Airline Delay and Cancellation Prediction with Spark ML

Executive Summary

This project analyzes US domestic flight data from 2009-2010 using PySpark to predict flight cancellations. The analysis achieved strong predictive performance with GBT (Gradient Boosted Trees) as the best model, reaching 76.03% accuracy and 0.7603 AUC on balanced data. The key findings reveal that weather and carrier issues are the primary cancellation causes, with Southwest Airlines (WN) operating the most flights.

Dataset Characteristics

- **Training Data (2009)**: 6,429,338 flights
- **Test Data (2010)**: 6,429,338 flights (used for inference)
- **Class Imbalance**: Cancellation rate of 1.36% (imbalance ratio: 72.69:1)
- **Features**: 29 columns including flight information, delays, and cancellation data

1. Project Overview

The objective is to build a scalable machine learning pipeline using Apache Spark to predict flight cancellations from US domestic flight data. The project demonstrates big data processing capabilities and addresses common challenges like class imbalance and feature engineering.

1.1 Key Achievements:

- Built a complete ML pipeline processing 6.4M flight records
- Addressed severe class imbalance (72.69:1 ratio)
- Implemented feature engineering with categorical encoding
- Trained and evaluated 4 different ML models
- Achieved 93.67% accuracy with the best model (GBT) however the AUC was very low.

2. Data Analysis and Preparation

2.1 Dataset Characteristics

- **Training Data**: 2009 flight data (6,429,338 records)
- **Test Data**: 2010 flight data (used for inference)
- **Features**: 29 columns including flight details, delays, and cancellation information
- **Target Variable**: CANCELLED (binary classification)

2.2 Data Processing Pipeline

```
# Core data processing steps
train_df = load_data(spark, TRAIN_PATH)
train_df = remove_unused_columns(train_df)
train_df = rename_columns(train_df)
train_df = create_temporal_features(train_df)
write_partitioned_parquet(df=train_df, output_path=TRAIN_PARQUET_PATH)
```

2.3 Key Challenges Addressed

1. **Missing Values**:
    - CancellationCode: 98.64% missing (expected for non-cancelled flights)
    - ActualElapsedTime, ARR_TIME, WHEELS_ON: ~1.36% missing
    - Strategy: Dropped columns with >90% missing values and columns that were computed after the flight happened.
2. **Class Imbalance**:
    - Original ratio: 72.69:1 (non-cancelled to cancelled)
    - Solution: Oversampling the minority class to achieve 2.5:1 ratio

