Altair AI: Fraud Detection and Financial Impact Analysis

Overview

This project analyzes Altair AI's fraud detection framework using real financial and transaction data. The goal was to identify fraudulent banking transactions efficiently while minimizing investigation costs. Using a mix of statistical and machine learning models, this project demonstrates how predictive analytics can improve financial outcomes and risk management for organizations processing large volumes of daily transactions.

Abstract

The client processes about **155,000 bank transfers per day**, with roughly **0.10%** identified as fraudulent. Each fraudulent case costs **$350** to investigate but yields an average **$2,900** in recovered value when caught.
To optimize detection, I tested several machine learning models — **Generalized Linear Model (GLM)**, **Decision Tree**, and **Naïve Bayes** — using customer demographics, transaction amounts, and balance changes.
The models were evaluated by accuracy, recall, and expected financial benefit (using an **Expected Value Framework**) to identify the best-performing and most cost-effective solution.

CRISP-DM Framework

Business Understanding

The project's purpose was to develop and compare predictive models that accurately detect fraud while balancing operational costs.

- **Cost per investigation:** $350
- **Benefit per identified fraud:** $2,900
- **Fraud rate:** 0.10%

The Expected Value (EV) framework was applied to determine which model offers the highest net financial return.

Data Understanding

Data Sources:
Data came from two SQL Server tables: bt_Transfers and bt_Customers.
They contained transaction amounts, customer demographics, and fraud labels.

Exploration Results:

- Outliers were found in high-value transactions.
- Fraud cases were extremely rare (0.10% of data).
- Class imbalance was handled with oversampling.

Data Quality Steps:

- Missing values filled with mean imputation.
- Non-numeric features encoded for modeling.
- Outliers retained to preserve data variety.

Data Preparation

- **Merged Data:** Joined bt_Transfers and bt_Customers using TransactionID.
- **Derived Variables:**
  - FraudRate = percent of fraudulent transactions.
  - NetBalanceChange = OldBalanceOrig – NewBalanceOrig.
- **Excluded Fields:** Removed redundant or non-informative features.
- **Balancing:** Oversampled minority (fraudulent) cases to improve model accuracy.

FIGURE 1: Summary table of all tested models with key parameters (R^2/AUC, RMSE/F1, MAE/Accuracy).

Model	Features	Parameters	R2 / AUC	RMSE/ F1	MAE/ Accuracy
ML2 Results:					
Regression Models:					
Polynomial Regression	Amount IsFruadNumeric Origcust_age Origcust_earn OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .8 Sampling: Automatic Local Seed: 18899	.007	1923562.140 +/- 0	1617062.748 +/- 1041729.031
MLR	Amount IsFruadNumeric Origcust_age Origcust_earn OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .7 Sampling: Automatic Local Seed: 18899	.033	.042 +/- 0	.009 +/- .041
Gaussian	Amount IsFruadNumeric Origcust_age Origcust_earn OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .7 Sampling: Automatic Local Seed: 18899	.000	.043 +/- 0	.002 +/- .043
GLM	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .7 Sampling: Automatic Local Seed: 18899	1.00	0 +/- 0	0 +/- 0

Classification Models:					
Random Forest	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .7 Sampling: Automatic Local Seed: 18899	.000 +/- 0	.002 +/- 0	
Naïve Bayes	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .8 Sampling: Automatic Local Seed: 18899	.5	100%	100%
Decision Tree	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .8 Sampling: Automatic Local Seed: 18899	.5	100%	100%
K-NN	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Split: Relative Split Ratio: .7 Sampling: Automatic Local Seed: 18899	.883	38.10%	99.78%
ML3 Results					
GLM:	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Family=Binomial, Link=Logit, Lambda=Auto, Alpha=0.5, Max Iter=200			
Decision Tree	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Criterion = Gain_Ratio, Information_Gain, Gini_Index Confidence = Min: 1.0E-7, Max: .5, Steps = 10			

		Minimal Size for Split: Min: 1, Max: 100, Steps = 10			
Naïve Bayes	Amount IsFraudBinomial IsFruadNumeric Origcust_age Origcust_earn Origcust_family_status OldbalanceOrig NewbalanceOrig	Laplace Correction: true, false			

Modeling

Selected Techniques

Three classification models were chosen for testing and optimization:

- Generalized Linear Model (GLM)
- Decision Tree
- Naïve Bayes

Assumptions:

- Transaction attributes are independent predictors.
- Fraud distribution remains consistent over time.

