Identifying Risk of Late Deliveries in a Sports Retail Business

Kowsalya Nitya Vootla
Yating Jiang
Shuyang Zhang
Rishikumar Mathiazhagan

Company Overview - A sports retail business in the Americas

Distribution of orders across regions

21k orders fulfilled on average each year, raking in $42m in sales.

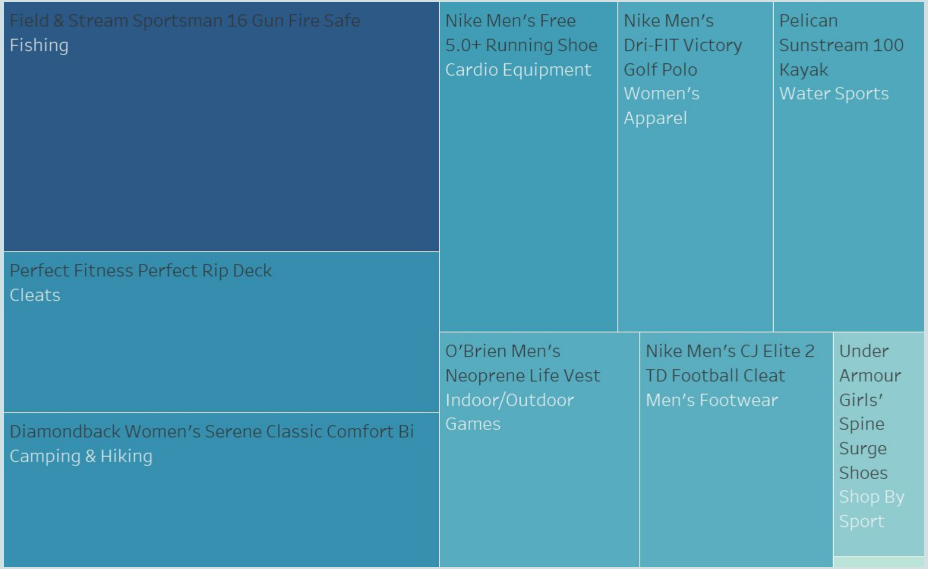
Orders sourced from USA and Puerto Rico and sold to **22 countries in 5 regions**:

- Central America
- South America
- South of USA
- East of USA
- West of USA

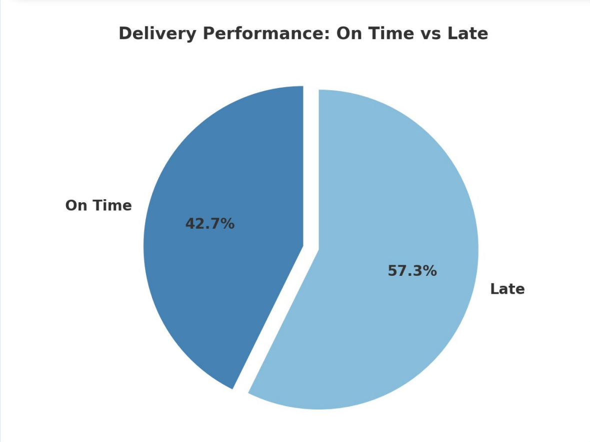
Company Overview - Sells a wide range of products, primarily fishing and sports accessories

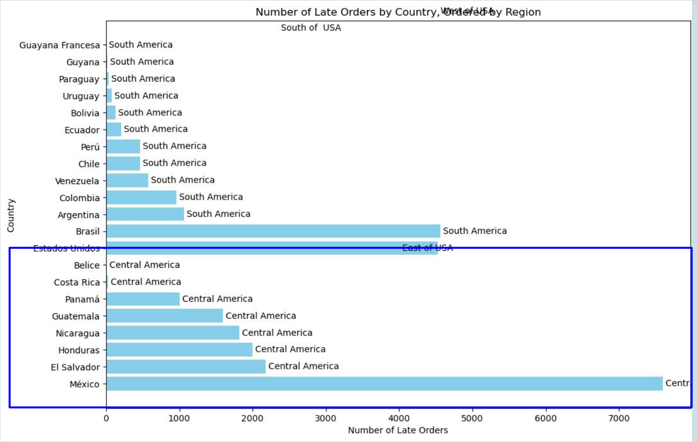
Top 10 Categories

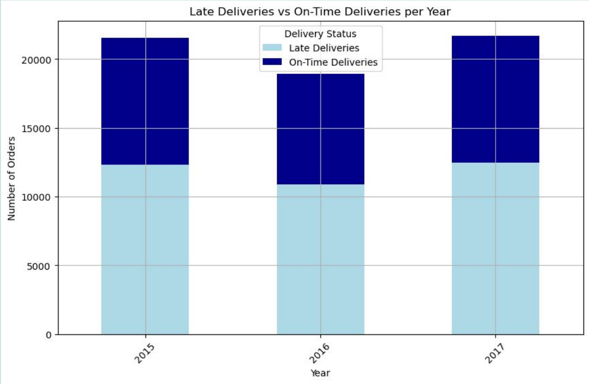
Top 10 Products

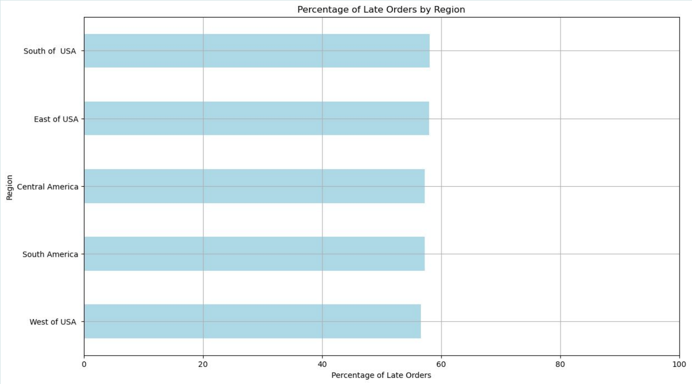


Key Problem: Late Deliveries









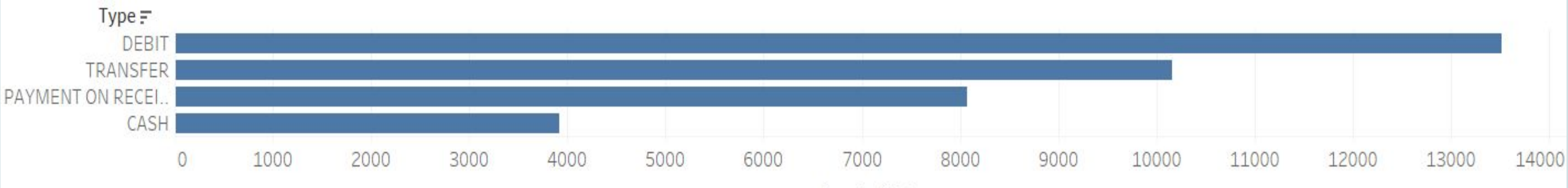
57% of the deliveries are late.