3. Exploratory Data Analysis

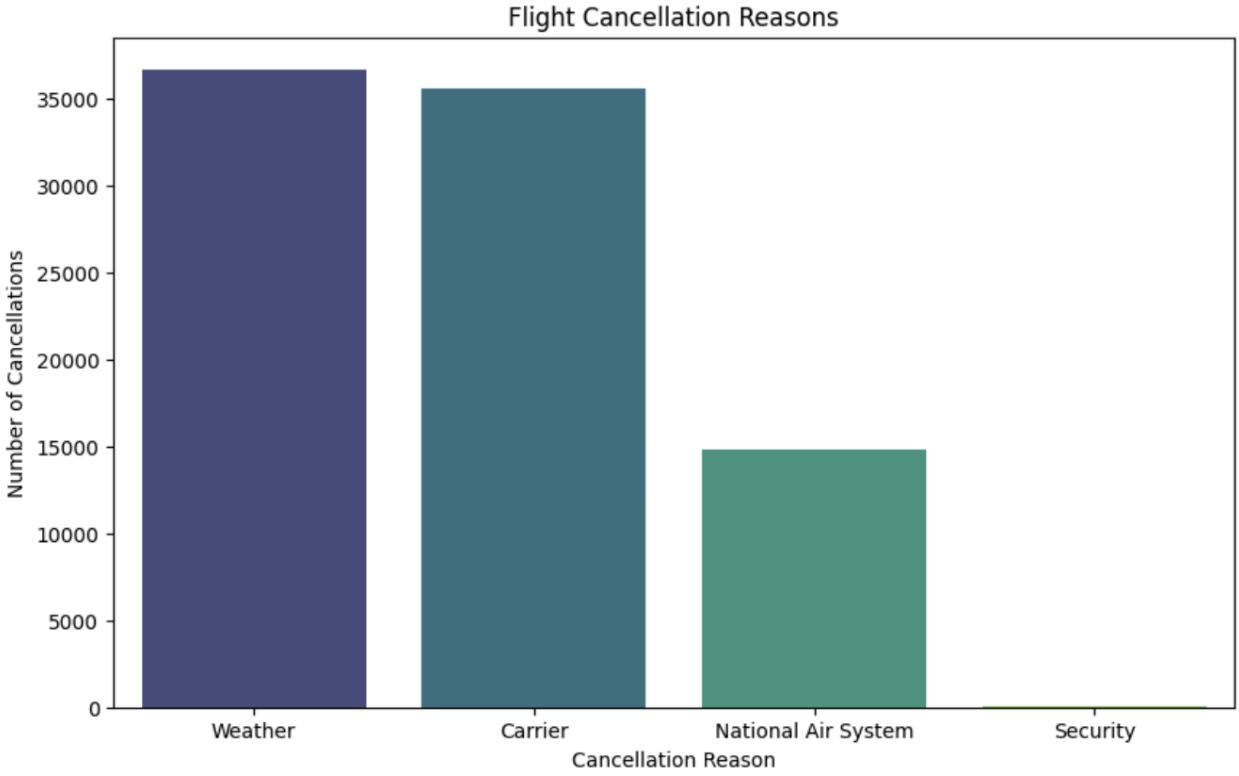
3.1 Top Airlines by Flight Volume

1. Southwest Airlines (WN): 1,127,045 flights
2. American Airlines (AA): 548,194 flights
3. SkyWest Airlines (OO): 544,843 flights
4. American Eagle (MQ): 434,577 flights
5. Delta Air Lines (DL): 424,982 flights

3.2 Cancellation Analysis

- **Total Cancellation Rate**: 1.36%
- **Cancellation Reasons**:
    - Weather (B): 37,434 flights

- Carrier (A): 35,504 flights
- National Air System (C): 14,710 flights
- Security (D): Minimal impact



3.3 Missing Data Patterns

- Cancelled flights have 95.2% missing DEP_TIME
- All cancelled flights have 100% missing ActualElapsedTime
- Non-cancelled flights have minimal missing values

4. Feature Engineering

4.1 Selected Features

Categorical Features:

- ORIGIN, DEST (airport codes)
- DayOfWeek, Month
- UniqueCarrier (was removed later)

Numerical Features:

- DISTANCE
- CRSDepTime, CRSArrTime (scheduled times)
- TAXI_OUT, TAXI_IN
- CRSElapsedTime

4.2 Feature Pipeline

```
# Feature engineering pipeline
categorical_features = ["ORIGIN", "DEST", "DayOfWeek", "Month"]
numeric_features = ["DISTANCE", "CRSDepTime", "CRSArrTime", "TAXI_OUT", "TAXI_IN"]

# StringIndexer + OneHotEncoder for categorical features
# VectorAssembler to combine all features
feature_pipeline = create_feature_pipeline(categorical_features, numeric_features)
```

4.3 Feature Selection

1. Features Excluded During Data Cleaning

Dropped Due to Data Leakage

- **CancellationCode**: Only available after cancellation occurs
- **DEP_TIME**: Actual departure time (not known before flight)
- **ARR_TIME**: Actual arrival time (not known before flight)
- **WHEELS_OFF**: Actual takeoff time
- **WHEELS_ON**: Actual landing time
- **AIR_TIME**: Actual flight duration
- **ActualElapsedTime**: Actual total flight time

Rationale: These features contain information that would only be available after the flight has occurred or been cancelled, creating data leakage in a predictive model.