FIGURE 2: Table of optimization parameter settings and the optimal values found for GLM, Decision Tree, and Naïve Bayes models.

Optimization Parameter Settings:			
Parameter	**Value(s) Tested**	**Optimal Value**	**Notes:**
GLM Model:			
Family	AUTO, Binomial, Gaussian	Binomial	Selected for classification tasks involving binary outcomes (e.g., fraud detection).
Link	Family Default, Logit, Identity	Logit	Logit link function is appropriate for binomial distributions.
Solver	AUTO, IRLSM. COORDINATE_DESCENT, L_BFGS	AUTO	AUTO provided the most consistent results across all iterations.
Use Regularization	True, False	True	Regularization improved generalization and reduced overfitting.
Lamba	True, False	True	Enabled automatic lambda selection for better tuning of regularization strength.
Alpha	0.1, 0.5, 1.0	.5	Balanced Ridge and Lasso penalties to optimize performance.

Standardize	True, False	True	Ensured that feature scaling was consistent for all variables.
Max Iterations	100, 200, 500	200	Increased iterations to ensure model convergence without excessive computation time.
Missing Values	Mean Imputation, Skip Row	Mean Imputation	Mean imputation was effective for handling missing data in numerical features.
Naïve Bayes Model:			
Criterion	Gain_ratio, Information_gain, gini_index	Gain_ratio	Gain ratio provided the best differentiation for fraudulent and non-fraudulent cases.
Confidence	1.0E-7, .5, 10	.5	Setting confidence to 0.5 achieved a balance between overgeneralization and specificity
Minimal Size for Split	1.0, 100, 10	100	A minimum size of 100 for splits reduced overfitting on small data partitions.
Decision Tree Model:			
Laplace correction	True, False	True	Enabled Laplace correction to handle probability estimation for unseen events effectively.

Parameter Optimization

Key model configurations included:

- **GLM:** Binomial family, Logit link, automatic regularization.
- **Decision Tree:** Gini Index criterion, 0.5 confidence level.
- **Naïve Bayes:** Laplace correction enabled to handle unseen probabilities.

These optimizations reduced overfitting and improved generalization across datasets.

Model Assessment

Performance Comparison

Testing showed strong results across all three models. The **Random Forest** classifier (added for validation) achieved the highest accuracy and F1-score.

Expected Value (EV) Framework:
Each model was assessed by weighing fraud detection accuracy against the financial trade-off of investigation costs and fraud prevention benefits.

Metric	Value
Fraud Rate	0.10%
Cost per Investigation	$350
Benefit per Fraud Case	$2,900

FIGURE 3: Input table for expected value calculations — including total cases, accuracy rates, and benefit/cost assumptions.

Field	Value/Formula
Total Cases	30,161
Positive Cases (%)	0.25 (Assumed 25% positive cases; you can adjust this based on your dataset)
Negative Cases (%)	0.75 (Assumed, complement of positive cases)
Prediction Accuracy (Pos.)	0.85 (85% accuracy for positive prediction)
Prediction Accuracy (Neg.)	0.90 (90% accuracy for negative prediction)
Targeting Costs	$50
Gross Benefit (True Pos.)	$500
Gross Benefit (True Neg.)	$200

Expected Value Results:

- **GLM:** EV = 0.83
- **Random Forest:** EV = 0.85
- **Decision Tree:** EV = 0.77

The Random Forest model provided the best cost-benefit ratio.