No decline in late delivery rate over the last 3 yrs.

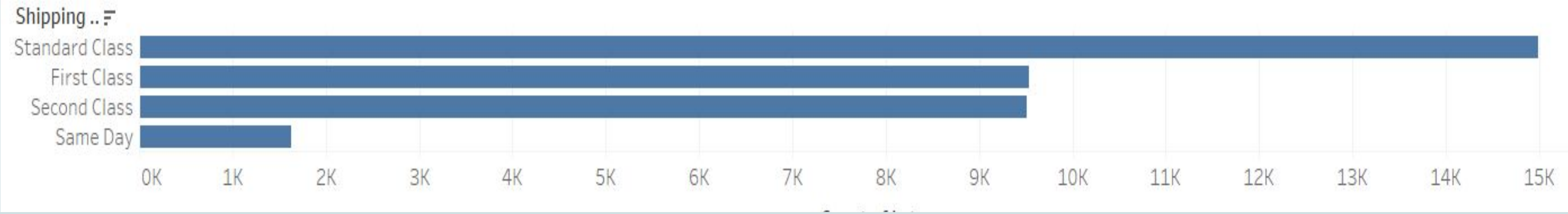
45% of the late deliveries are orders sent to Central America, but **late delivery rate across regions is consistent.**

Considered Variables vs Number of Late Deliveries

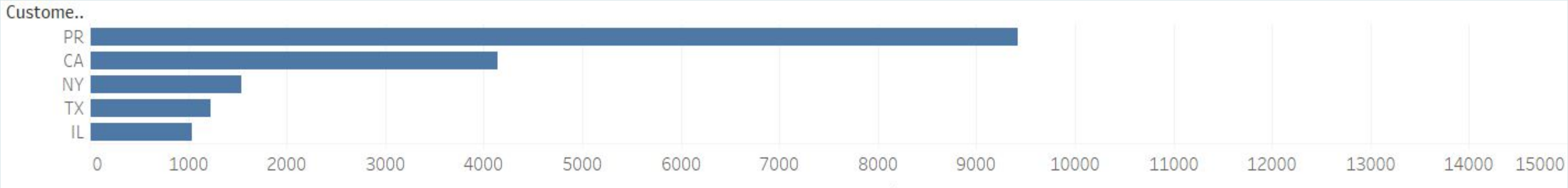
Payment Type



Shipping mode

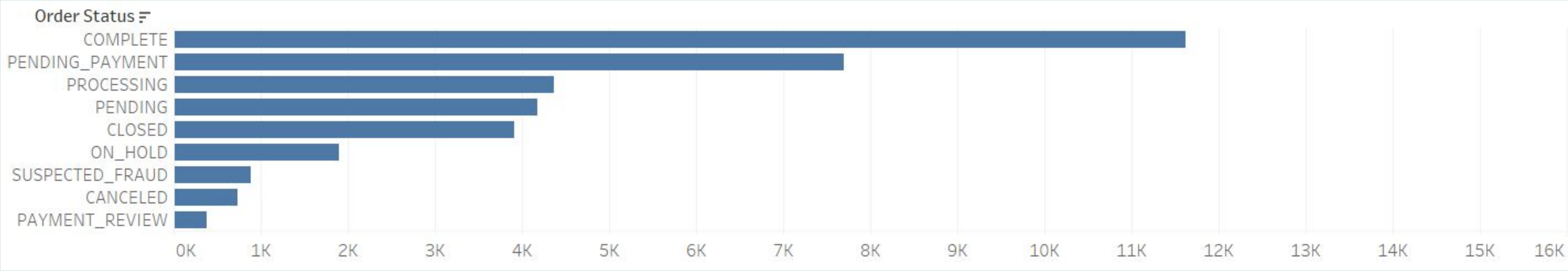


Dispatch Location

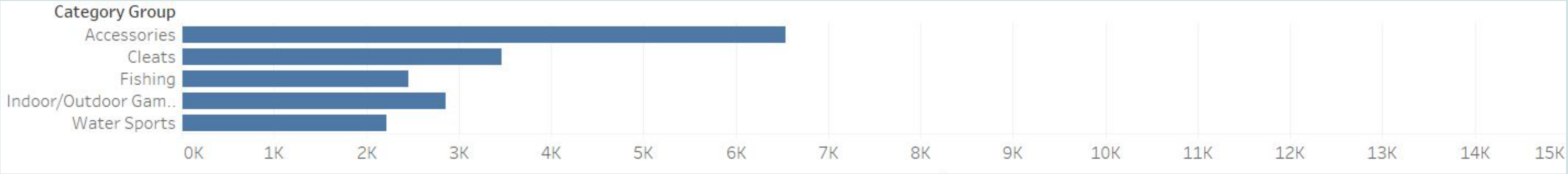


Considered Variables vs Number of Late Deliveries

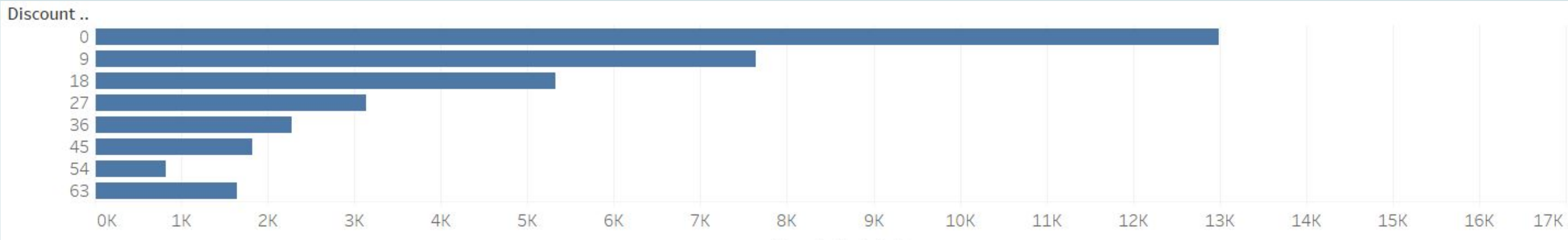
Order Status



Category



Discount Given

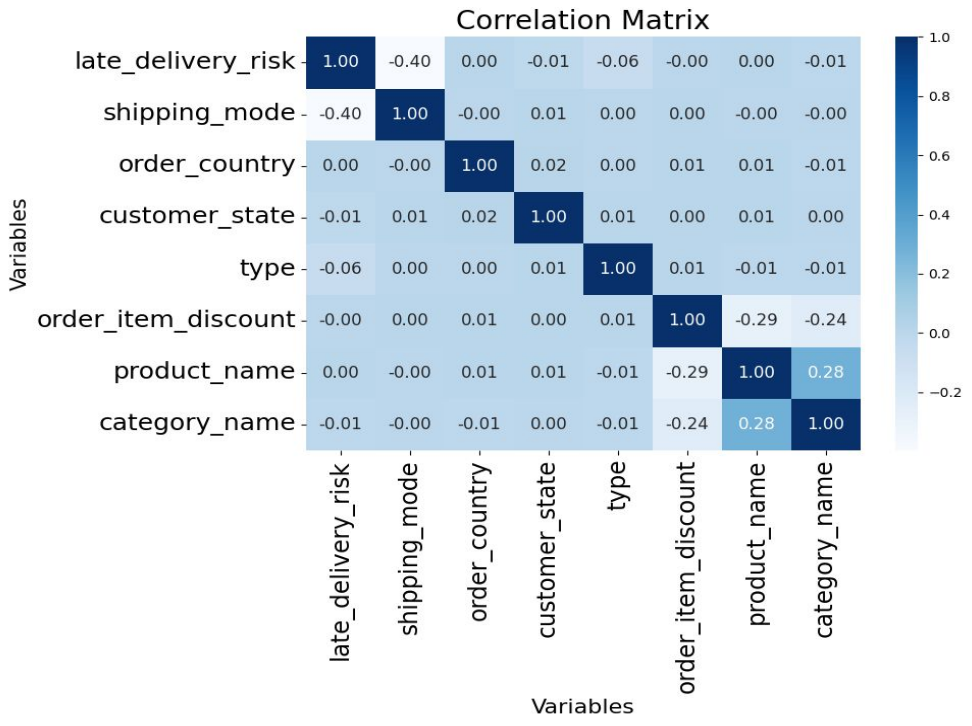


Statistical Test: Chi-square Analysis to identify significant variables

Variable	Chi-Square Statistic	P-Value	Reject Null Hypothesis
