Junior Data Scientist

Technical case presentation

16 July, 2025

LANGLOIS Camille
Paris Dauphine University - ENSAE
camille.langlois.contact@gmail.com
+33 6 02 34 57 31

BlaBlaCar

Presentation Overview

Context & objectives

BlaBlaCar is a ridesharing platform that connects drivers with empty seats to passengers.

Business Problem

- Many published rides end up with no passenger
- Matching is crucial to user satisfaction and retention
- Need to predict success (ride booked or not) at publication time

Objectives

- Analyze key success factors
- Build a predictive model (binary classification)
- Balance performance and inference speed
- Explore product use cases for deployment

This work is based on historical ride data provided by BlaBlaCar.

Dataset & definitions

Dataset overview

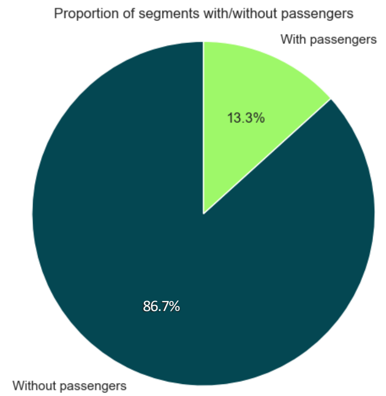
- ~3.5 million rides (in France and other countries),
- 20 features about the driver and the ride
- Target: whether the ride had at least one passenger
  → *Binary classification task: Success (1) vs Failure (0)*

Target definition

y = 1 → ride had ≥1 passenger
y = 0 → driver travelled alone (or cancelled)
Class imbalance: ~13% positives only

Key variables

Price: price per seat (€)
Seats: number of seats offered by the driver
Distance: length of the ride (km)
Departure and arrival coordinates
Date: date and time the trip will take place
Signup date: date the driver signed up on Blablacar
Success: if at least 1 seat has been confirmed



Data Cleaning & Feature Engineering

Dataset cleaning

Missing values
18 496 fixed_signup_country missing (also 'XX' value) → labeling 'missing' + grouping foreign countries

Inconsistencies removed
Distances < 0
More confirmed seats than offered, of 0 seat offered
Trip published after date of the segment

Outliers
Distances > 6000km removed

Useless features
Is_comfort adds no information

The classic IQR method was not used to handle outliers, as business considerations were deemed more relevant in this context.

New features

- Hours until departure
- Driver account age
- Number of driver trips
- Departure hour
- Day of the week
- Weekend (yes/no)
- Holiday (yes/no)
- Route popularity
- Departure location popularity
- Arrival location popularity
- Price per kilometer
- Long trip (yes/no)
- Price * popularity
- Seats * distance

Exploratory Data Analysis

BlaBlaCar

Factors impacting success

Based on visualizations, correlation matrix and effect size metrics

- Departure zone popularity
- Arrival zone popularity
- Trip length * number of seats
- Total trip distance
- Price per seat
- Price * route popularity
- Main ride segment

Zone popularity → departure and arrival areas matter

Trip characteristics → distance, price and number of seats impact success

Segment type → main segments tend to attract more passengers

Drivers departing from more popular zones are more likely to find passengers.

Modeling approach

Models implemented

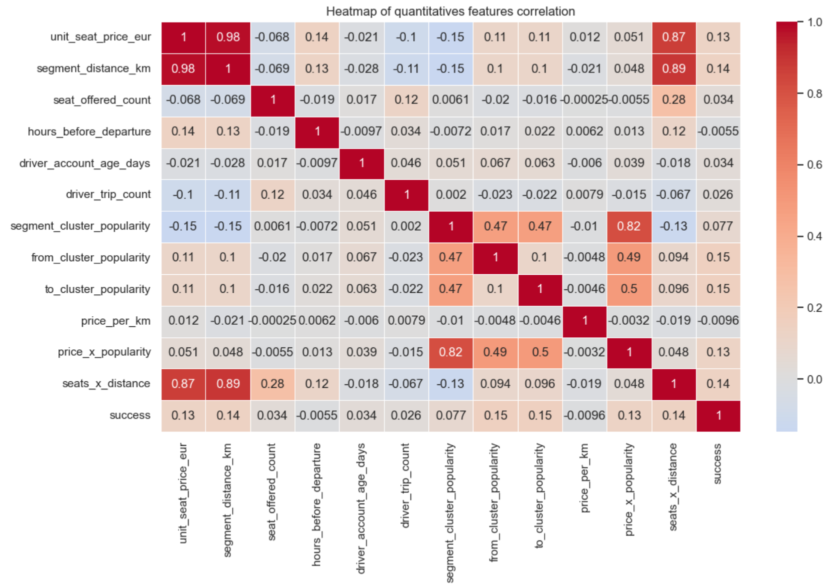
- **Logistic Regression** for its simplicity and interpretability,
- **Decision Tree** as a basic tree model,
- **Random Forest, LightGBM, XGBoost,** and **CatBoost** as more advanced ensemble methods,
- **Naive Bayes** for a probabilistic baseline,
- **LinearBoost**, an ensemble of linear models.

