BA 5200 - Information Systems Management and Data Analytics

Healthcare Provider Fraud Detection

A Project Report

Submitted by

Lenin Goud Athikam

Saketha Kusu

Varshitha Reddy Davarapalli

Karunakar Uppalapati

Rashmitha Eri

Kavya Pachchava

Michigan Tech

Contents

Executive Summary

Healthcare fraud leads to billions of dollars in losses annually, affecting both financial systems and patient care. This project investigates Medicare claims data to identify patterns and behaviors that signal fraudulent activities by healthcare providers. We integrated multiple datasets, covering inpatient, outpatient, and beneficiary details, to create a comprehensive view of provider activity. Through thorough data preprocessing and feature engineering, we developed key indicators such as claim duration, chronic condition counts, and reimbursement values.

Using exploratory data analysis, we uncovered trends in age distribution, health conditions, insurance coverage, and racial demographics that informed our modeling decisions. For classification, we chose a Decision Tree model due to its interpretability and suitability for structured healthcare data. The model was optimized using GridSearchCV and evaluated using metrics such as recall and ROC-AUC, prioritizing the identification of true fraud cases. Our final model achieved 90% accuracy and 88% recall.

This report presents a full analysis pipeline from data acquisition to predictive modeling. It provides a strong foundation for deploying machine learning in fraud detection workflows, supporting faster audits and improved decision-making by healthcare investigators.

Introduction

Healthcare fraud presents a significant challenge to modern health systems, leading to increased costs, inefficiencies, and erosion of public trust. Common forms of fraud include billing for services not rendered, exaggerating service complexity, and misrepresenting diagnoses to inflate reimbursements. These actions compromise both the financial integrity of healthcare systems and the quality of care delivered to patients.

To address this issue, our project leverages Medicare claims data and applies data science techniques to detect potentially fraudulent providers. We focus on integrating multiple datasets, spanning inpatient, outpatient, and beneficiary-level details, to capture a full picture of provider behavior across medical, demographic, and financial dimensions.

This report outlines a comprehensive approach that includes data merging, preprocessing, feature engineering, and machine learning modeling. By using exploratory data analysis and building interpretable models like decision trees, we aim to identify key patterns and indicators of fraud. The ultimate goal is to assist healthcare auditors and regulators in efficiently prioritizing high-risk cases for further investigation.

Project Objective

The primary objective of this project is to build a reliable and interpretable machine learning model to detect potentially fraudulent healthcare providers using Medicare claims data. Healthcare fraud undermines both financial sustainability and quality care delivery; hence, identifying suspicious provider behavior is critical for system efficiency and integrity.

This project is designed with the following specific goals:

- **Integrate multiple datasets** (inpatient, outpatient, and beneficiary data) to create a comprehensive view of provider behavior.

- **Engineer meaningful features** such as claim duration, chronic condition counts, same physician flags, and financial metrics to capture fraud-indicative patterns.

- **Conduct exploratory data analysis (EDA)** to understand trends in age distribution,Race distribution, insurance coverage, comorbidities, and claim characteristics.

- **Develop and tune a classification model**, primarily using Decision Trees, to flag suspicious claim patterns with high recall and accuracy.

- **Ensure interpretability and real-world applicability** of the model by using transparent algorithms, making it easy for healthcare auditors to adopt and act on predictions

Data Understanding

In this project, we analyzed Medicare claim data using four main datasets, each contributing a different dimension of information necessary for identifying fraudulent providers.

1. Inpatient Data – InpatientData.csv

- Contains information on claims submitted by providers for **admitted patients**.
- Includes:
    - **Identifiers**: ClaimID, Provider, BeneID
    - **Temporal Details**: ClaimStartDt, ClaimEndDt, AdmissionDt, DischargeDt
    - **Medical Codes**: Up to 10 diagnosis codes and 6 procedure codes
    - **Financials**: InscClaimAmtReimbursed, DeductibleAmtPaid

2. Outpatient Data – OutpatientData.csv

- Captures claims for **non-admitted (outpatient) visits**.
- Follows a similar structure as the inpatient data but includes same-day and short-duration healthcare interactions.
- Features are mostly aligned with those in the inpatient dataset.