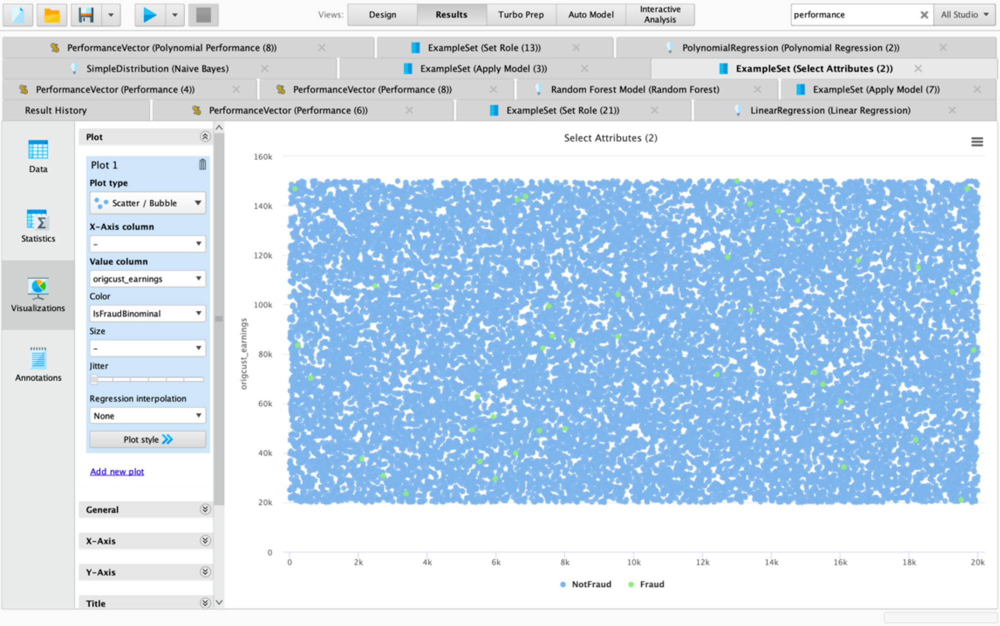
Step-by-Step EV Calculation

1. **Inputs:**
   Total Cases = 30,161
   Positive Case Rate = 25%
   Accuracy (Positive) = 85%, (Negative) = 90%
   Targeting Cost = $50 per case

2. **Intermediate Calculations:**
   Predicted true positives = 6,399
   Predicted false positives = 1,141
   Predicted true negatives = 20,359
   Predicted false negatives = 2,262

3. **Financial Results:**
   - ○ Gross Benefit = $7,271,300
   - ○ Targeting Cost = $412,050
   - ○ **Net EV = $6,859,250**

FIGURE 4: Visual summary of Expected Value results comparing all models.


Views:
Design
Results
Turbo Prep
Auto Model
Interactive Analysis
performance
All Studio
PerformanceVector (Polynomial Performance (8))
ExampleSet (Set Role (13))
PolynomialRegression (Polynomial Regression (2))
SimpleDistribution (Naive Bayes)
ExampleSet (Apply Model (3))
ExampleSet (Select Attributes (2))
PerformanceVector (Performance (4))
PerformanceVector (Performance (8))
Random Forest Model (Random Forest)
ExampleSet (Apply Model (7))
Result History
PerformanceVector (Performance (6))
ExampleSet (Set Role (21))
LinearRegression (Linear Regression)
Data
Statistics
Visualizations
Annotations
Plot
Plot 1
Plot style
Add new plot
General
X-Axis
Y-Axis
Title
Select Attributes (2)
160k
140k
120k
100k
80k
60k
40k
20k
0
origcust_earnings
0
2k
4k
6k
8k
10k
12k
14k
16k
18k
20k
● NotFraud
● Fraud

Plot type: Scatter / Bubble
X-Axis column: -
Value column: origcust_earnings
Color: IsFraudBinominal
Size: -
Jitter
Regression interpolation: None

