

Will I get my packet on time?

Predictive Analytics and Machine Learning for Timely E-Commerce Deliveries



Introduction



► Relevance:

- Not needed, we all have been there

► Dataset Summary:

- Source: [Kaggle - DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS](#)
- 181,519 orders
- 53 features, including order date, client info, delivery details, product category, estimated shipping days, and **delay risk label**

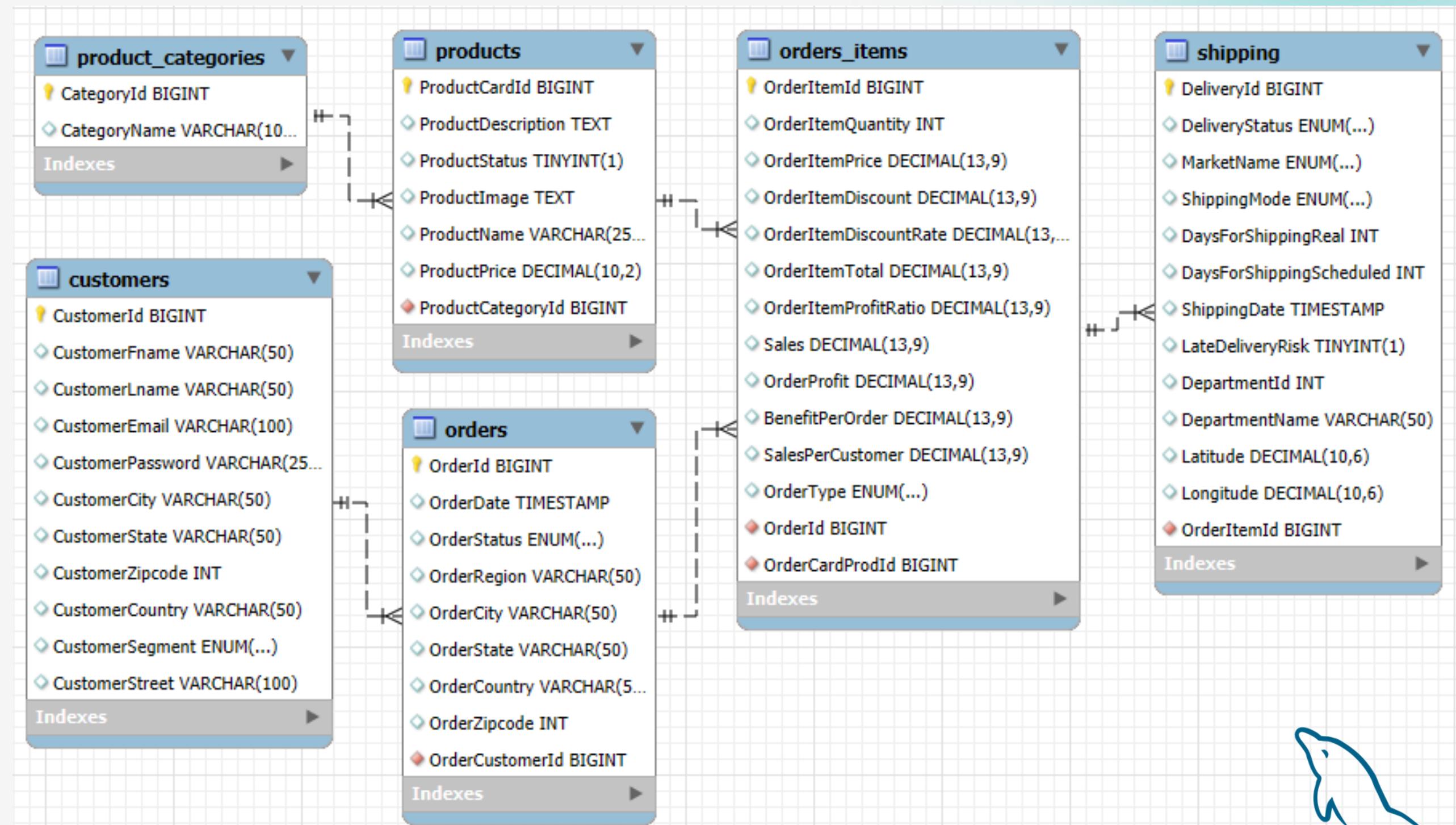
► Goal:

- Using machine learning to predict whether an order will be delivered on time or not.

Database

► Tables:

- Orders
- Order items
- Shipping
- Products
- Product categories
- Customers

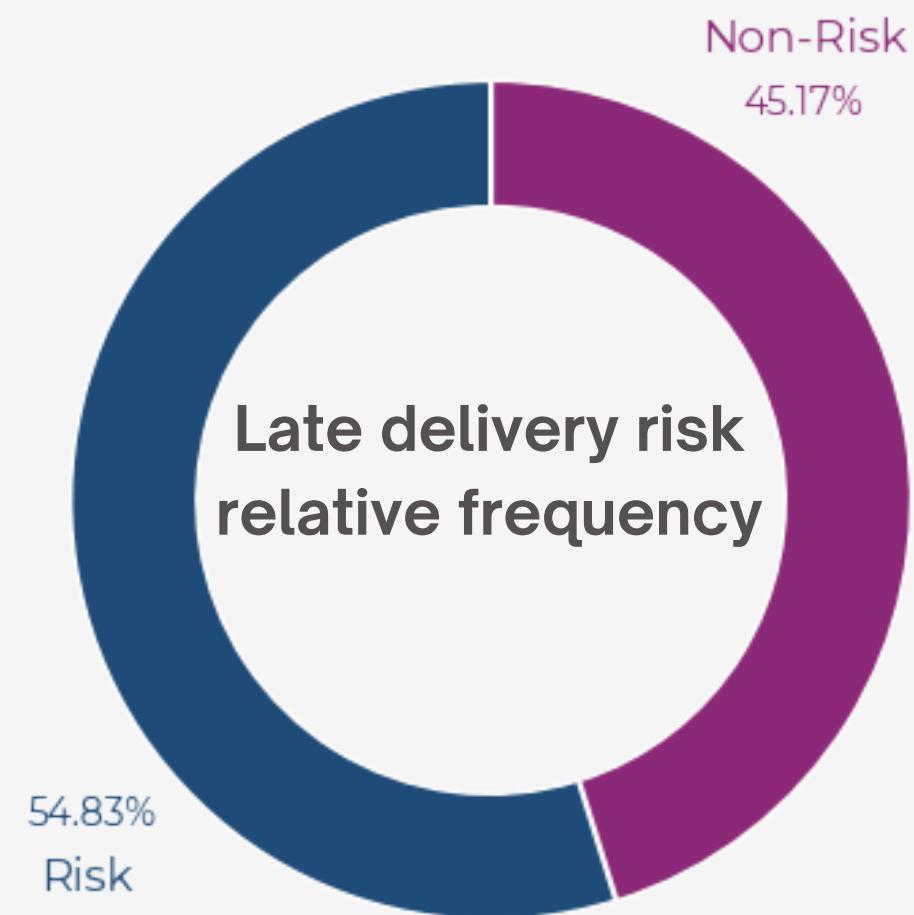


Data Cleaning

The dataset was mostly clean, but some columns had missing values that were needed to be handled