3. Beneficiary Summary File – BeneficiaryData.csv

- Provides **demographic and clinical condition information** about beneficiaries.

- Includes:

    - **Personal Data**: DOB, DOD, Gender, Race

    - **Insurance Coverage**: NoOfMonths_PartACov, NoOfMonths_PartBCov

    - **Chronic Conditions**: 13 indicators (e.g., ChronicCond_HeartFailure, ChronicCond_Diabetes, etc.)

4. Fraud Labels – Train-Labels.csv

- Contains the **target variable** PotentialFraud indicating whether a provider is flagged as fraudulent (Yes) or not (No).

- This dataset serves as the foundation for the supervised learning task.

Why We Merged the Data

Merging datasets was a crucial step in our analysis pipeline. Each dataset provided a different but complementary perspective on healthcare claims, and bringing them together allowed us to build a unified, patient-centered view of provider activity.

1. To Combine Clinical, Demographic, and Financial Context

- **The inpatient and outpatient datasets c**ontained detailed records of services provided, including medical codes and financial reimbursement information.

- The **beneficiary dataset** included essential demographic data such as age, gender, and chronic health conditions, which provide vital clinical context.

- The **label dataset** (Train-Labels.csv) was needed to supervise the machine learning task by indicating which providers were flagged for fraud.

Without merging, each of these datasets would have remained isolated, preventing us from understanding the full picture of how provider services interact with patient profiles and reimbursement behaviors.

2. To Enable Feature Engineering Across Domains

By merging:

- We were able to compute features like Claim_Duration (from claim start and claim end dates)

- Calculate patient Age (from DOB and DOD),

- Count chronic conditions for each patient

- Aggregate claim statistics per provider.

These cross-dataset features were essential for training a model that reflects real-world fraud patterns.

3. To Ensure One Record per Patient-Claim-Provider Unit

Merging allowed us to structure the data so each row represented a comprehensive healthcare interaction—linking a patient (BeneID), their provider, clinical history, and financial claim. This alignment made the data suitable for supervised learning and meaningful EDA.

4. To Retain All Relevant Information Using Outer Joins

We used outer joins to avoid losing any records

- Some providers had only inpatient or outpatient data.
- Some patients appeared in only one type of claim file.
- Outer joins helped us preserve as much information as possible, which was especially important due to the already limited number of fraud-labeled providers.

We Used a Right Join to

- To retain all providers present in the fraud label dataset (Train-Labels.csv), even if corresponding claim data was missing.
- Ensures no labeled (fraud or non-fraud) provider is excluded, which is crucial for model training.
- Helps preserve edge cases or sparse claim records, which may still contain fraud indicators.
- Prioritizes the fraud label as the anchor dataset while merging, maintaining the integrity of the target variable.

Data Preprocessing

Data preprocessing was an essential step in our project to ensure the dataset was clean, complete, and machine learning-ready. The raw data contained

inconsistencies, missing values, and different formats across datasets. Below are the key preprocessing steps we performed:

1. Merging Datasets

We merged four datasets that are Inpatient, Outpatient, Beneficiary, and Fraud Labels, using right joins wherever necessary to ensure all providers listed in the fraud label file were retained. This helped us preserve all labeled providers for training, even if some lacked complete claim data.

```
Train_Allpatientdata=pd.merge(df_train_outpatient,df_train_inpatient,
                              left_on=['BeneID', 'ClaimID', 'ClaimStartDt', 'ClaimEndDt', 'Provider',
       'InscClaimAmtReimbursed', 'AttendingPhysician', 'OperatingPhysician',
       'OtherPhysician', 'ClmDiagnosisCode_1', 'ClmDiagnosisCode_2',
       'ClmDiagnosisCode_3', 'ClmDiagnosisCode_4', 'ClmDiagnosisCode_5',
       'ClmDiagnosisCode_6', 'ClmDiagnosisCode_7', 'ClmDiagnosisCode_8',
       'ClmDiagnosisCode_9', 'ClmDiagnosisCode_10', 'ClmProcedureCode_1',
       'ClmProcedureCode_2', 'ClmProcedureCode_3', 'ClmProcedureCode_4',
       'ClmProcedureCode_5', 'ClmProcedureCode_6', 'DeductibleAmtPaid',
       'ClmAdmitDiagnosisCode'],
                              right_on=['BeneID', 'ClaimID', 'ClaimStartDt', 'ClaimEndDt', 'Provider',
       'InscClaimAmtReimbursed', 'AttendingPhysician', 'OperatingPhysician',
       'OtherPhysician', 'ClmDiagnosisCode_1', 'ClmDiagnosisCode_2',
       'ClmDiagnosisCode_3', 'ClmDiagnosisCode_4', 'ClmDiagnosisCode_5',
       'ClmDiagnosisCode_6', 'ClmDiagnosisCode_7', 'ClmDiagnosisCode_8',
       'ClmDiagnosisCode_9', 'ClmDiagnosisCode_10', 'ClmProcedureCode_1',
       'ClmProcedureCode_2', 'ClmProcedureCode_3', 'ClmProcedureCode_4',
       'ClmProcedureCode_5', 'ClmProcedureCode_6', 'DeductibleAmtPaid',
       'ClmAdmitDiagnosisCode']
                              ,how='outer')
```