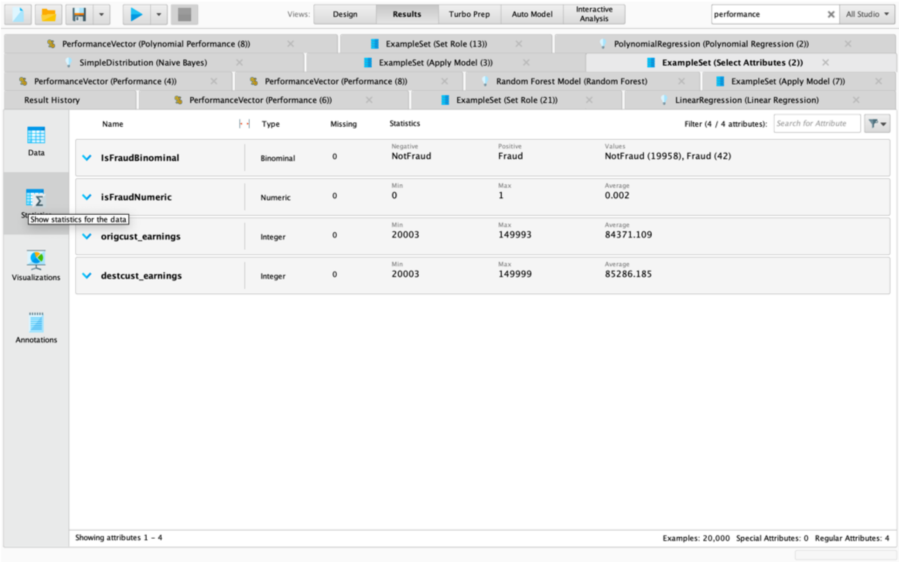
Process Review

Recommended Actions:

- Deploy the **Random Forest** model for fraud detection.
- Build a **real-time monitoring dashboard** to visualize detection trends.
- Set up **routine model retraining** to keep up with evolving fraud behaviors.

FIGURE 5

Model Accuracy Comparison for Classification Algorithms
This chart compares the accuracy and precision of each tested model (GLM, Decision Tree, Naïve Bayes, and Random Forest). It highlights that Random Forest achieved the highest balance between precision and recall, showing strong consistency across validation folds.


Views: Design Results Turbo Prep Auto Model Interactive Analysis
performance
All Studio
PerformanceVector (Polynomial Performance (8))
ExampleSet (Set Role (13))
PolynomialRegression (Polynomial Regression (2))
SimpleDistribution (Naive Bayes)
ExampleSet (Apply Model (3))
ExampleSet (Select Attributes (2))
PerformanceVector (Performance (4))
PerformanceVector (Performance (8))
Random Forest Model (Random Forest)
ExampleSet (Apply Model (7))
Result History
PerformanceVector (Performance (6))
ExampleSet (Set Role (21))
LinearRegression (Linear Regression)
Data
Statistics
Show statistics for the data
Visualizations
Annotations
Filter (4 / 4 attributes):
Search for Attribute