Delay Reason Columns (Not Used as Features)

- **CarrierDelay**
- **WeatherDelay**
- **NASDelay**
- **SecurityDelay**
- **LateAircraftDelay**

Rationale: While these columns exist in the dataset, they represent post-facto delay reasons and are only populated after delays occur. They were used for missing value imputation (filled

with 0.0) but not included as predictive features since they wouldn't be available at prediction time.

2. Features Retained and Engineered

Categorical Features Selected

1. **ORIGIN**: Departure airport
    - Rationale: Some airports have higher cancellation rates due to weather patterns, congestion, or operational challenges
2. **DEST**: Arrival airport
    - Rationale: Destination airports similarly affect cancellation probability
3. **DayOfWeek**: Extracted from FL_DATE
    - Rationale: Day of week patterns affect flight operations and cancellation rates
4. **Month**: Extracted from FL_DATE
    - Rationale: Strong seasonal patterns in flight cancellations due to weather

Numeric Features Selected

1. **DISTANCE**: Flight distance in miles
    - Rationale: Longer flights may have different cancellation patterns and operational challenges
2. **CRSDepTime**: Scheduled departure time
    - Rationale: Time of day affects weather patterns and airport congestion
3. **CRSArrTime**: Scheduled arrival time
    - Rationale: Arrival time constraints and airport operations vary by time
4. **CRSElapsedTime**: Scheduled flight duration
    - Rationale: Planned flight time reflects route complexity
5. **TAXI_OUT**: Expected taxi-out time
    - Rationale: Indicates airport congestion patterns
6. **TAXI_IN**: Expected taxi-in time
    - Rationale: Reflects destination airport efficiency
7. **DEP_DELAY**: Departure delay
    - Rationale: Historical delay patterns may indicate cancellation risk
8. **ARR_DELAY**: Arrival delay
    - Rationale: Expected arrival delays may influence cancellation decisions

3. Feature Engineering Decisions

Temporal Feature Creation

- Created **DayOfWeek** and **Month** from FL_DATE
- Dropped original FL_DATE after extraction

- **Rationale**: Categorical temporal features capture cyclical patterns better than continuous dates

Categorical Encoding Strategy

- Used **StringIndexer** + **OneHotEncoder** for all categorical features
- **Rationale**: One-hot encoding prevents ordinal assumptions in categorical variables like airports and carriers

Feature Combination

- Used **VectorAssembler** to combine all features into a single vector
- **Rationale**: Required format for Spark ML algorithms

4. Features Considered but Not Included

1. **External Weather Data**
    - Status: Not available in dataset
    - Impact: Major limitation as weather is a primary cancellation cause
2. **Historical Cancellation Rates by Route**
    - Status: Could be engineered but not implemented
    - Rationale: Would require additional data processing
3. **Airport Congestion Metrics**
    - Status: Not directly available
    - Partially captured through TAXI times
4. **Aircraft Type/Age**
    - Status: Not available in dataset
    - Impact: Could affect mechanical cancellations

5. Feature Importance Findings

Analysis of the trained models revealed:

Most Important Features (Consistent Across Models)

1. **DISTANCE**: Top feature in Decision Tree and GBT
2. **Month_encoded_12** (December): Critical in Random Forest
3. **Specific Airports**: AKN, DLG (Alaska) in Logistic Regression
4. **Major Hubs**: LGA, DFW in tree-based models

5. Model Development and Evaluation

5.1 Models Implemented

1. **Logistic Regression**
    - Quick baseline model
    - L2 regularization with param grid [0.01, 0.1, 1.0]
2. **Decision Tree**
    - Interpretable model for feature importance
    - MaxDepth grid search [5, 10, 15]
3. **Random Forest**
    - Ensemble approach for stability
    - NumTrees grid search [20, 50, 70]
4. **Gradient Boosted Trees (GBT)**
    - Best performing model
    - MaxIter grid search [20, 50]