late_delivery_risk	56028.18	0	Yes
shipping_mode	14009.53	0	Yes
order_country	109.5	0	Yes
customer_state	85.46	0.0001	Yes
type	9.14	0.0274	Yes
order_item_discount	591.28	0.6365	No
product_name	88.46	0.4064	No
category_name	28.99	0.5178	No

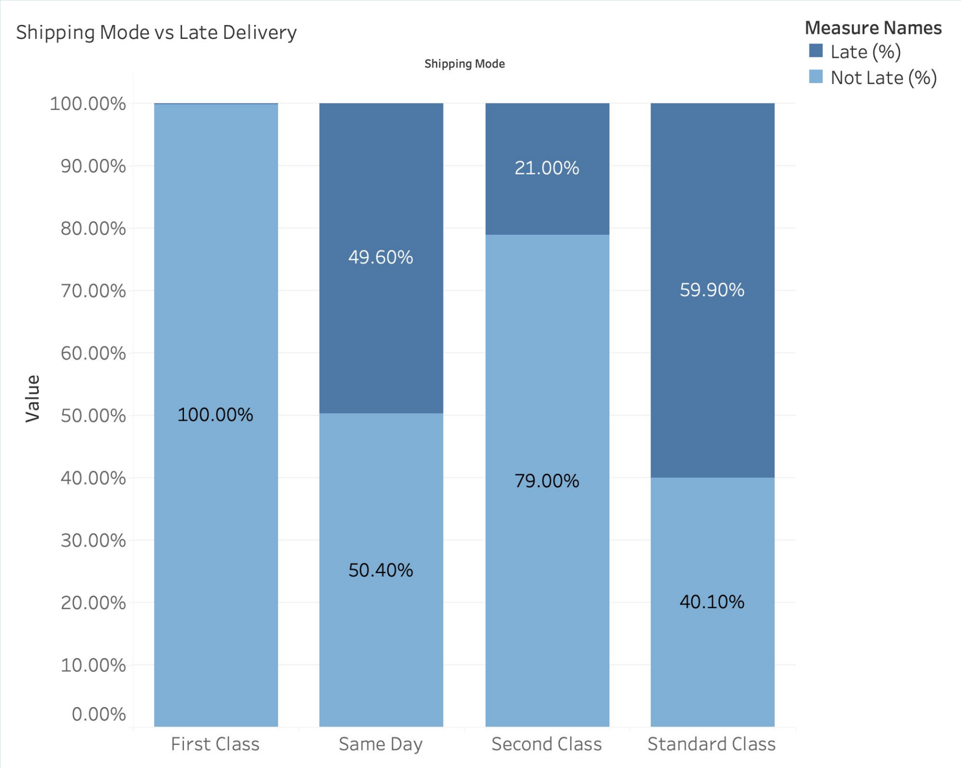
Chi-Square Analysis: Shipping mode, order country, customer state, and type significantly impact late delivery.

Low correlations ensure independent variables for stable model predictions



Low correlation: Minimal multicollinearity, ensuring stable and reliable model performance.

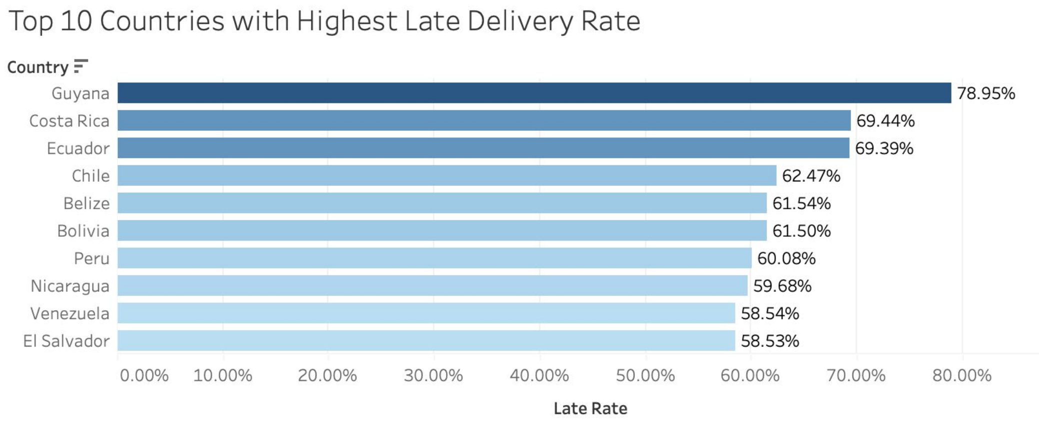
First Class ensures 0% late deliveries, while Standard Class has the highest rate at 59.9%.