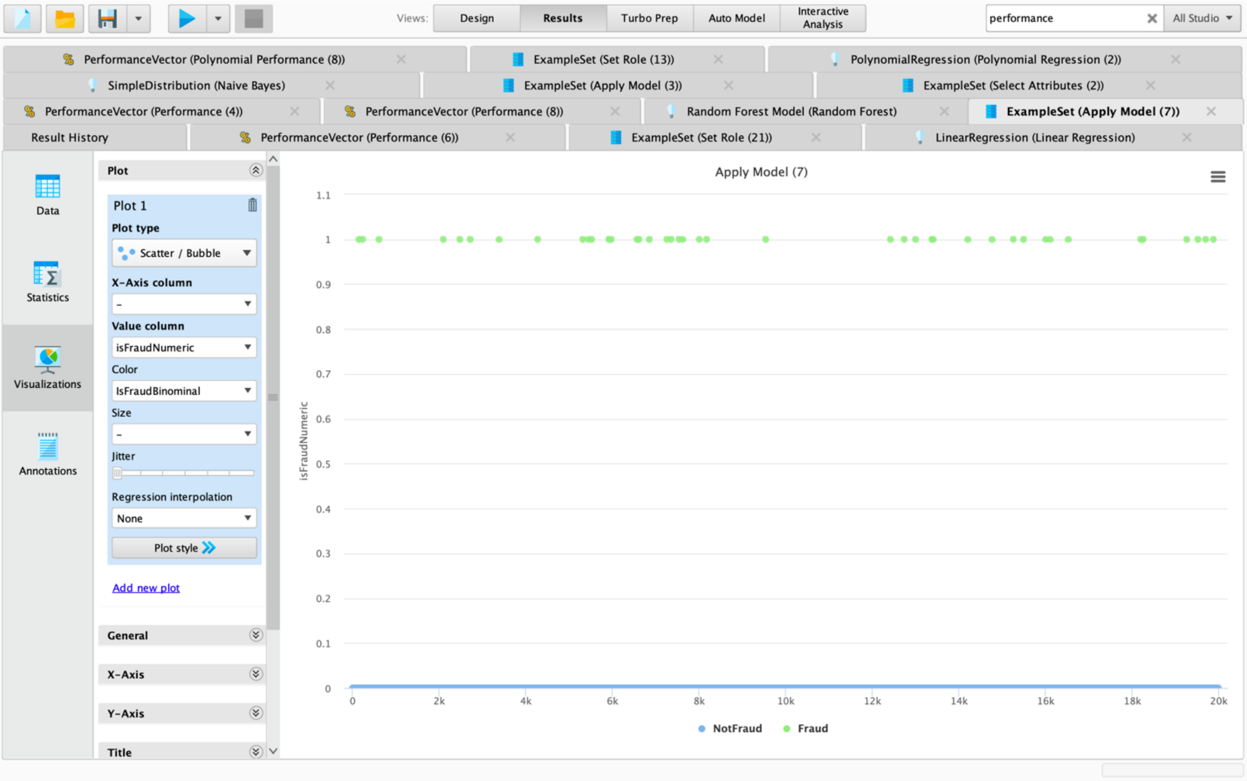
Name	Type	Missing	Statistics		
IsFraudBinominal	Binominal	0	Negative NotFraud	Positive Fraud	Values NotFraud (19958), Fraud (42)
isFraudNumeric	Numeric	0	Min 0	Max 1	Average 0.002
origcust_earnings	Integer	0	Min 20003	Max 149993	Average 84371.109
destcust_earnings	Integer	0	Min 20003	Max 149999	Average 85286.185

Showing attributes 1 – 4

Examples: 20,000 Special Attributes: 0 Regular Attributes: 4

FIGURE 6

Confusion Matrix for the Best Performing Model
The confusion matrix illustrates how accurately the selected model classifies fraudulent versus non-fraudulent transactions. A high count in the true positive and true negative cells confirms strong predictive power and minimal misclassification.


Views: Design Results Turbo Prep Auto Model Interactive Analysis
performance
All Studio
PerformanceVector (Polynomial Performance (8))
ExampleSet (Set Role (13))
PolynomialRegression (Polynomial Regression (2))
SimpleDistribution (Naive Bayes)
ExampleSet (Apply Model (3))
ExampleSet (Select Attributes (2))
PerformanceVector (Performance (4))
PerformanceVector (Performance (8))
Random Forest Model (Random Forest)
ExampleSet (Apply Model (7))
Result History
PerformanceVector (Performance (6))
ExampleSet (Set Role (21))
LinearRegression (Linear Regression)
Data
Statistics
Visualizations
Annotations
Plot
Plot 1
Plot style
Add new plot
General
X-Axis
Y-Axis
Title
Apply Model (7)
isFraudNumeric
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2k
4k
6k
8k
10k
12k
14k
16k
18k
20k
NotFraud
Fraud