► Feature engineering

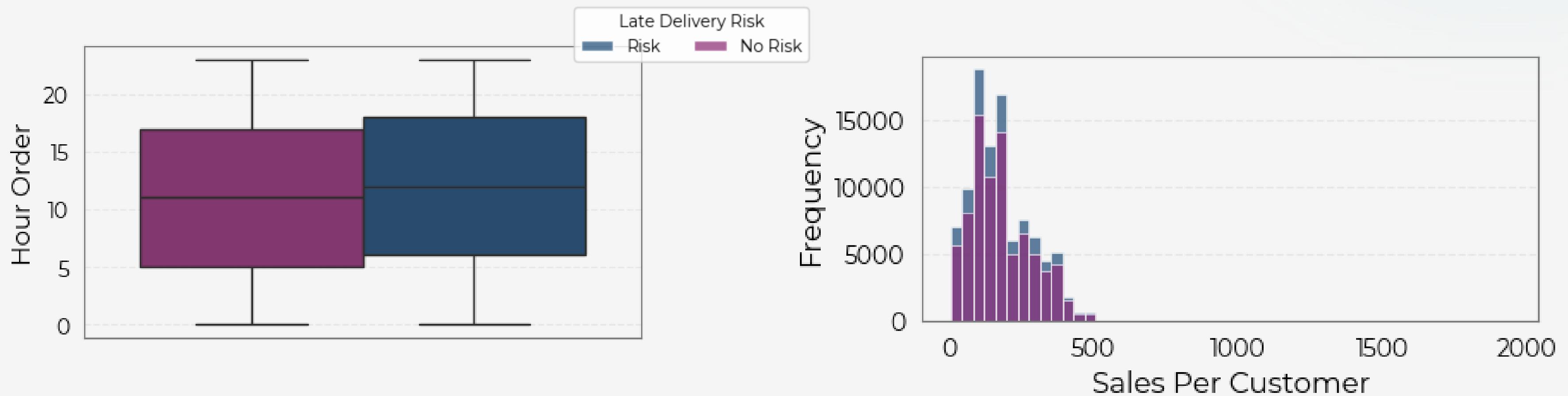
- Extracted date-based features from the order date.
- One-hot encoding of categorical data (e.g. shipping mode and delivery market)



For the ML models training:

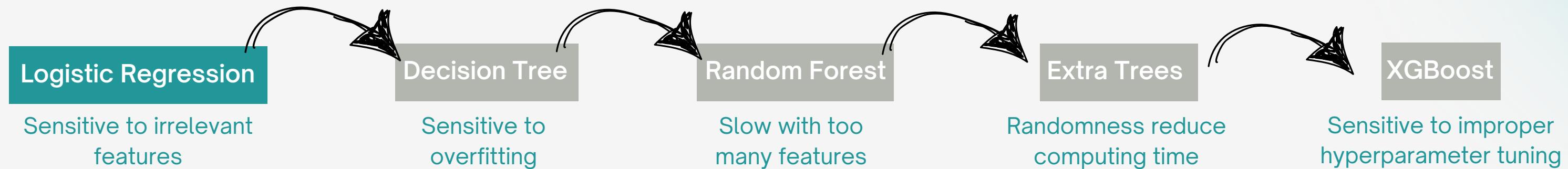
- 32 feature
- Maximum correlation coefficient with target: 0.4

Exploratory Data Analysis



EDA showed no distinct pattern between no-Risk and Risk transactions

Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.567** | Precision = 0.584 | Recall = 0.730 | F1-score = 0.649
✗ The model performs poorly.

Removing most irrelevant features according to correlation coefficient

Model Metrics: **Accuracy = 0.709** | Precision = 0.855 | Recall = 0.565 | F1-score = 0.680

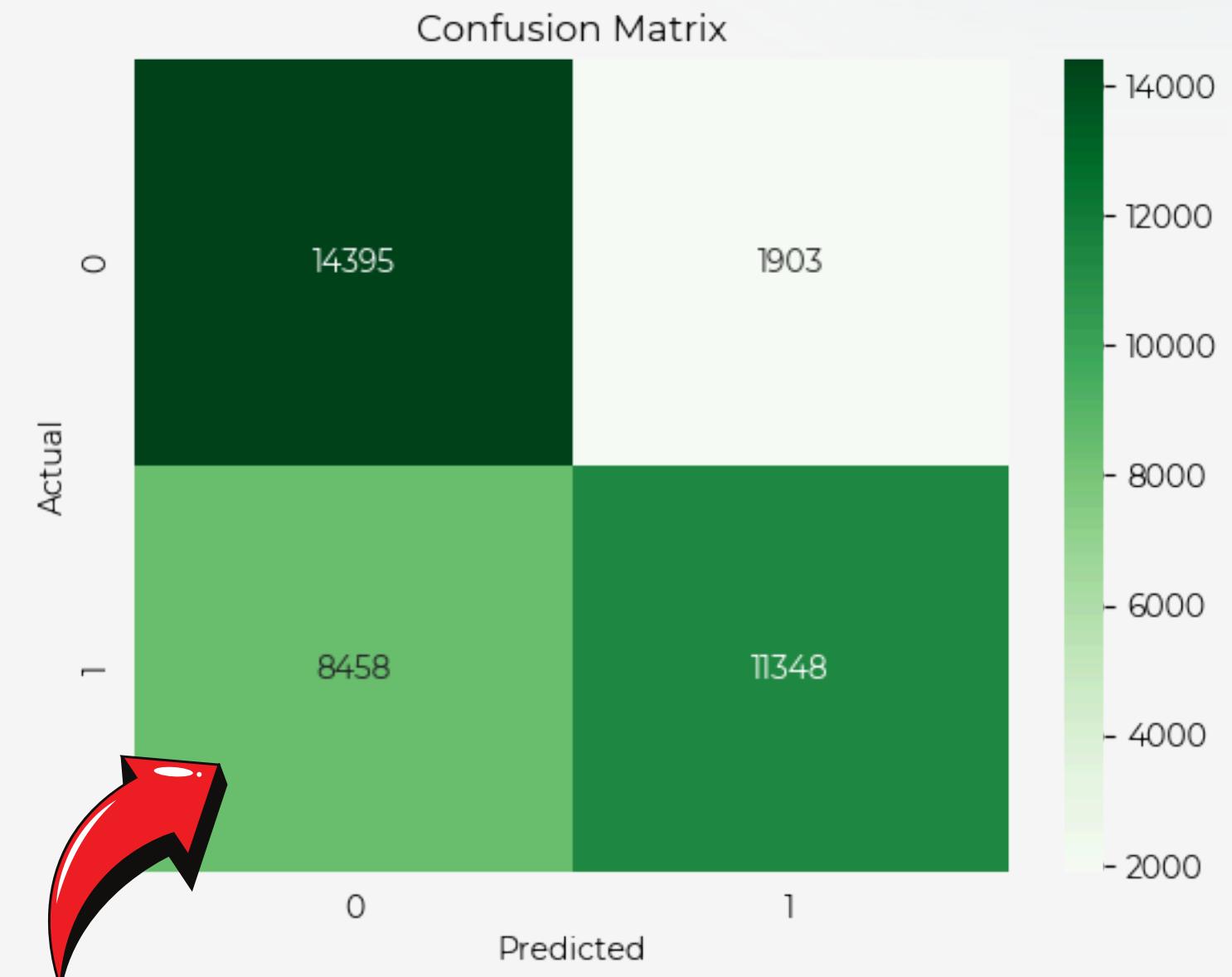
⚠ The model is moderately good but could be improved.