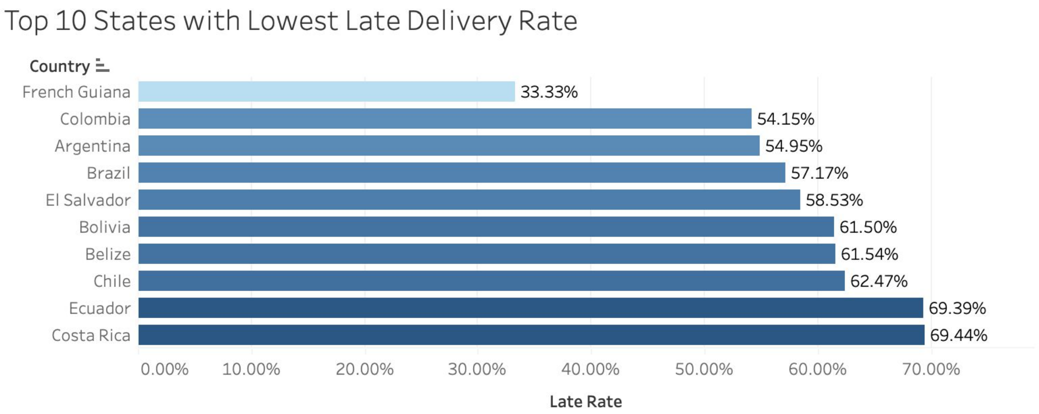


Ranking of Shipping Modes by Late Rate

- First Class: 0% (No late deliveries)
- Second Class: 21%
- Same Day: 49.6%
- Standard Class: 59.9%

Guyana leads late delivery rates, French Guiana achieves the lowest rate.





Top: Guyana and Costa Rica lead in late delivery rates, over 69%.

Bottom: French Guiana has the lowest late delivery rate at 33.33%, followed by Colombia (54.15%) and Argentina (54.95%).

No statistical significance was found among created features

New features were created as following:

- shipment_efficiency
- price_per_customer
- is_high_value_order
- order_weekday
- is_weekend

None of them are statistically significant

	Feature	F-Score	P-Value
0	shipment_efficiency	0.846904	0.357434
2	is_high_value_order	0.161667	0.687628
4	is_weekend	0.082937	0.773357
3	order_weekday	0.016138	0.898912
1	price_per_customer	0.001362	0.970560

Logistic regression selected for feature significance and interpretability

Variable Selection

Categorical Features: ['shipping_mode', 'type', 'order_country']

Numerical Features: None included due to statistical insignificance.

Data Split

Training Set: 80% of the data for model training

Testing Set: 20% of the data for evaluating performance

Pre processing

One-Hot Encoding: for categorical variables

Class Balancing: Used class_weight ='balanced' to handle imbalanced data

Evaluation Metrics

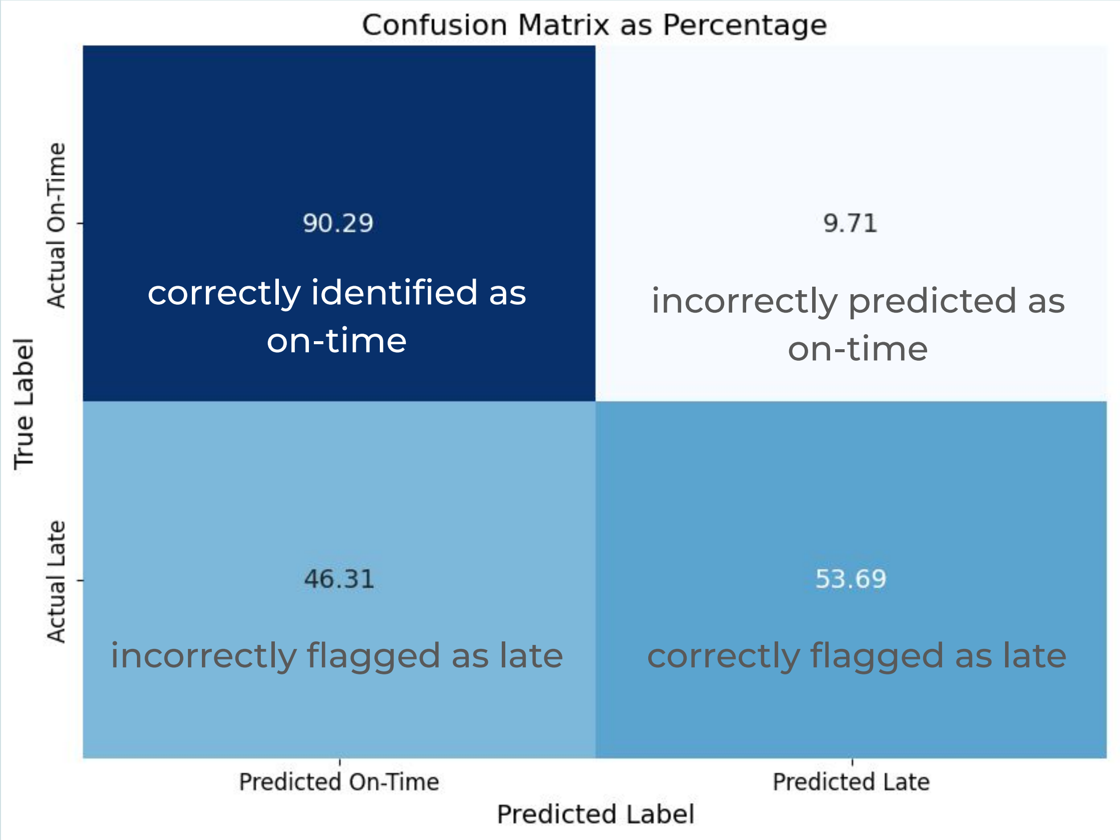
Precision, Recall, F1-Score: To assess classification performance

ROC AUC Score: To evaluate the model’s ability to distinguish between late and on-time deliveries