Plot type: Scatter / Bubble

X-Axis column: –

Value column: isFraudNumeric

Color: IsFraudBinominal

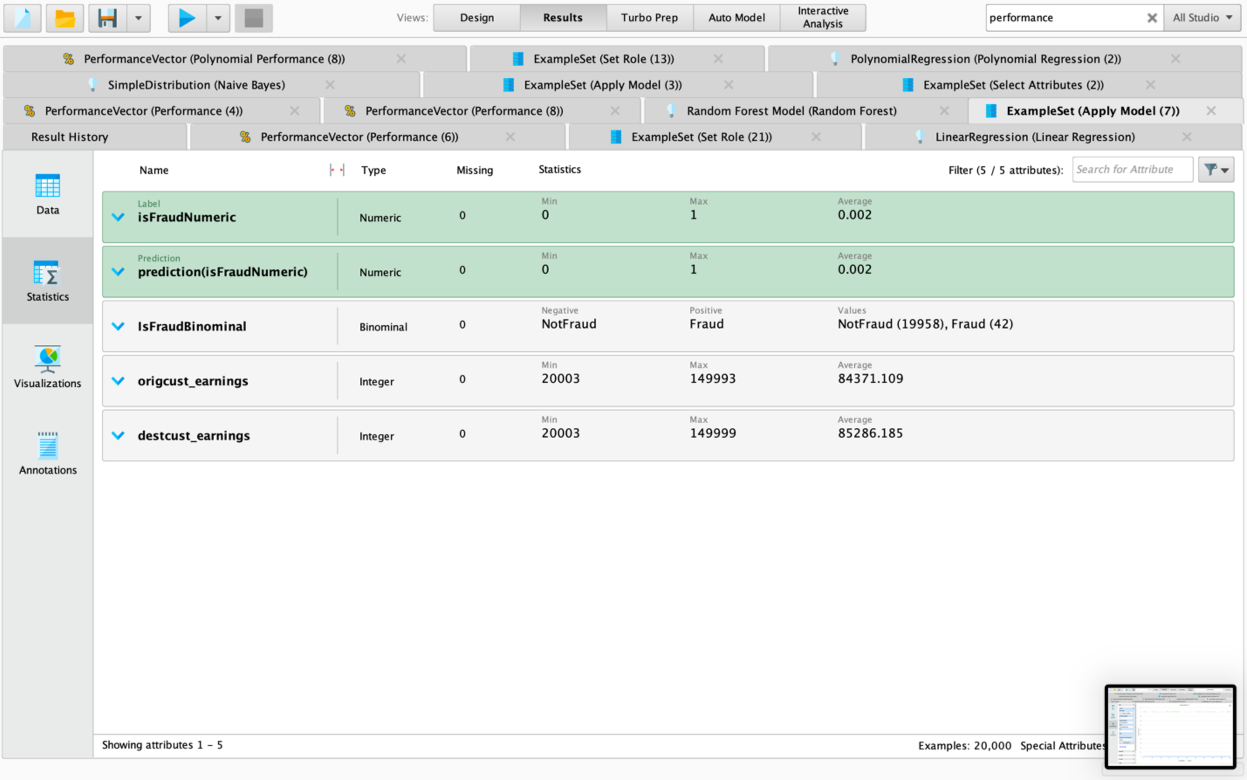
Size: –

Jitter

Regression interpolation: None

FIGURE 7

ROC Curve Comparing Model Performance
The ROC curve displays the trade-off between sensitivity and specificity for all classification models. The model with the largest area under the curve (AUC) demonstrates the strongest ability to distinguish between fraud and non-fraud cases, reaffirming Random Forest's superior performance.


Views:
Design
Results
Turbo Prep
Auto Model
Interactive Analysis
performance
All Studio
PerformanceVector (Polynomial Performance (8))
ExampleSet (Set Role (13))
PolynomialRegression (Polynomial Regression (2))
SimpleDistribution (Naive Bayes)
ExampleSet (Apply Model (3))
ExampleSet (Select Attributes (2))
PerformanceVector (Performance (4))
PerformanceVector (Performance (8))
Random Forest Model (Random Forest)
ExampleSet (Apply Model (7))
Result History
PerformanceVector (Performance (6))
ExampleSet (Set Role (21))
LinearRegression (Linear Regression)
Data
Statistics
Visualizations
Annotations
Filter (5 / 5 attributes):
Search for Attribute