Cross-Validation: **Average Accuracy Training 0.709 | Test 0.709**

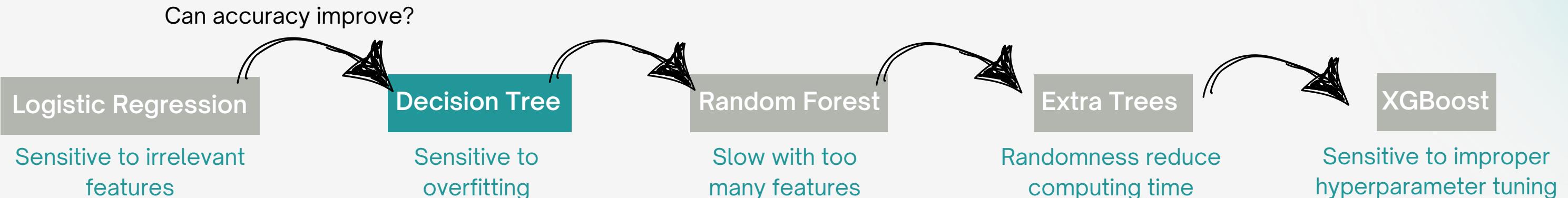
Hyperparameter tuning → Not significant improvement (**Accuracy = 0.711**)

Feature importance → Not significant improvement

Features transformation: PCA, MinMax → Not improvement



Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.841** | Precision = 0.854 | Recall = 0.856 | F1-score = 0.855
⚠ The model is moderately good, but there's room for improvement.
Cross-Validation: **Average Accuracy Training 1.0 | Test 0.825**

Hyperparameter tuning → Not improvement

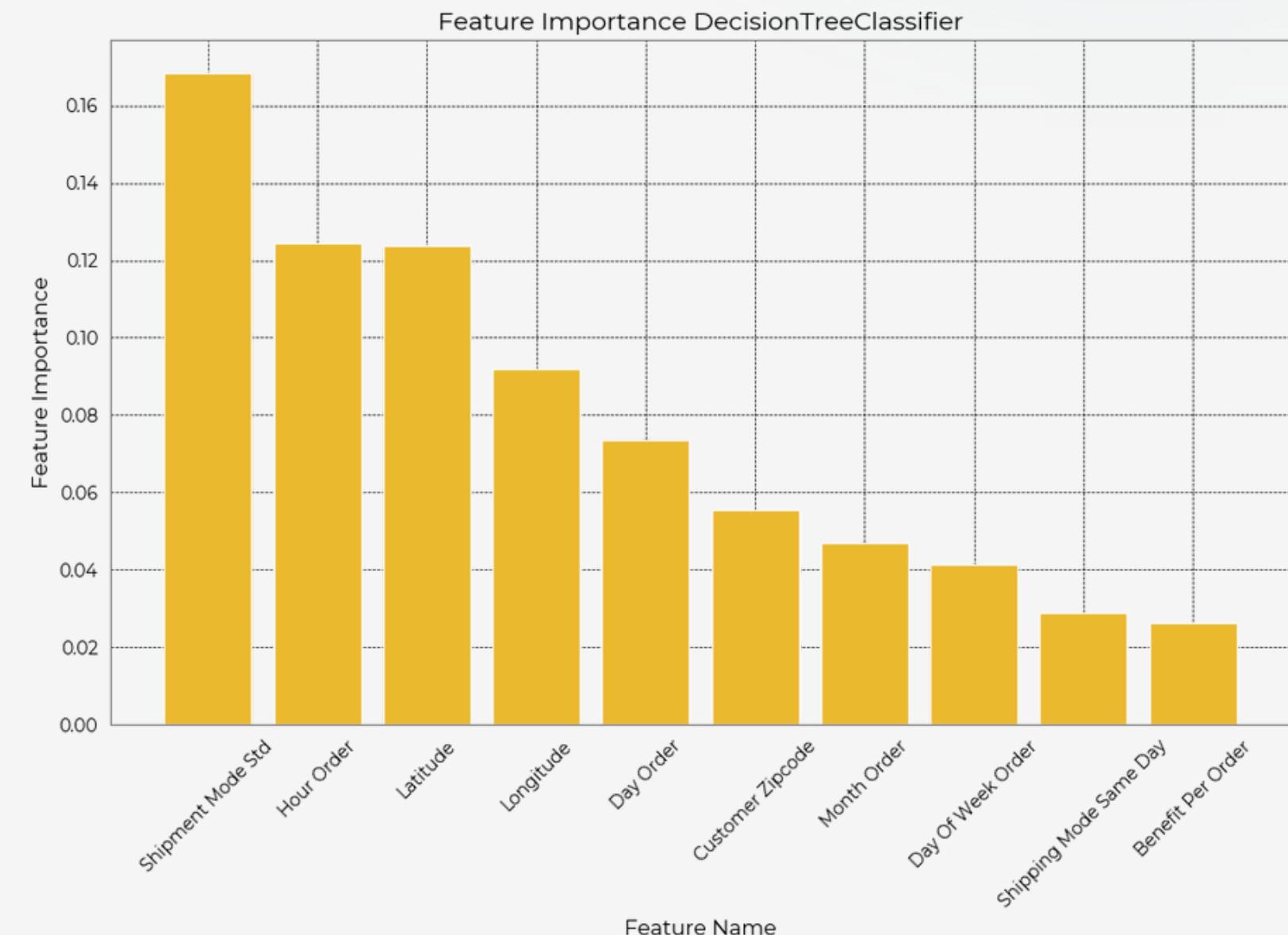
Feature importance → RFE with top 5 features

Model Metrics: **Accuracy = 0.923**

✓ The model performs well! It has high accuracy.

Cross-Validation: **Average Accuracy Training 0.977 | Test 0.923**

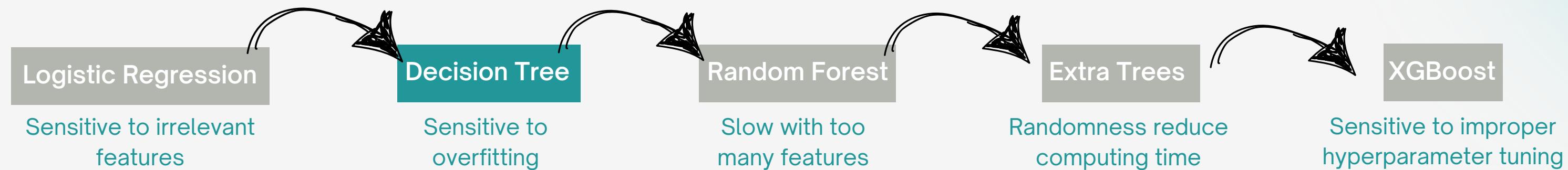
Features transformation: PCA, MinMax, Standardization → Not improvement



Building a Machine Learning Model for Delivery Risk Predictions



Can accuracy improve?



Model Metrics: **Accuracy = 0.841** | Precision = 0.854 | Recall = 0.856 | F1-score = 0.855

⚠ The model is moderately good, but there's room for improvement.