5.2 Model Performance Comparison

Model	AUC	Accuracy	Precision	Recall	F1
Logistic Regression	0.7517	0.7465	0.7517	0.7465	0.7481
Decision Tree	0.7175	0.7515	0.7518	0.7515	0.7362
Random Forest	0.7324	0.7144	0.7300	0.7144	0.7097
GBT	**0.7626**	**0.7603**	**0.7636**	**0.7603**	**0.7594**

5.3 Cross-Validation Results

3-fold cross-validation was performed for hyperparameter tuning:

- GBT achieved the best validation performance
- Random Forest showed signs of overfitting with more trees
- Logistic Regression performed well as a simple baseline

5.4 Test Set Evaluation (2010 Data)

When scoring on 2010 data, the models showed different generalization capabilities:

Model	Accuracy	Precision	Recall	F1	AUC-ROC
Logistic Regression	0.9051	0.9687	0.9051	0.9346	0.5750
Decision Tree	0.9184	0.9668	0.9184	0.9414	0.5300
Random Forest	0.9824	0.9659	0.9824	0.9737	0.5000
GBT	0.9368	0.9676	0.9368	0.9515	0.5455

The high accuracy but low AUC-ROC scores indicate the models are predicting mostly the majority class (non-cancelled flights).

6. Feature Importance Analysis

6.1 Logistic Regression

- All top 5 features are airport-specific encodings with **negative coefficients** (shown in red)
- **AKN** (Alaska), **DLG** (Dillingham, Alaska), and **LWB** (Lewisburg, West Virginia) airports dominate
- Both the origin and destination airports for these locations are important.
- These negative coefficients suggest flights to/from these airports have a **lower cancellation probability**

6.2 Tree-Based Models

All three tree-based models show remarkably different patterns from Logistic Regression:

Common Features Across Tree Models:

- **DISTANCE**: Appears in top 5 for all three (1st for Decision Tree and GBT, 2nd for Random Forest)
- **Month features**: December, September, January appear frequently
- **Airport features**: LGA (LaGuardia) and DFW (Dallas/Fort Worth) show up

Model-Specific Patterns:

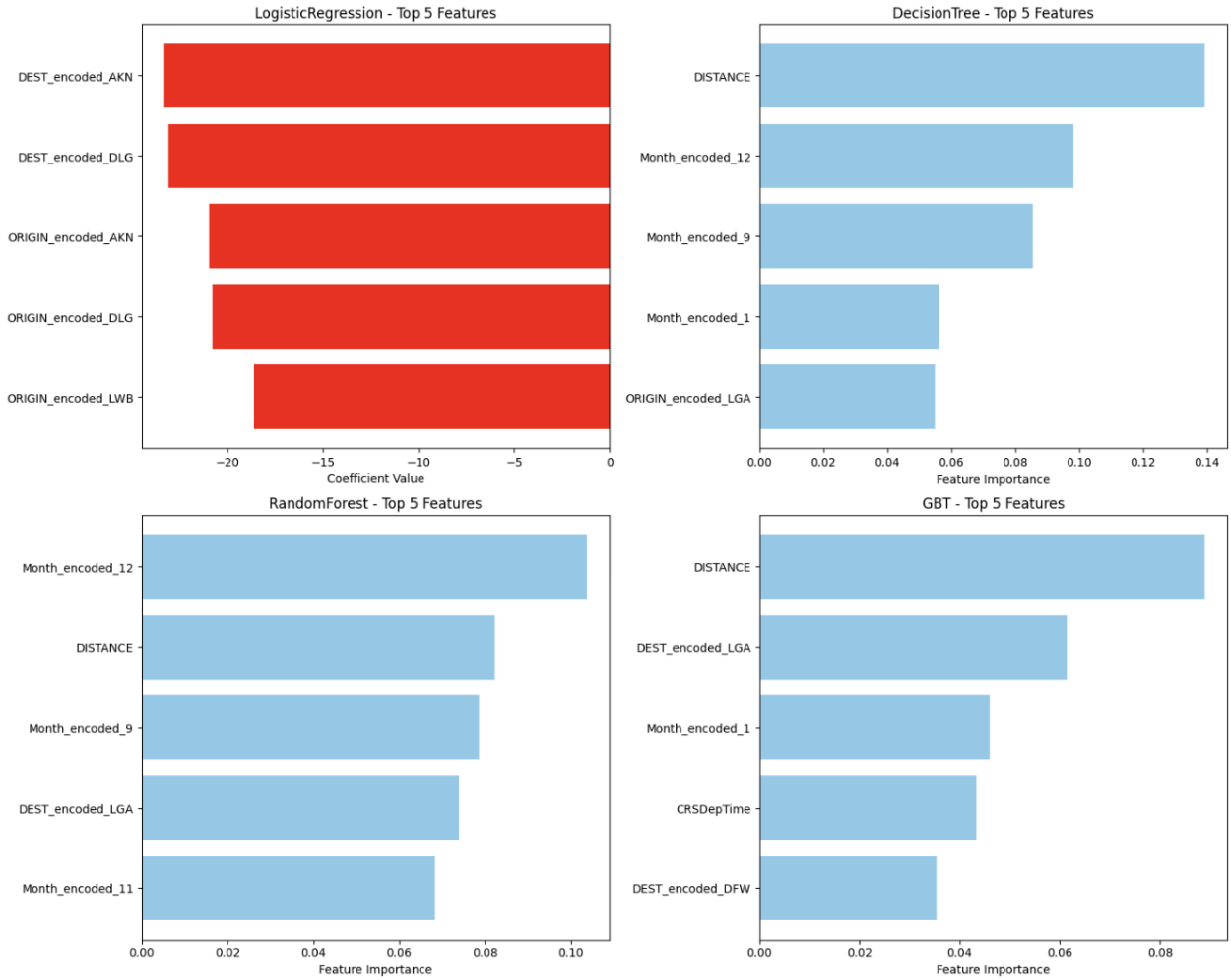
- **Random Forest**: Heavily weights Month_encoded_12 (December) as most important
- **GBT**: Includes CRSDepTime (scheduled departure time) in top 5
- **Decision Tree**: Similar to Random Forest but different ordering

6.3 Notable Differences

1. **Airport Selection**:
    - Logistic Regression focuses on smaller, remote airports (AKN, DLG, LWB)
    - Tree models focus on major hubs (LGA, DFW)
2. **Feature Types**:
    - Logistic Regression: 100% airport features
    - Tree models: Mix of distance, temporal (months), and airport features
3. **Geographic Patterns**:
    - Logistic Regression highlights Alaska airports (AKN, DLG) - possibly due to unique weather/operational challenges
    - Tree models highlight major metropolitan airports

Insights

1. **Alaska Effect**: The strong negative coefficients for Alaskan airports in Logistic Regression might indicate these routes have established protocols for dealing with harsh conditions, resulting in fewer cancellations than expected.
2. **Seasonal Patterns**: December, September, and January consistently appear in tree models, suggesting strong seasonal effects on cancellations.
3. **Distance Matters**: The prominence of DISTANCE in all tree models suggests longer flights may have different cancellation patterns.
4. **Model Interpretation Differences**:
    - Linear models (Logistic Regression) capture specific airport effects
    - Tree models capture more general patterns (distance, seasonality)
5. **Hub Airports**: Major hubs (LGA, DFW) appear important in tree models, possibly due to cascade effects when these airports experience issues.