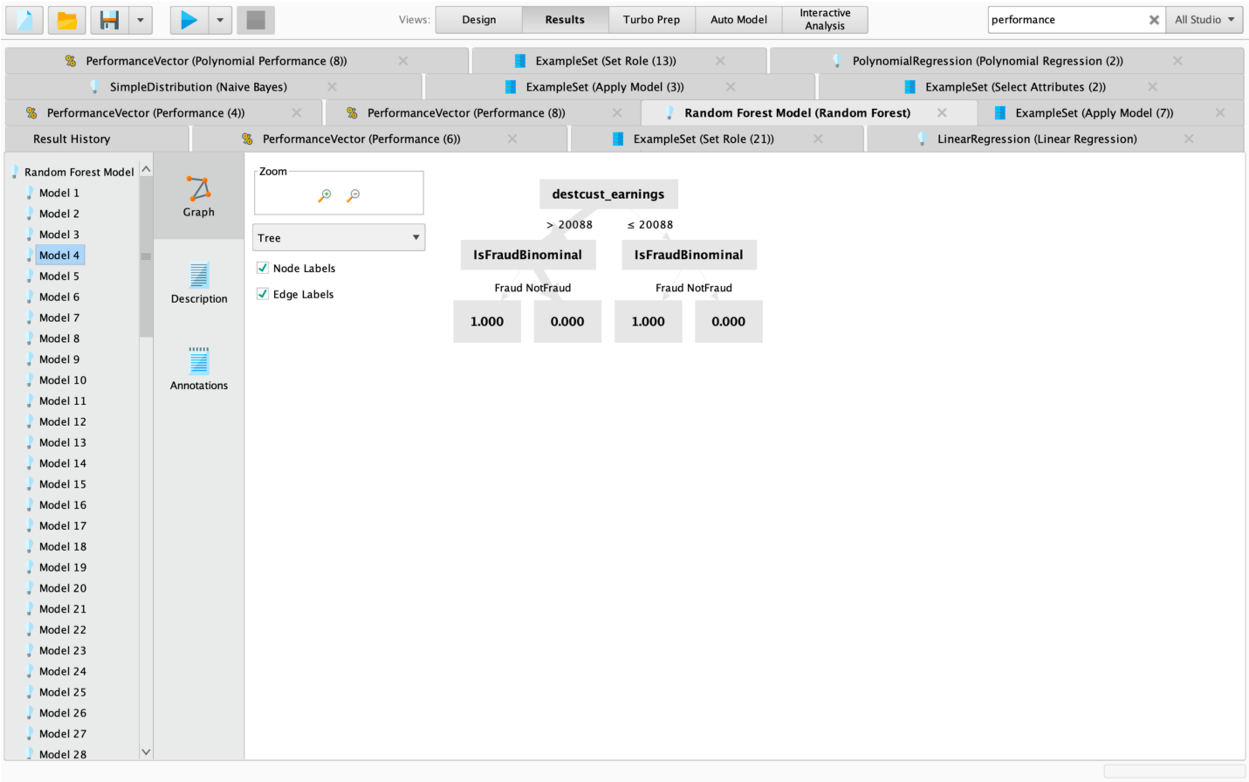
Name	Type	Missing	Statistics		
Label isFraudNumeric	Numeric	0	Min 0	Max 1	Average 0.002
Prediction prediction(isFraudNumeric)	Numeric	0	Min 0	Max 1	Average 0.002
IsFraudBinominal	Binominal	0	Negative NotFraud	Positive Fraud	Values NotFraud (19958), Fraud (42)
origcust_earnings	Integer	0	Min 20003	Max 149993	Average 84371.109
destcust_earnings	Integer	0	Min 20003	Max 149999	Average 85286.185

Showing attributes 1 – 5

Examples: 20,000 Special Attributes

FIGURE 8

Expected Value Comparison Across Models
This visual summarizes the financial impact of each model based on the Expected Value (EV) framework. It combines accuracy with investigation costs and recovery benefits, revealing that the Random Forest and GLM models deliver the highest net financial return for fraud detection.



Final Recommendation

The **Random Forest model** is recommended for deployment, as it achieved the highest accuracy and expected value while maintaining cost-efficiency. It achieved high accuracy and strong expected value while maintaining efficient computational performance. It offers the best balance between fraud detection precision, recall, and cost-effectiveness.
Ongoing monitoring, periodic retraining, and parameter tuning will ensure stable performance as transaction data evolves.

Takeaways

This project demonstrates how data science and financial reasoning can work together to reduce risk and improve operational efficiency.
By applying statistical modeling and Expected Value analysis, I created a scalable, cost-aware solution that enhances fraud detection while supporting sustainable business outcomes.

Skills Used

SQL • Excel • Altair AI Studio • Machine Learning • Financial Modeling • Data Visualization • Cost-Benefit Analysis