Cross-Validation: **Average Accuracy Training 1.0 | Test 0.825**

Overfitting

Hyperparameter tuning → Not improvement

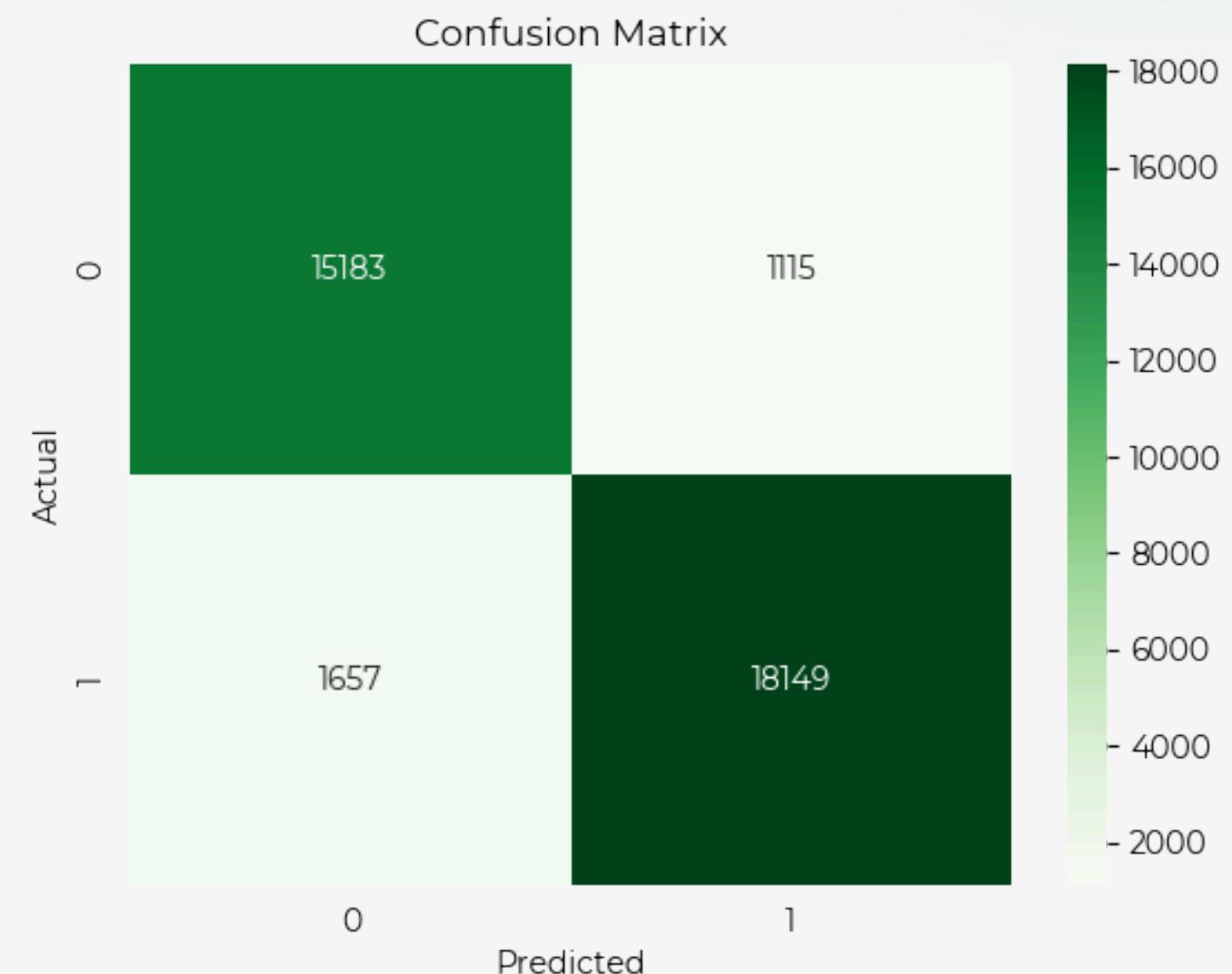
Feature importance → RFE with top 5 features

Model Metrics: **Accuracy = 0.923**

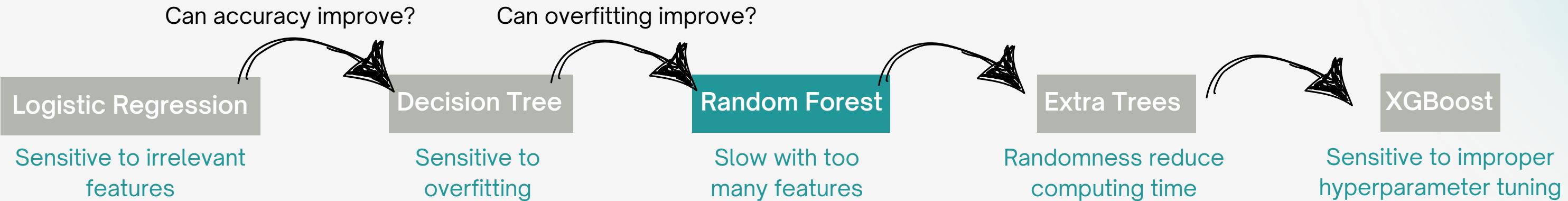
✓ The model performs well! It has high accuracy.

Cross-Validation: **Average Accuracy Training 0.977 | Test 0.923**

Features transformation: PCA, MinMax, Standardization → Not improvement



Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.787** | Precision = 0.878 | Recall = 0.709 | F1-score = 0.785
⚠ The model is moderately good, but there's room for improvement.

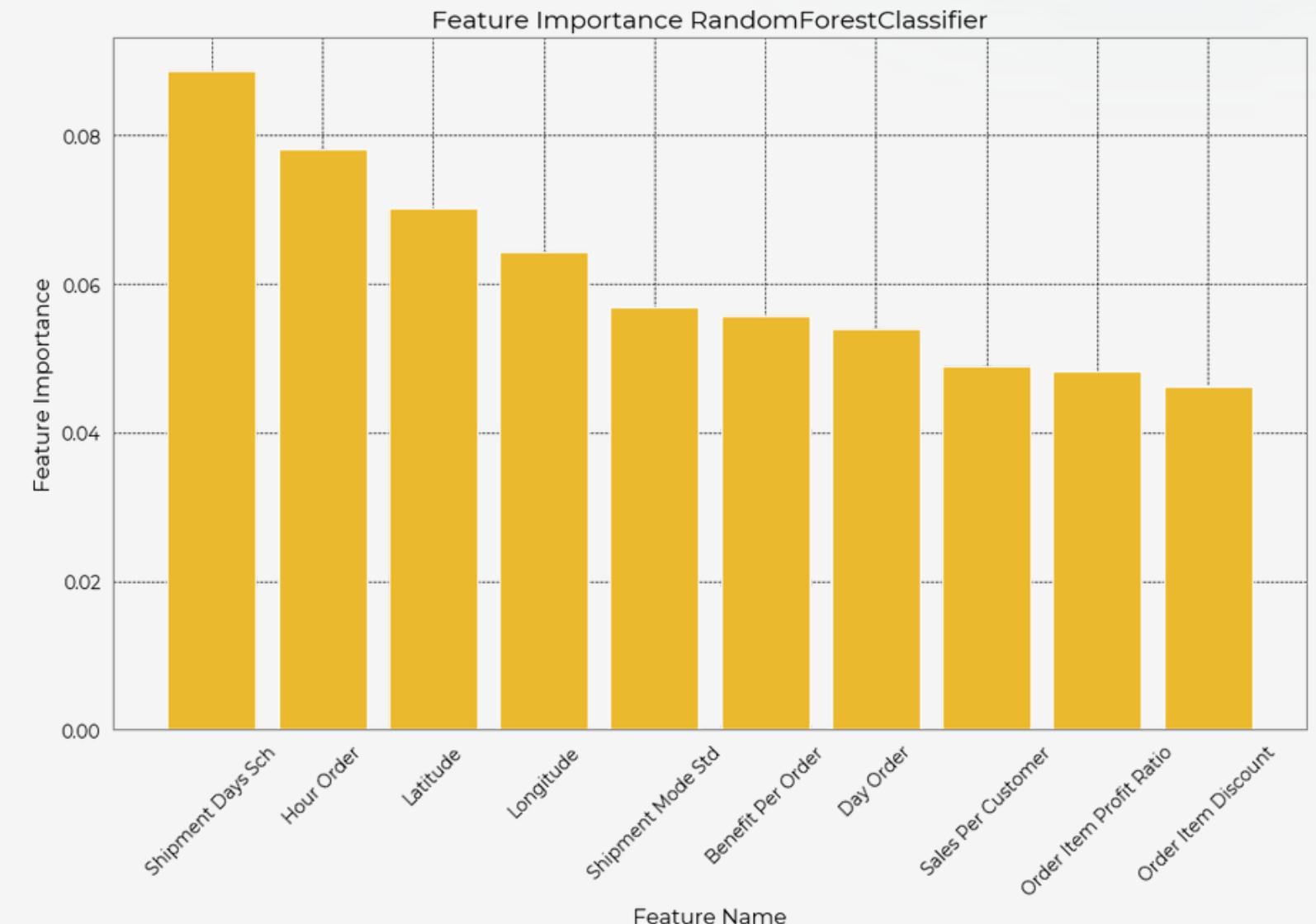
Cross-Validation: **Average Accuracy Training 1.0 | Test 0.783**

Hyperparameter tuning → Not improvement (need more computational power to try a larger range of values)