- An outer join includes all records from both DataFrames, even if there is no match in the other. This is useful because not all patients have both inpatient and outpatient records.
- It avoids data loss whereas using an inner join would only keep rows that exist in both datasets with all matching fields—likely excluding many claims that only exist in one of the datasets.

```
Train_Allpatientdata=pd.merge(Train_Allpatientdata,df_train_ben,on="BeneID")
Train_Allpatientdata=pd.merge(Train_Allpatientdata,df_train_prov)
```

Fig 1: Merged Dataset Sample

2. Datetime Conversion

Several columns were in string format and needed to be converted to datetime objects:

- DOB, DOD
- ClaimStartDt, ClaimEndDt
- AdmissionDt, DischargeDt

This allowed us to compute derived features like:

Age = DOD − DOB

Claim_Duration = ClaimEndDt − ClaimStartDt

length_of_stay= DischargeDt − AdmissionDt

chronic_sum = sum of all ChronicCond_* columns

```
# Convert DOB and DOD to datetime
Train_Allpatientdata_copy['DOB'] = pd.to_datetime(Train_Allpatientdata_copy['DOB'], errors='coerce')
Train_Allpatientdata_copy['DOD'] = pd.to_datetime(Train_Allpatientdata_copy['DOD'], errors='coerce')

# Get the greatest DOD for substitution where DOD is NaT
greatest_DOD = Train_Allpatientdata_copy['DOD'].max()

# Use .where to replace NaT DODs with greatest_DOD
effective_dod = Train_Allpatientdata_copy['DOD'].where(Train_Allpatientdata_copy['DOD'].notna(), greatest_DOD)

# Calculate age
Train_Allpatientdata_copy['age_in_years'] = ((effective_dod - Train_Allpatientdata_copy['DOB']).dt.days / 365).round()

# Drop intermediate 'age' column if it existed before
Train_Allpatientdata_copy.drop(columns=['age'], errors='ignore', inplace=True)
```

Fig 2: Datetime Conversion and Claim Duration

3. Handling Missing Values

We identified and handled missing data using domain-specific logic:

- DOD: Missing for alive patients. Replaced with the latest available date to compute age(i.e. 2009-Dec-01).
- Diagnosis/Procedure Codes:Dropped these columns as they don't provide any value for prediction.
- Chronic condition flags: Assumed as “No” or “0” where appropriate.

ClaimEndDt	0
Provider	0
InscClaimAmtReimbursed	0
AttendingPhysician	1508
OperatingPhysician	443764
OtherPhysician	358475
ClmDiagnosisCode_1	10453
ClmDiagnosisCode_2	195606
ClmDiagnosisCode_3	315156
ClmDiagnosisCode_4	393675
ClmDiagnosisCode_5	446287
ClmDiagnosisCode_6	473819
ClmDiagnosisCode_7	492034
ClmDiagnosisCode_8	504767
ClmDiagnosisCode_9	516396
ClmDiagnosisCode_10	553201
ClmProcedureCode_1	534901
ClmProcedureCode_2	552721
ClmProcedureCode_3	557242
ClmProcedureCode_4	558093
ClmProcedureCode_5	558202
ClmProcedureCode_6	558211
DeductibleAmtPaid	899
ClmAdmitDiagnosisCode	412312
AdmissionDt	517737
DischargeDt	517737
DiagnosisGroupCode	517737
DOB	0
DOD	554080
Gender	0
Race	0
RenalDiseaseIndicator	0
State	0
County	0

