue from sold-out shuttles.
2. **Introduce Time-Based Pricing:**
   - **Action:** Implement a dynamic pricing strategy. Slightly increase ticket prices for peak travel times (e.g., morning and evening departures) and offer discounts for off-peak hours.
   - **Impact:** Maximize revenue during high-demand periods and stimulate demand during lulls.
3. **Target Marketing Efforts:**
   - **Action:** Launch marketing campaigns promoting travel on weekdays (Monday-Thursday), which currently have lower average demand. Highlight the benefits of a less crowded journey.
   - **Impact:** Smooth out demand across the week, leading to more consistent and predictable revenue streams.

---

## 8. Conclusion & Future Work

This project successfully developed a machine learning model that provides valuable predictions for transport demand. By leveraging this tool, Mobiticket can move from a reactive to a proactive operational strategy, optimizing vehicle allocation and pricing to improve profitability.

**Future Work:**

- **Hyperparameter Tuning:** Fine-tuning the model's parameters could yield further accuracy improvements.
- **Incorporate External Data:** Integrating data on public holidays, school calendars, and major local events would likely capture more demand fluctuations.
- **Explore Other Models:** Experimenting with gradient boosting models (like XGBoost or LightGBM) may offer enhanced performance.