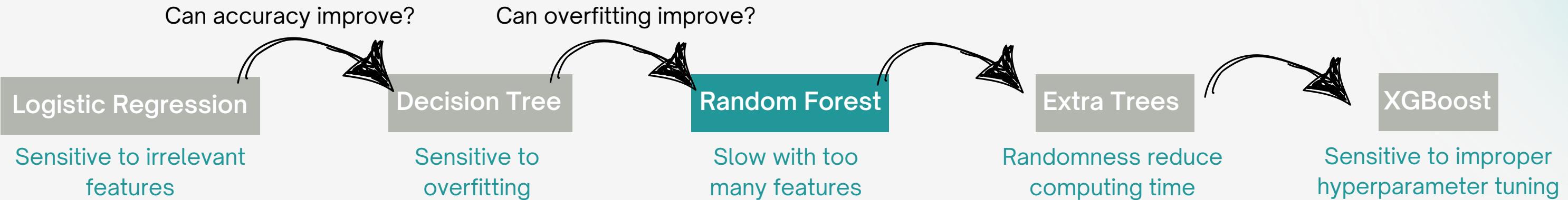
Feature importance → RFE didn't show an improve

To the computational-time the model could not be improved. A wider hyperparameter tuning might improve the model performance

Overfitting



Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.787** | Precision = 0.878 | Recall = 0.709 | F1-score = 0.785

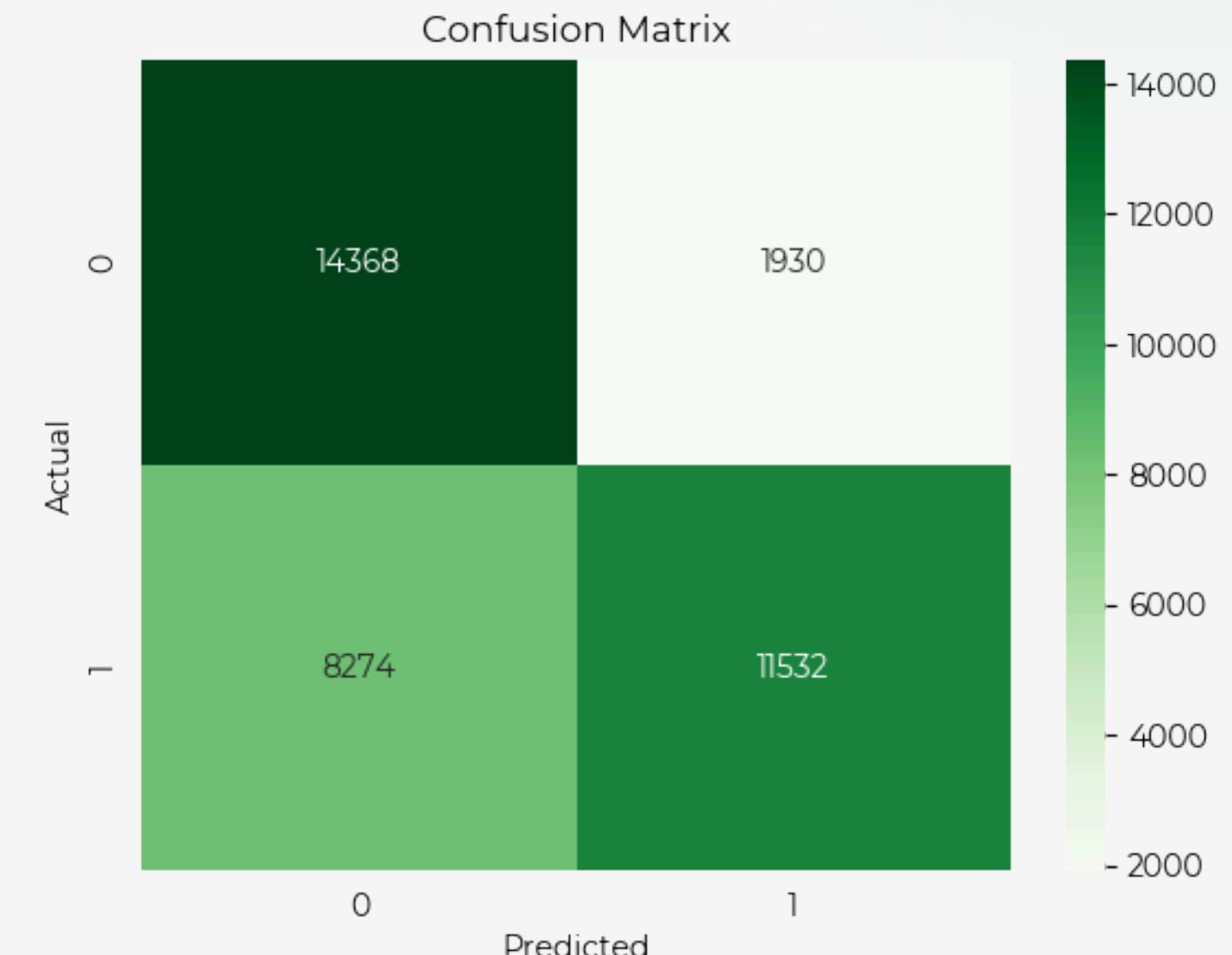
⚠ The model is moderately good, but there's room for improvement.

Cross-Validation: **Average Accuracy Training 1.0 | Test 0.803**

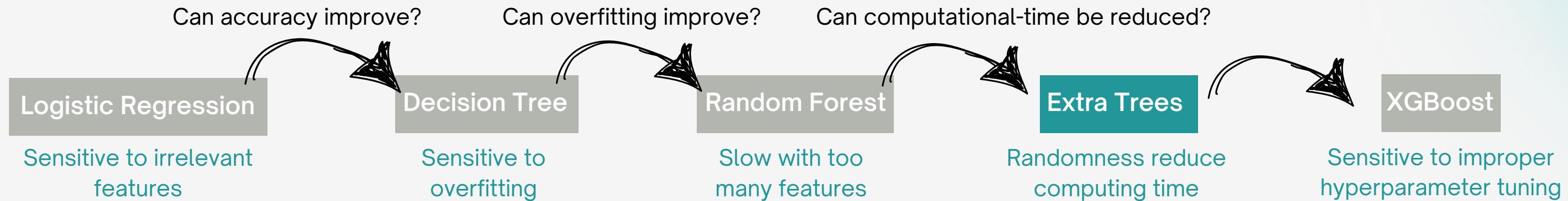
Hyperparameter tuning → Not improvement (need more computational power to try a larger range of values)

Feature importance → RFE didn't show an improve

To the computational-time the model could not be improved. A wider hyperparameter tuning might improve the model performance



Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.800** | Precision = 0.86 | Recall = 0.750 | F1-score = 0.804

⚠ The model is moderately good but could be improved.

Cross-Validation: **Average Accuracy Training 1.0 | Test 0.801**

Overfitting

Hyperparameter tuning → **Accuracy = 0.921**

✓ The model performs well! It has high accuracy.