Threshold Optimization

Optimized decision threshold based on F1-score to balance precision and recall

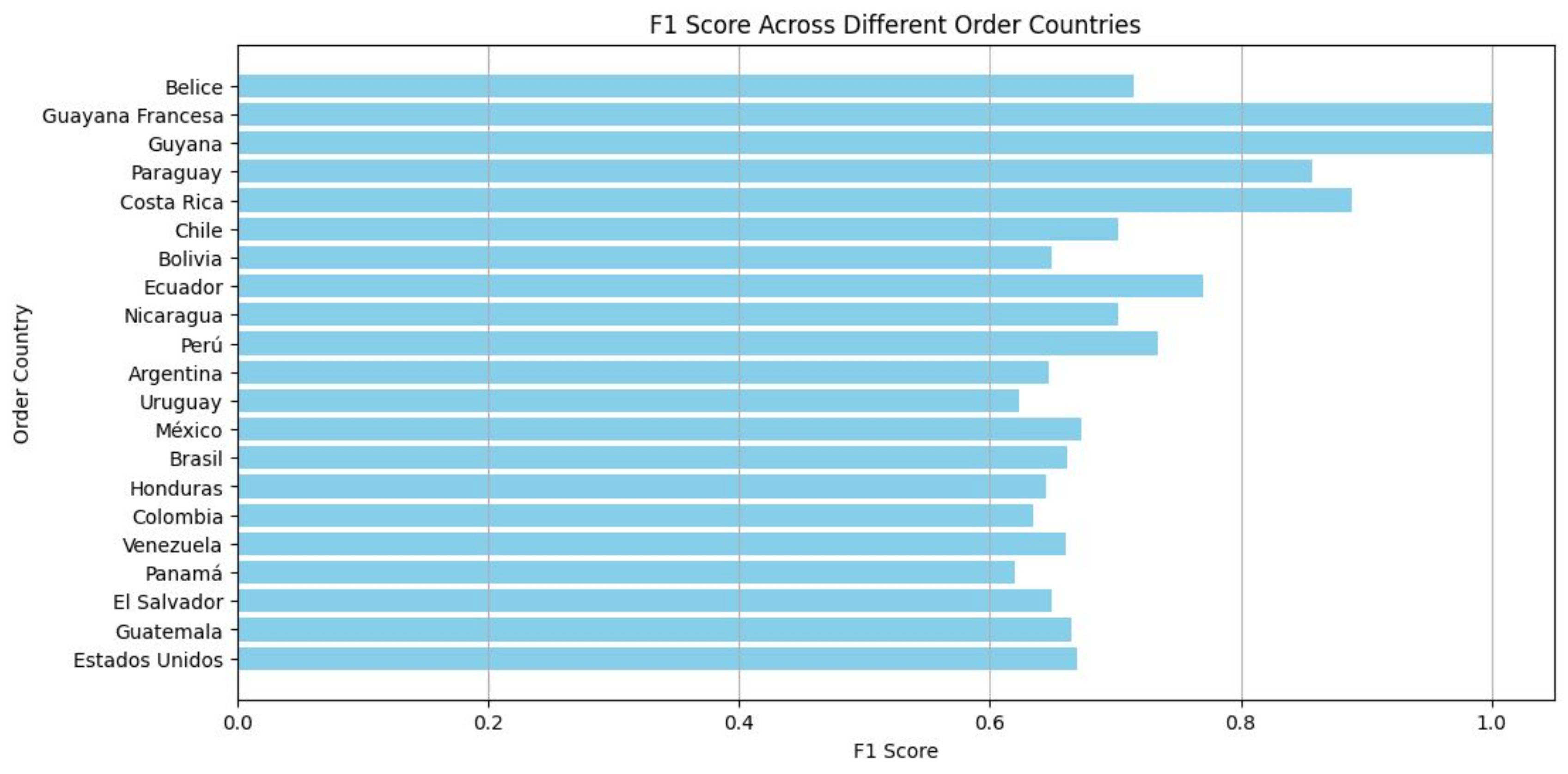
Model effectively flags late deliveries while minimizing false positives



- Optimal Threshold: 0.4545
- Max Avg F1: 0.8021
- ROC AUC Score: 0.7445
- Accuracy: 69%

- **True positive/negative:**
  - 53.69% of late deliveries flagged as late.
  - 90.29% of on-time deliveries flagged as on-time.
- **False positive/negative:**
  - 9.71% of on-time deliveries flagged as late
  - 46.31% of late deliveries flagged as on-time

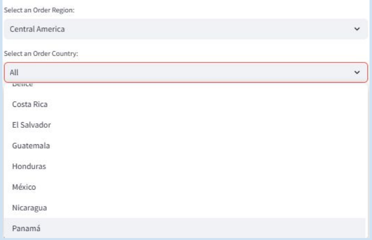
F1 scores varies across order countries



Two Interactive Streamlit Dashboards were developed using the model based on order country, shipping mode, and payment type

Dashboard 1:
High level overview of late delivery rates

- Target user: Operations Executive
- Trends based on Existing data



Dashboard 2:
Order level late deliver rates

- Target user: Operations Analysts/ Executives
- Trends based on simulated orders

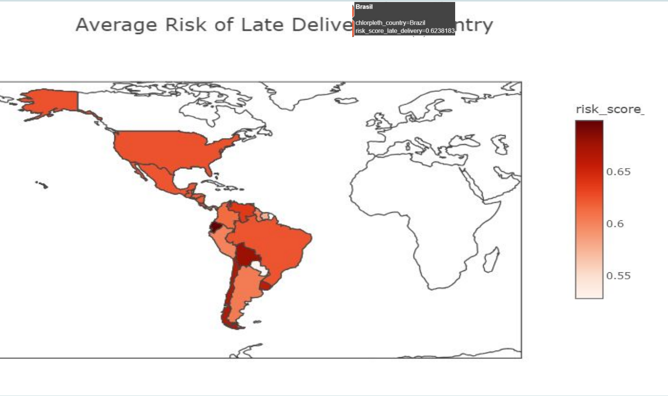

Filtered Order Data

	order_id	order_region	order_country	order_city	order_status	shipping_mode	ty
0	ORD-001	South of USA	Estados Unidos	Líbano	CLOSED	Standard Class	D
1	ORD-002	West of USA	Estados Unidos	Clovis	SUSPECTED_FRAUD	First Class	T
2	ORD-003	South America	Uruguay	Rivera	ON_HOLD	Standard Class	D
3	ORD-004	West of USA	Estados Unidos	Olympia	SUSPECTED_FRAUD	Second Class	P
4	ORD-005	West of USA	Estados Unidos	Parma	CANCELED	Standard Class	T
5	ORD-006	East of USA	Estados Unidos	Chattanooga	PENDING_PAYMENT	Second Class	D
6	ORD-007	South America	Guayana Francesa	Cayenne	CLOSED	Standard Class	T
7	ORD-008	South America	Paraguay	Asunción	CANCELED	First Class	P
8	ORD-009	South America	Guyana	Linden	COMPLETE	Standard Class	T
9	ORD-010	West of USA	Estados Unidos	Eugene	ON_HOLD	First Class	T

latedeliverydashboard.streamlit.app/

Dashboard 1 : High level overview of late delivery risks.
Graphs vary based on region and country selected