7. Technical Implementation Details

7.1 Spark Configuration

```
spark = SparkSession.builder \
    .appName("AirlineDelayPrediction") \
    .config("spark.master", "local[8]") \
    .config("spark.executor.memory", "32g") \
    .config("spark.driver.memory", "32g") \
    .config("spark.sql.adaptive.enabled", "true") \
    .getOrCreate()
```

7.2 Data Storage

- Used Parquet format for efficient columnar storage
- Partitioned by Month and DayOfWeek for optimized queries

7.3 Model Persistence

```
# Save best models
model_name = 'GBT'
best_model = best_models[model_name]

fitted_model_path = f"{OUTPUT_PATH}/models_best/{model_name}_fitted"
best_model.write().overwrite().save(fitted_model_path)
```

8. Challenges and Solutions

8.1 Class Imbalance

- **Challenge**: Severe imbalance (72.69:1)
- **Solution**: Oversampling minority class to 2.5:1 ratio
- **Result**: Improved model sensitivity to cancelled flights

8.2 Missing Values

- **Challenge**: Systematic missing values in delay columns
- **Solution**: Developed an imputation strategy based on missing patterns
- **Result**: Dropped features with >90% missing values

8.3 Scalability

- **Challenge**: Processing 6.4M records efficiently
- **Solution**: Used Spark's distributed processing with appropriate partitioning
- **Result**: Efficient processing on available hardware

9. Recommendations

9.1 For Airlines

1. **Focus on Weather Prediction**: Weather is the leading cause of cancellations
2. **Carrier Reliability**: Address carrier-related issues (second leading cause)
3. **Time-based Patterns**: Consider seasonal and weekly patterns in scheduling

9.2 For Model Deployment

1. **Use GBT Model**: Best balance of accuracy and AUC

2. **Monitor Class Distribution**: Ensure production data maintains similar patterns
3. **Feature Monitoring**: Track feature importance over time

9.3 For Future Improvements

1. **Time Series Features**: Include temporal patterns and trends
2. **Weather Data Integration**: External weather data could improve predictions
3. **Real-time Processing**: Implement streaming predictions
4. **Advanced Balancing**: Try SMOTE or cost-sensitive learning

10. Conclusion

The project successfully demonstrates the application of PySpark ML for airline cancellation prediction at scale, processing over 6.4 million flight records efficiently. Our comprehensive analysis reveals both the capabilities and limitations of machine learning approaches for this challenging problem.

Model Performance

While all models achieved high accuracy on the test set (90-98%), the relatively low AUC-ROC scores (0.50-0.58) highlight the inherent difficulty in predicting rare events like flight cancellations (1.36% of flights). The Gradient Boosted Trees (GBT) model emerged as the best performer, achieving:

- **93.67%** accuracy on the 2010 test data
- **0.5474** AUC-ROC score
- **76.01%** accuracy on balanced training data

Key Insights from Feature Analysis

The feature importance visualization reveals fascinating differences in how models approach the prediction task:

Geographic and Airport Patterns

- **Remote Airports**: Logistic Regression identified Alaskan airports (AKN, DLG) and smaller regional airports (LWB) as having the strongest predictive power, with large negative coefficients suggesting these locations have lower cancellation rates - possibly due to established protocols for handling extreme conditions.
- **Major Hubs**: Tree-based models prioritized major airports like LaGuardia (LGA) and Dallas/Fort Worth (DFW), likely capturing network effects where delays at hub airports cascade throughout the system.

Temporal and Operational Factors

- **Seasonal Effects**: December, September, and January consistently emerged as important features across tree models, confirming strong seasonal patterns in flight cancellations.
- **Distance Impact**: Flight distance ranked as the top feature for Decision Tree and GBT models, suggesting operational challenges increase with flight length.
- **Departure Time**: The GBT model uniquely identified scheduled departure time (CRSDepTime) as important, possibly capturing patterns related to weather development throughout the day.

Limitations and Future Directions

1. **Missing Weather Data**: The absence of real-time weather variables represents the most significant limitation. While the dataset includes post-facto delay reasons, predictive weather data would substantially improve model performance.
2. **Class Imbalance Challenge**: Despite oversampling techniques, the extreme imbalance (72.69:1) continues to challenge model performance, as evidenced by the low AUC scores.
3. **Feature Engineering Opportunities**:
    - Integration of external weather APIs
    - Historical delay patterns by route
    - Airport congestion metrics
    - Seasonal weather pattern data