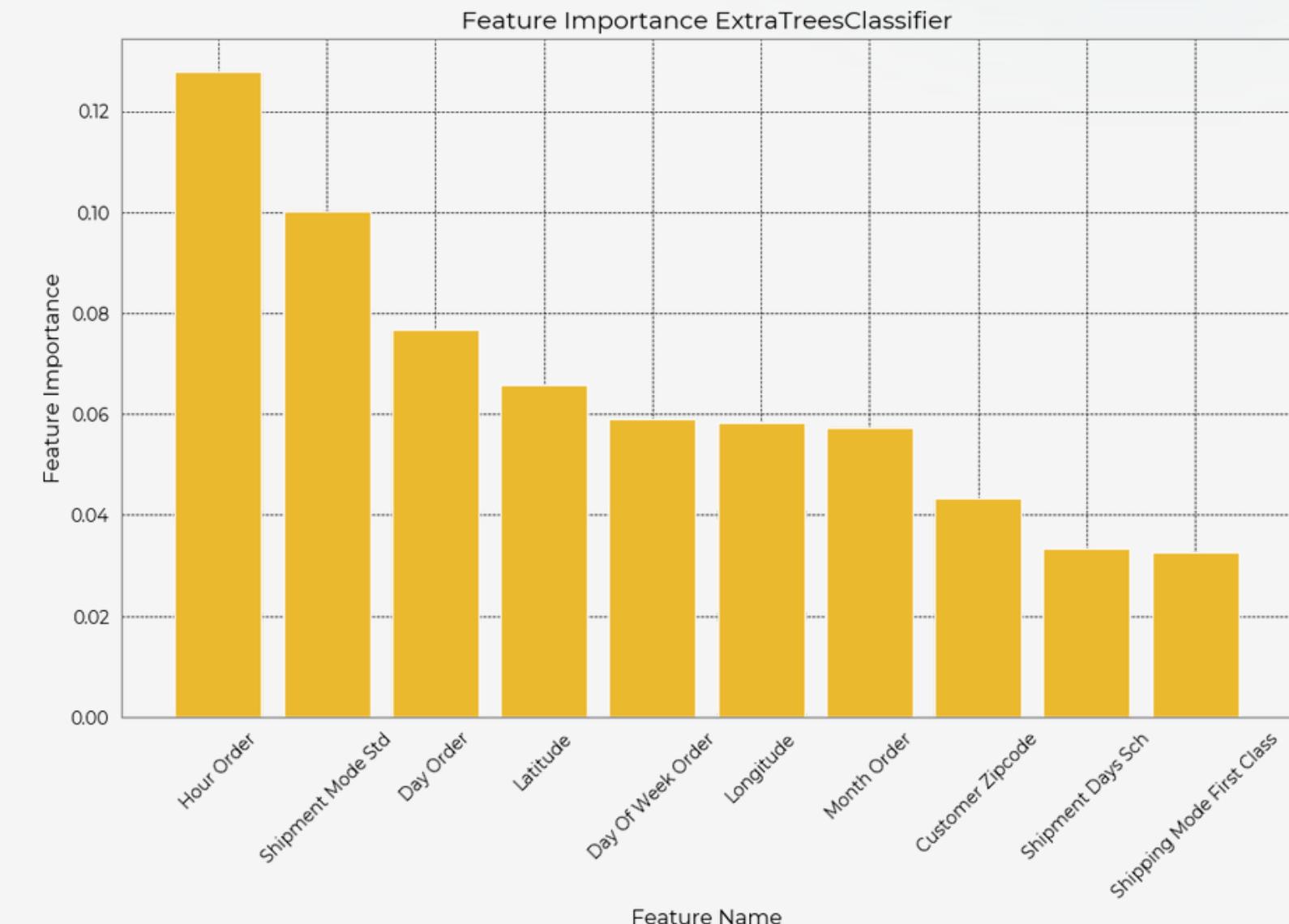
Feature importance → Top 10 features

Model Metrics: **Accuracy = 0.942**

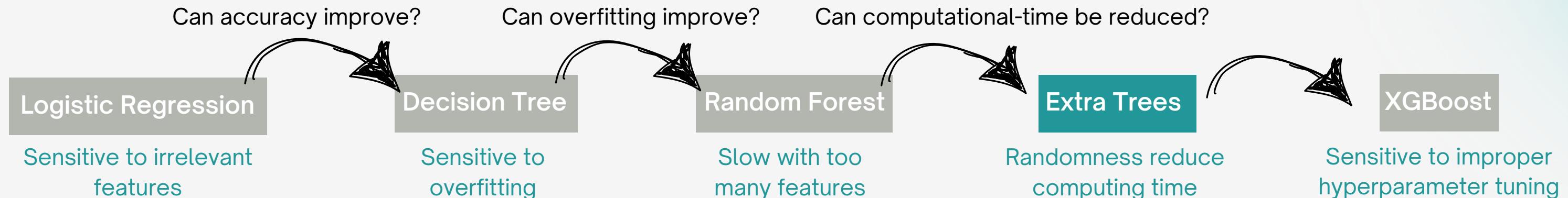
✓ The model performs well! It has high accuracy.

Cross-Validation: **Average Accuracy Training 0.991 | Test 0.940**

Features transformation: PCA, MinMax, Standardization → Not improvement



Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.800** | Precision = 0.86 | Recall = 0.750 | F1-score = 0.804

⚠ The model is moderately good but could be improved.

Cross-Validation: **Average Accuracy Training 1.0 | Test 0.801**

Overfitting

Hyperparameter tuning → **Accuracy = 0.921**

✓ The model performs well! It has high accuracy.