Average late delivery risk by country



Top 10 riskiest combination of variables

Top 10 Risky Combinations

	shipping_mode	type	order_city	order_country	order_region	risk_score_late_delivery	ri
14,193	First Class	TRANSFER	Portoviejo	Ecuador	South America	0.9986	H
14,161	First Class	TRANSFER	Cuenca	Ecuador	South America	0.9986	H
14,177	First Class	TRANSFER	Guayaquil	Ecuador	South America	0.9986	H
14,209	First Class	TRANSFER	Quevedo	Ecuador	South America	0.9986	H
14,225	First Class	TRANSFER	Quito	Ecuador	South America	0.9986	H
14,241	First Class	TRANSFER	Santo Domi	Ecuador	South America	0.9986	H
14,224	First Class	PAYMENT	Quito	Ecuador	South America	0.9986	H
14,240	First Class	PAYMENT	Santo Domi	Ecuador	South America	0.9986	H
14,160	First Class	PAYMENT	Cuenca	Ecuador	South America	0.9986	H
14,176	First Class	PAYMENT	Guayaquil	Ecuador	South America	0.9986	H

Heat map of risk based on shipping method and payment

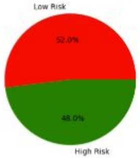
- Country with max risk score: Ecuador (0.70), Country with min risk score: Paraguay (0.53)
- “Premium” shipping methods have the highest risk of late delivery
- Transfer and cash payments have high amount of late delivery risk, while debit has the least amount of late delivery risk

Dashboard 2 : Order level late delivery risks.
Automatic Charts and Recommendations based on selected country, city, & order

Order Division by Risk Category

Optimal Threshold for High Risk: 0.45

Distribution of order based on risk of delivery compared to optimal threshold



List of High-Risk Orders

	order_id	order_region	order_country	order_city	order_status	shipping_mode	type
1	ORD-002	West of USA	Estados Unidos	Clovis	SUSPECTED_FRAUD	First Class	TRA
3	ORD-004	West of USA	Estados Unidos	Olympia	SUSPECTED_FRAUD	Second Class	PAYI
5	ORD-006	East of USA	Estados Unidos	Chattanooga	PENDING_PAYMENT	Second Class	DEB
7	ORD-008	South America	Paraguay	Asunción	CANCELED	First Class	PAYI
9	ORD-010	West of USA	Estados Unidos	Eugene	ON_HOLD	First Class	TRA
13	ORD-014	East of USA	Estados Unidos	Owensboro	PENDING_PAYMENT	First Class	CAS
15	ORD-016	East of USA	Estados Unidos	Long Beach	CLOSED	First Class	TRA
17	ORD-018	West of USA	Estados Unidos	Brentwood	ON_HOLD	Second Class	TRA
19	ORD-020	South of USA	Estados Unidos	Asheville	PROCESSING	First Class	CAS
20	ORD-021	East of USA	Estados Unidos	Laurel	PROCESSING	First Class	PAYI

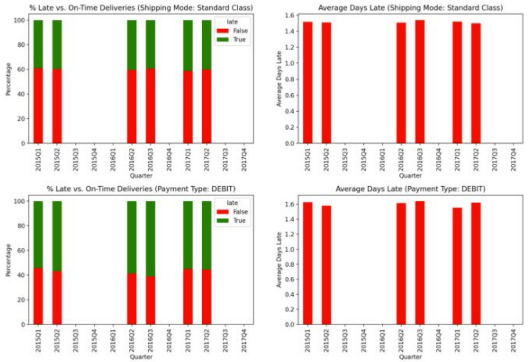
Select Order ID

ORD-001

Selected Order Details:

	order_id	order_region	order_country	order_city	order_status	shipping_mode	type	product
0	ORD-001	South of USA	Estados Unidos	Líbano	CLOSED	Standard Class	DEBIT	Product

Late Delivery Risk Prediction: Low Risk (0.33)



Recommendations for Selected Order

The selected order has a **High Risk** of late delivery (Risk Score: 0.73).

- **Switch to 'Standard Class' shipping mode to reduce risk'.**
- **Switch to 'DEBIT' payment type to reduce risk'.**

Validation of Order Level Recommendation by generating recommendations for all simulated data

Late Delivery Risk Percentages:

- **Original Late Risk Deliveries:** 48.00%
- **% Late Risk Deliveries (Only Shipping Updated):** 0.00%
- **% Late Risk Deliveries (Only Payment Updated):** 47.00%
- **% Late Risk Deliveries (Both Updated):** 0.00%

Overall recommendation: Update the payment type and shipping mode based on recommendation, otherwise **prioritize the update of shipping mode**.

Conclusion for the business to improve its on time delivery

Improve payment processing systems to minimize delays

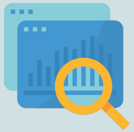
Aim for fewer transfer, and more cash and debit payment

Offer optimal shipping options for time-sensitive deliveries

Standard class has better efficiency

Investigate why premium options like first class and same day are not as efficient

Size of delivery fleet



Collect more data to help meet customer needs efficiently

Potentially stocking products in local hubs instead of just US and PR

Alternate efficient delivery routes

Local regulations

Thank You

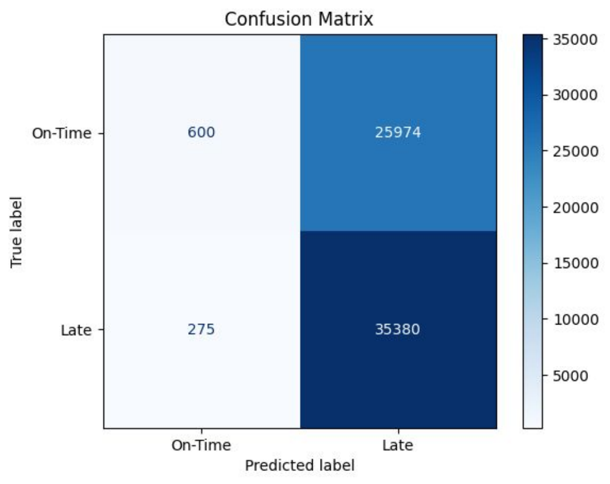
And Happy Holidays!
Hope your holiday shopping reaches you on time.