These included both linear and tree-based classifiers

Handling Class Imbalance

- **Class weights** were used for Logistic Regression, Decision Tree, CatBoost, Random Forest and LightGBM
- **Scale_pos_weight** parameter used with XGBoost

→ *Ensures that the models do not ignore the minority class (successful trips).*



Primary feature selection based on the exploratory analysis (correlation matrix, influent features) and business relevance.

Model performance comparison

Model	Precision	Recall	F1 Score	Inference time (s)
XGBoost	0.3081	0.7879	**0.4430**	0.1899
CatBoost	0.3026	0.7901	**0.4376**	0.0391
LightGBM	0.2940	0.7835	**0.4276**	0.5176
Random Forest	0.2646	0.7906	0.3965	3.0184
Decision Tree	0.2388	0.8206	0.3699	0.0348
Logistic Regression	0.2630	0.6787	0.3791	0.0349
Naive Bayes	0.3470	0.3094	0.3271	0.1315
LinearBoost	0.2350	0.4171	0.3007	0.3842

Why these metrics and inference time?

Precision, recall, and F1-score give a balanced view of model accuracy, especially with **imbalanced data**. **Inference time** is crucial for ensuring fast, real-time predictions in production. Balancing **performance** and **speed** is key for a good user experience.

- Trade-off between performance and inference speed
- Which model offers the best balance for production use?

→ XGBoost has been chosen got its best F1-Score and reasonable inference time

Model optimization

Hyperparameters tuning (Random Search)

- **Parameters tested:**
  - *n_estimators: [100, 200, 300]*
  - *max_depth: [3, 5, 7, 10]*
  - *learning_rate: [0.01, 0.1, 0.2]*
  - *subsample: [0.7, 0.8, 1]*
  - *colsample_bytree: [0.7, 0.8, 1]*
- **Method:** RandomizedSearchCV
- **Objective:** Maximize F1-score on validation set

Best hyperparameters

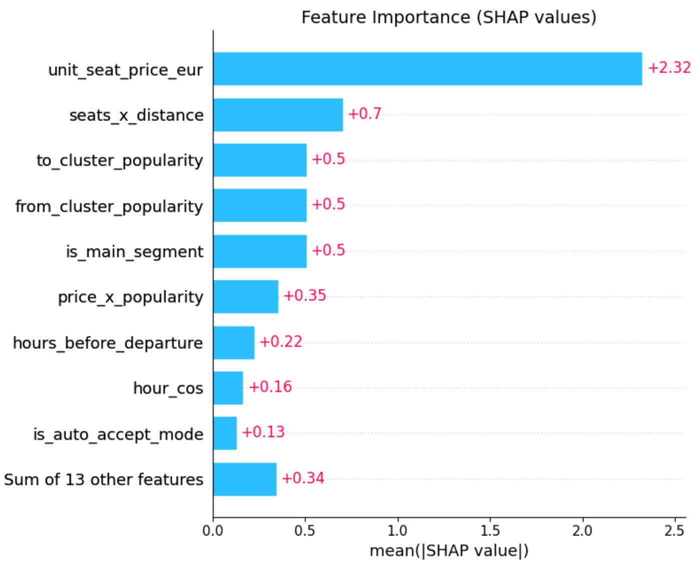
- *n_estimators:* ***300***
- *max_depth:* ***10***
- *learning_rate:* ***0.1***
- *subsample:* ***1***
- *colsample_bytree:* ***0.8***

Threshold Optimization

- Default threshold (0.5) not optimal for imbalanced data
- Tested multiple thresholds to maximize F1-score
- **Best threshold ≈ 0.656**

→ Final model chosen for balance between precision and recall

Feature importance

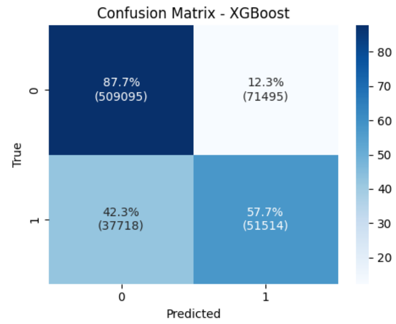


SHAP values explain how each feature contributes to the prediction (positive or negative impact on success probability).

→ Top 13 most important variables selected based on SHAP values to optimize performance and reduce complexity.

Final model performance

- ***Precision: 0.419***
- ***Recall: 0.577***
- ***F1-score: 0.485***
- ***Inference time: 0.899s***



Applications of the prediction model

Publishing Assistance
Real-time feedback helps drivers optimize ride details (price, time…) to increase booking chances.

Personalized Passenger Recommendations
Suggest to passengers only the rides with a high probability of success, increasing booking confidence and overall user satisfaction.

Operational Monitoring Dashboard
Enable targeted marketing by flagging at-risk rides and regions for proactive intervention.

Limitations & future improvements

BlaBlaCar

Current Limitations

- Limited features available (e.g., no driver ratings, weather or seasonality data).
- Simplified data cleaning and feature engineering steps
- Potential class imbalance impacting model robustness.
- Some ride details (like last-minute changes) not captured in the dataset

Next Steps

- Integrate additional external data (weather, holidays, user ratings).
- Explore advanced imbalance handling techniques.
- Develop a real-time scoring API for integration with the platform.
- Test model performance on live data and gather user feedback for continuous improvement.

Thank you!

Feel free to ask any questions.