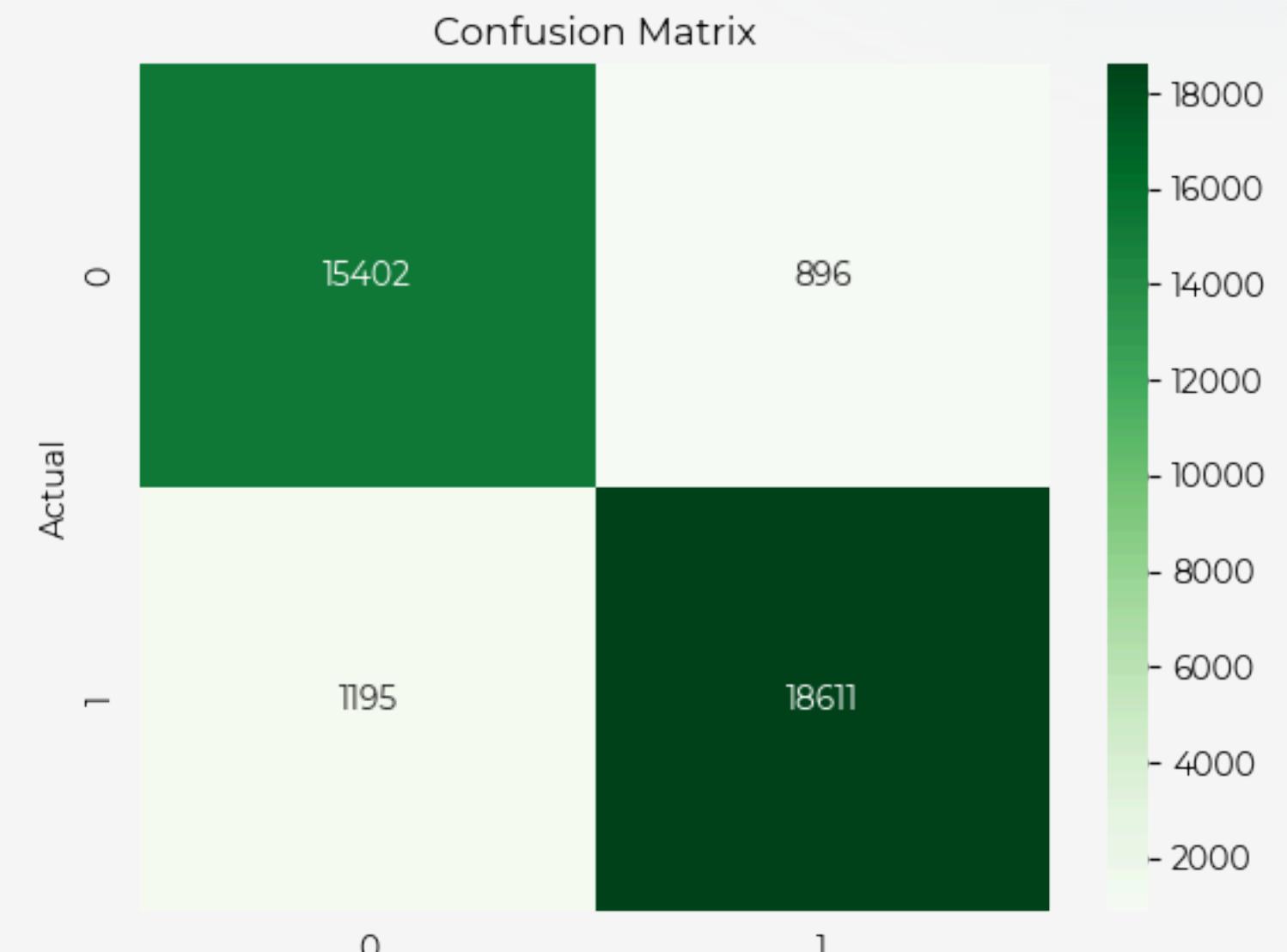
Feature importance → Top 10 features

Model Metrics: **Accuracy = 0.942**

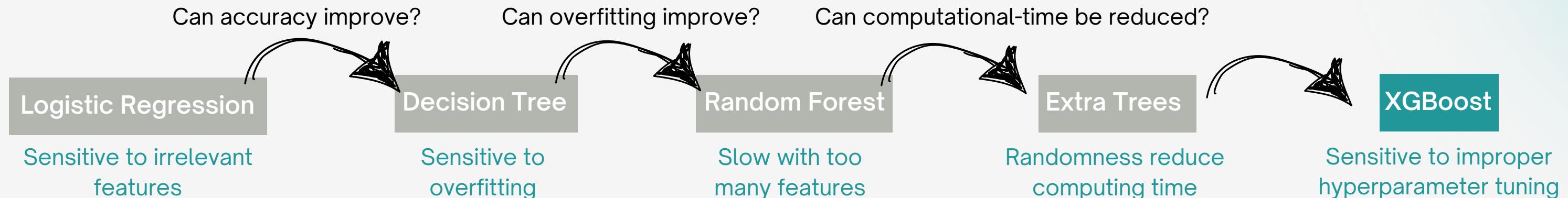
✓ The model performs well! It has high accuracy.

Cross-Validation: **Average Accuracy Training 0.991 | Test 0.940**

Features transformation: PCA, MinMax, Standardization → Not improvement



Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.800** | Precision = 0.86 | Recall = 0.750 | F1-score = 0.804

⚠ The model is moderately good but could be improved.

Cross-Validation: **Average Accuracy Training 1.0 | Test 0.801**

Overfitting

Hyperparameter tuning → **Accuracy = 0.921**

✓ The model performs well! It has high accuracy.

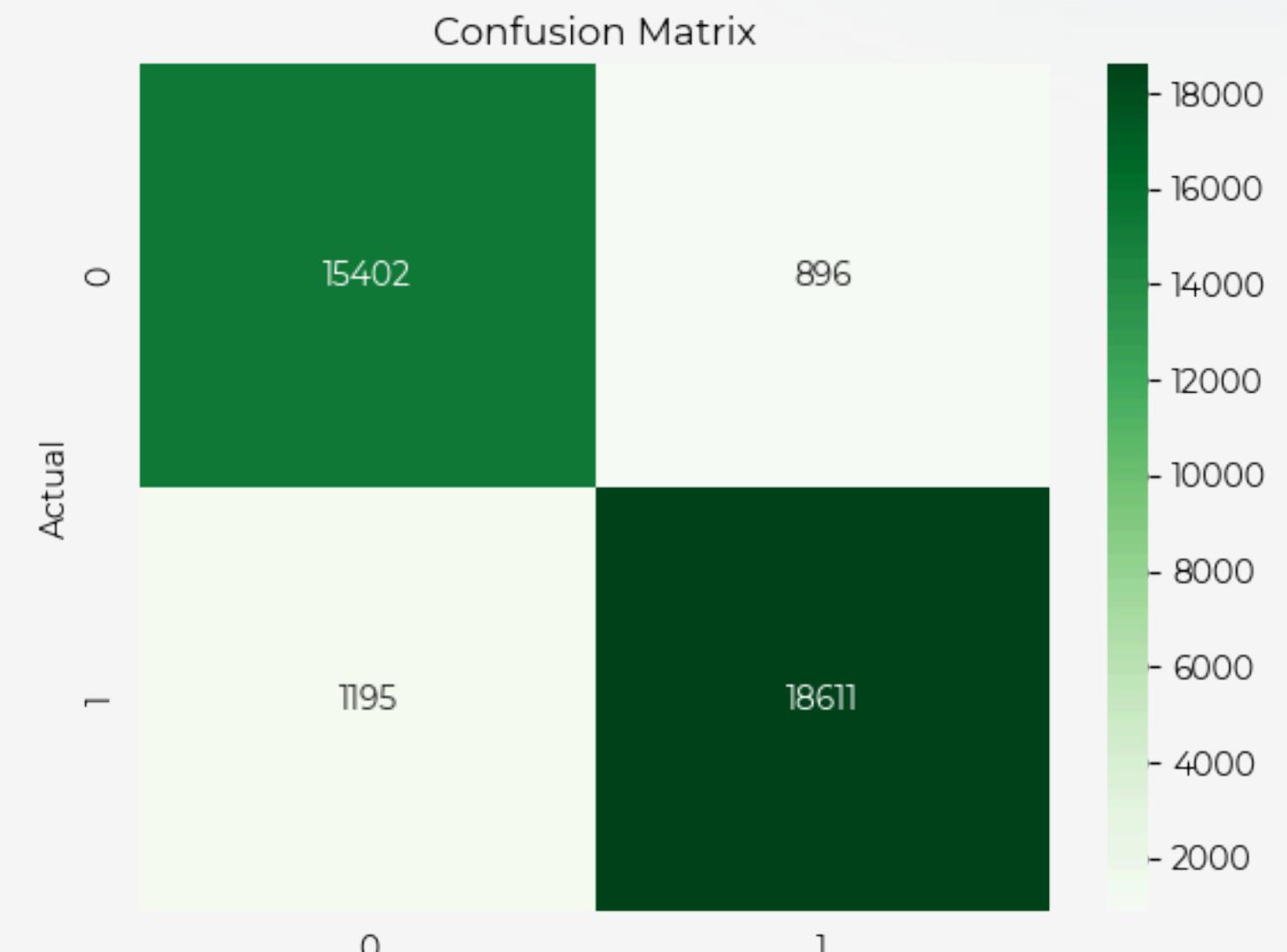
Feature importance → Top 10 features

Model Metrics: **Accuracy = 0.942**

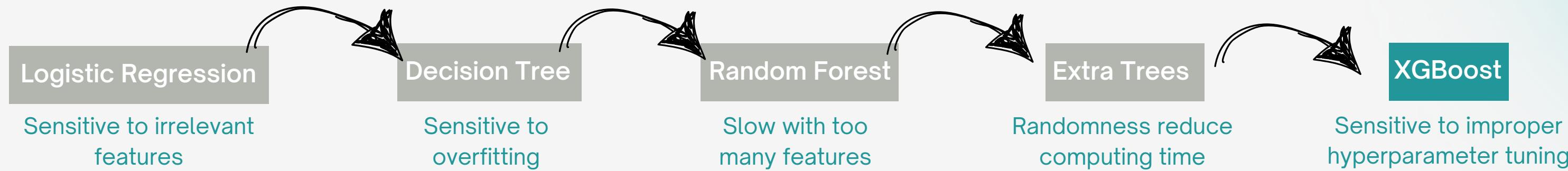
✓ The model performs well! It has high accuracy.

Cross-Validation: **Average Accuracy Training 0.991 | Test 0.940**

Features transformation: PCA, MinMax, Standardization → Not improvement



Building a Machine Learning Model for Delivery Risk Predictions



Model Metrics: **Accuracy = 0.737** | Precision = 0.835 | Recall = 0.649 | F1-score = 0.730

The model performs well! It has high accuracy.