APPENDIX

Predictive Model: Model optimize recall for late deliveries

```python
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.metrics import classification_report, precision_recall_curve
import joblib  # Correctly importing joblib
import pandas as pd
from sklearn.metrics import ConfusionMatrixDisplay, roc_curve, auc

# Define the simplified features and categorical preprocessing
simplified_features = ['shipping_mode', 'type', 'order_country']
simplified_categorical_features = simplified_features

# Create a preprocessing pipeline for simplified features
simplified_preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(drop='first'), simplified_categorical_features)
    ]
)

# Train the final logistic regression model
final_model_pipeline = Pipeline(steps=[
    ('preprocessor', simplified_preprocessor),
    ('classifier', LogisticRegression(max_iter=500, class_weight='balanced', random_state=42))
])

final_model_pipeline.fit(X[simplified_features], y)

# Calculate the optimal threshold based on F1-score
precision, recall, thresholds = precision_recall_curve(y, final_model_pipeline.predict_proba(X[simplified_features])[:, 1])
optimal_threshold = None
max_f1 = 0
for p, r, t in zip(precision, recall, thresholds):
    f1 = 2 * (p * r) / (p + r) if (p + r) > 0 else 0
    if f1 > max_f1:
        max_f1 = f1
        optimal_threshold = t

print(f"Optimal Threshold: {optimal_threshold}, Max F1: {max_f1}")
```

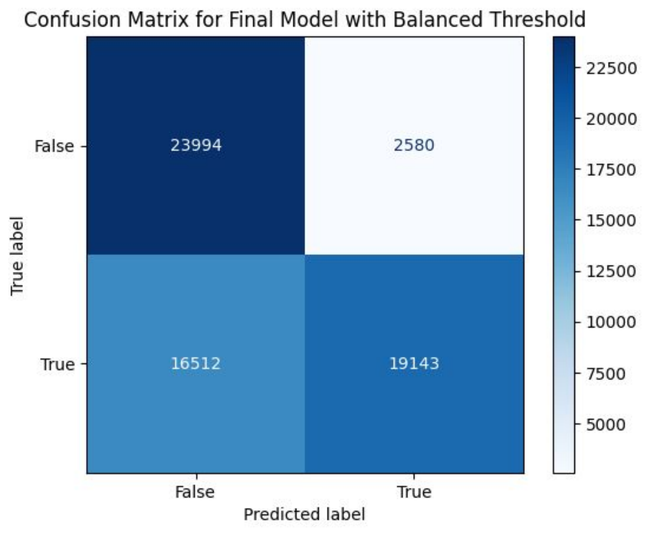
```
Optimal Threshold: 0.2980952536454114, Max F1: 0.7294168582296487
Final Model Performance with Optimal Threshold:
              precision    recall  f1-score   support

       False       0.69      0.02      0.04     26574
        True       0.58      0.99      0.73     35655

    accuracy                           0.58     62229
   macro avg       0.63      0.51      0.39     62229
weighted avg       0.62      0.58      0.44     62229

ROC AUC Score: 0.7445080251898064
```



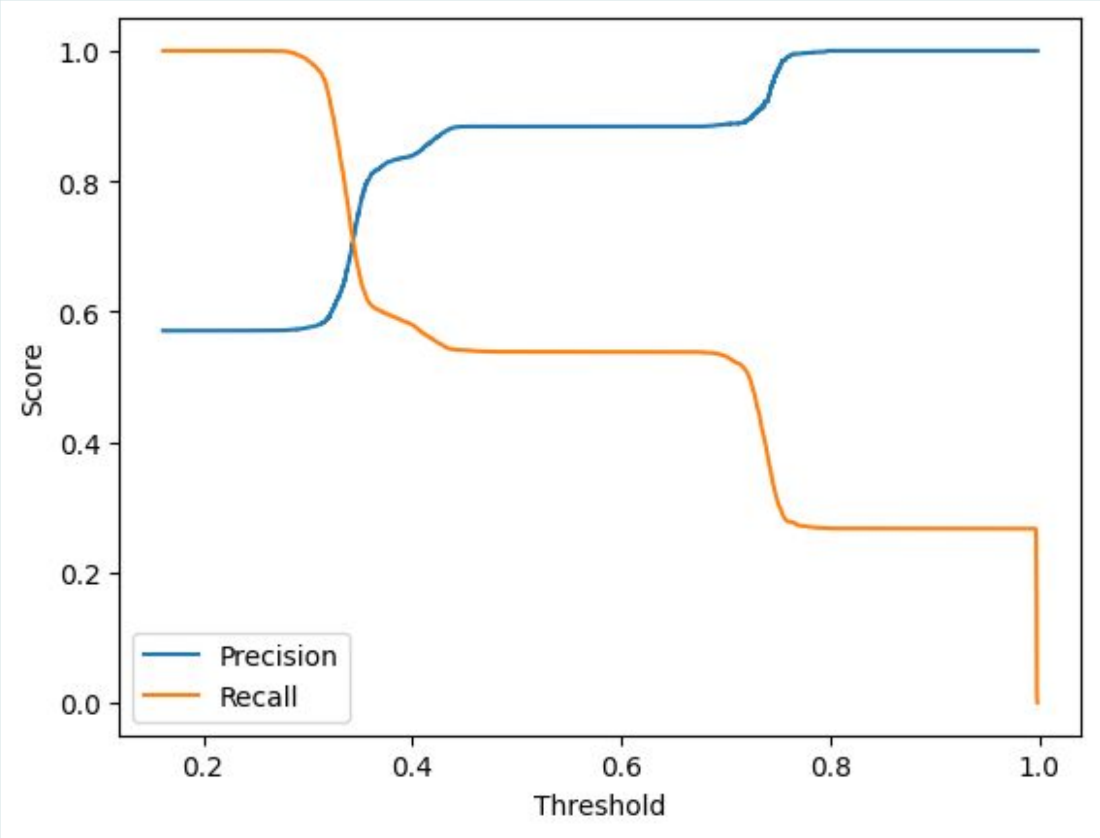
Predictive Model: Model optimize average F1 scores

```python
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.metrics import classification_report, precision_recall_curve, roc_auc_score
from sklearn.preprocessing import OneHotEncoder
import pandas as pd
from sklearn.metrics import ConfusionMatrixDisplay

# Define the simplified features and categorical preprocessing
simplified_features = ['shipping_mode', 'type', 'order_country']
simplified_categorical_features = simplified_features

# Create a preprocessing pipeline for simplified features
simplified_preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(drop='first'), simplified_categorical_features)
    ]
)

# Train the final logistic regression model
final_model_pipeline = Pipeline(steps=[
    ('preprocessor', simplified_preprocessor),
    ('classifier', LogisticRegression(max_iter=500, class_weight='balanced', random_state=42))
])

final_model_pipeline.fit(X[simplified_features], y)

# Calculate the optimal threshold based on F1-score for balanced classes
precision, recall, thresholds = precision_recall_curve(y, final_model_pipeline.predict_proba(X[simplified_features])[:, 1])
optimal_threshold = None
max_avg_f1 = 0

for p, r, t in zip(precision, recall, thresholds):
    f1_false = 2 * (p * recall[0]) / (p + recall[0]) if (p + recall[0]) > 0 else 0  # F1 for False (On-Time)
    f1_true = 2 * (p * r) / (p + r) if (p + r) > 0 else 0  # F1 for True (Late)
    avg_f1 = (f1_false + f1_true) / 2  # Average F1 score for both classes

    if avg_f1 > max_avg_f1:
        max_avg_f1 = avg_f1
        optimal_threshold = t

print(f"Optimal Threshold (Balanced Classes): {optimal_threshold}, Max Avg F1: {max_avg_f1}")
```

```
Optimal Threshold (Balanced Classes): 0.45451189375164375, Max Avg F1: 0.8020630306699034
Final Model Performance with Balanced Threshold:
              precision    recall  f1-score   support

       False       0.59      0.90      0.72     26574
        True       0.88      0.54      0.67     35655

    accuracy                           0.69     62229
   macro avg       0.74      0.72      0.69     62229
weighted avg       0.76      0.69      0.69     62229

ROC AUC Score: 0.7445080251898064
```