Cross-Validation: **Average Accuracy Training 0.772 | Test 0.739**

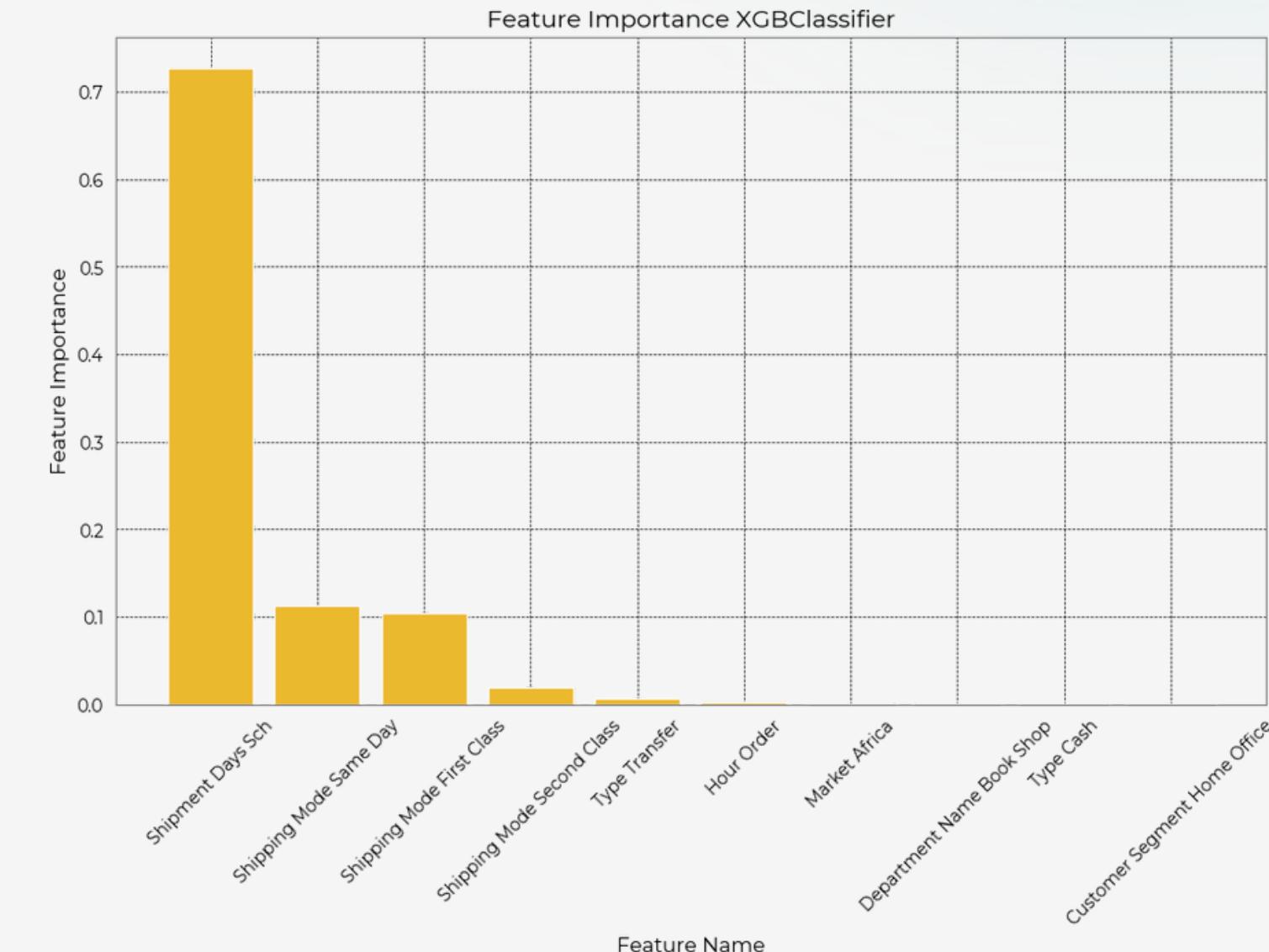
Hyperparameter tuning:

Model Metrics: **Accuracy = 0.900**

The model performs well! It has high accuracy.

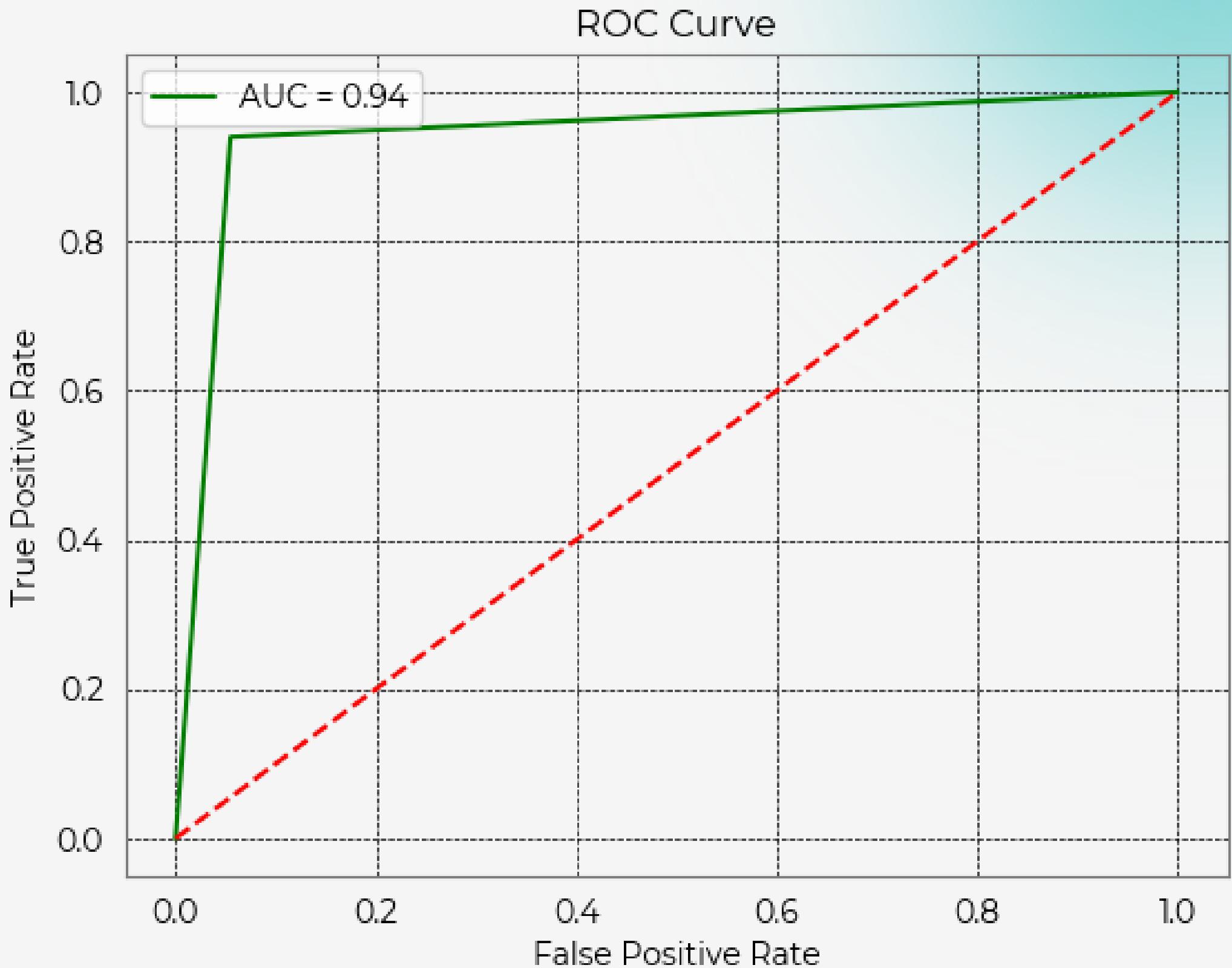
Feature importance → Top 5 features not improvement

Features transformation: PCA, MinMax, Standardization → Not improvement



Best Model

ExtraTreesClassifier



Hyperparameter tuning → Benefit from computational power