Predictive Model: Visualization of Precision-Recall Curve



Simulated Data for Current Orders

```python
# Simulate 100 orders
np.random.seed(42)
regions = np.random.choice(list(region_to_countries.keys()), size=100)
countries = [np.random.choice(region_to_countries[region]) for region in regions]
cities = [np.random.choice(country_to_cities[country]) for country in countries]
order_statuses = ["PENDING_PAYMENT", "PENDING", "PROCESSING", "ON_HOLD", "COMPLETE", "CLOSED", "SUSPECTED_FRAUD", "CANCELED", "PAYMENT_REVIEW"]
shipping_modes = ["First Class", "Same Day", "Second Class", "Standard Class"]
payment_types = ["PAYMENT", "TRANSFER", "DEBIT", "CASH"]
products = [f"Product {i}" for i in range(1, 21)]

simulated_data = pd.DataFrame({
    "order_id": [f"ORD-{i:03d}" for i in range(1, 101)],
    "order_region": regions,
    "order_country": countries,
    "order_city": cities,
    "order_status": np.random.choice(order_statuses, 100),
    "shipping_mode": np.random.choice(shipping_modes, 100),
    "type": np.random.choice(payment_types, 100),
    "product_name": np.random.choice(products, 100),
    "order_date_(dateorders)": [datetime(2023, 1, 1) + timedelta(days=np.random.randint(1, 365)) for _ in range(100)]
})


# Predict late delivery risk based on shipping mode
simulated_data["late_delivery_risk"] = final_model_pipeline.predict_proba(simulated_data[["shipping_mode","order_status","type","order_country"]])[:, 1]
simulated_data["risk_category"] = np.where(simulated_data["late_delivery_risk"] >= optimal_threshold, "High Risk", "Low Risk")
```

Recommendation for Current Orders - at order level and for all simulations

```python
if not selected_order.empty:
    st.subheader("Recommendations for Selected Order")
    # Extract selected order details
    selected_shipping_mode = selected_order["shipping_mode"].iloc[0]
    selected_payment_type = selected_order["type"].iloc[0]
    selected_risk_score = selected_order["late_delivery_risk"].iloc[0]

    # Provide recommendations
    if selected_risk_score >= optimal_threshold:
        st.write(f"The selected order has a **High Risk** of late delivery (Risk Score: {selected_risk_score:.2f}).")

        # Check and recommend lower-risk shipping mode
        if selected_payment_type in heatmap_data.columns:
            recommended_shipping = heatmap_data[selected_payment_type].idxmin()
            st.write(f"- **Switch to '{recommended_shipping}' shipping mode to reduce risk'.**")

        # Check and recommend lower-risk payment type
        if selected_shipping_mode in heatmap_data.index:
            recommended_payment = heatmap_data.loc[selected_shipping_mode].idxmin()
            st.write(f"- **Switch to '{recommended_payment}' payment type to reduce risk'.**")
    else:
        st.write("The selected order has a **Low Risk** of late delivery. No changes recommended.")
```

```python
# Iterate through late orders
for _, order in late_orders.iterrows():
    current_shipping_mode = order["shipping_mode"]
    current_payment_type = order["type"]

    # Recommend a new shipping mode and payment type
    recommended_shipping = heatmap_data.loc[:, current_payment_type].idxmin()
    recommended_payment = heatmap_data.loc[current_shipping_mode].idxmin()

    # Predict risk scores with updates
    updated_shipping_risk = heatmap_data.loc[recommended_shipping, current_payment_type]
    updated_payment_risk = heatmap_data.loc[current_shipping_mode, recommended_payment]
    updated_both_risk = heatmap_data.loc[recommended_shipping, recommended_payment]

    # Append results
    recommended_changes.append(
        {
            "order_id": order["order_id"],
            "current_shipping_mode": current_shipping_mode,
            "current_payment_type": current_payment_type,
            "recommended_shipping_mode": recommended_shipping,
            "recommended_payment_type": recommended_payment,
            "original_risk": order["late_delivery_risk"],
            "updated_shipping_risk": updated_shipping_risk,
            "updated_payment_risk": updated_payment_risk,
            "updated_both_risk": updated_both_risk,
        }
    )
    new_risk_scores_shipping.append(updated_shipping_risk)
    new_risk_scores_payment.append(updated_payment_risk)
    new_risk_scores_both.append(updated_both_risk)
```