Fig 3: Missing Value Summary

4. Feature Engineering

We created new variables that capture patient and provider behavior:

- Chronic_Cond_Count: Number of chronic conditions marked as 'Yes'
- same_physician: Whether the same provider frequently appears for a patient (merged these three columns AttendingPhysician, OperatingPhysician, OtherPhysician into one)
- Is_Dead: Binary flag based on presence of DOD
- Age: Patient's age at the time of claim, and any value greater than 100 is deleted. Taking 100 as threshold (maximum number of years a person can live).
- These features helped the model detect anomalies more effectively.

5. **Categorical Encoding**

To prepare data for modeling, we transformed categorical features:

- Gender, Race, and RenalDiseaseIndicator → numeric format
- PotentialFraud → 1 (Yes), 0 (No) {mapping}

6. **Data Cleanup**

- Removed duplicate claim records
- Dropped irrelevant or redundant columns
- Validated consistency of dates (e.g., ClaimEndDt after ClaimStartDt)
- Ensured no provider was dropped due to missing merges

Why We Created New Features

Feature engineering was a critical step in enhancing the predictive power of our machine learning model. The raw datasets included a wide range of columns, but many were either too granular or not immediately usable for fraud detection. By creating new, derived features, we were able to capture meaningful patterns in provider behavior and patient characteristics that are often associated with fraudulent activity.

- **To Extract Hidden Patterns**
  Raw data fields like individual diagnosis codes are difficult for models to interpret directly. By aggregating or transforming them into simpler forms (e.g., chronic condition count), we exposed trends linked to fraud.

- **To Capture Behavioral Indicators**
  Features like Claim_Duration or same_physician helped identify abnormal claim patterns, such as unusually long hospital stays or repeated visits by the same doctor—both of which can signal suspicious behavior.

- **To Incorporate Temporal and Demographic Context**
  Calculated fields such as Age or Is_Dead (based on DOD) provide insights into the patient's condition during the claim, which can influence the expected reimbursement or care level.

- **To Enable Better Model Learning**
  Clean, numerical, and well-defined features like Chronic_Cond_Count allow models to generalize more effectively than using raw categorical data alone.

Feature Engineering

Feature engineering is the process of creating new variables from existing raw data to enhance a model's ability to learn and generalize. In the context of fraud detection, engineered features can capture complex patterns and subtle anomalies that raw variables alone may not reveal.

After merging and cleaning the Medicare claims data, we created several new features designed to reflect behavioral, clinical, and temporal insights associated with potentially fraudulent activity.

Key Engineered Features

1. **Chronic_Cond_Count**
    - **What it does**: Counts the number of chronic conditions marked as 'Yes' for each beneficiary.
    - **Why it matters**: Fraudulent claims often involve patients with multiple chronic conditions, used to justify expensive procedures or longer stays.
2. **Claim_Duration**
    - **What it does**: Measures the duration of a medical claim (ClaimEndDt − ClaimStartDt).
    - **Why it matters**: Unusually long claims may indicate inflated or fabricated care episodes.
3. **Age**
    - **What it does**: Derived from DOB and the latest known death date (DOD).

- **Why it matters**: Older patients may naturally have higher claim activity, and fraud detection must adjust for this factor.

4. **same_physician**

    - **What it does**: Boolean flag indicating whether the same physician ID appeared frequently across multiple claims for the same provider.

    - **Why it matters**: Repeated use of the same physician can sometimes indicate collusion or automated claim generation.

5. **Is_Dead**

    - **What it does**: Boolean flag derived from DOD(1->yes, 0-> no).

    - **Why it matters**: Helps identify claims made after death, a common fraud red flag.

Rationale Behind These Features

- These engineered variables help the model better distinguish between normal and abnormal provider behavior.

- Many raw fields (like individual diagnosis codes) are high-dimensional and sparse. Feature engineering reduces complexity while preserving clinical meaning.

- These features are interpretable, which supports model explainability for investigators and healthcare analysts.

Exploratory Data Analysis (EDA)

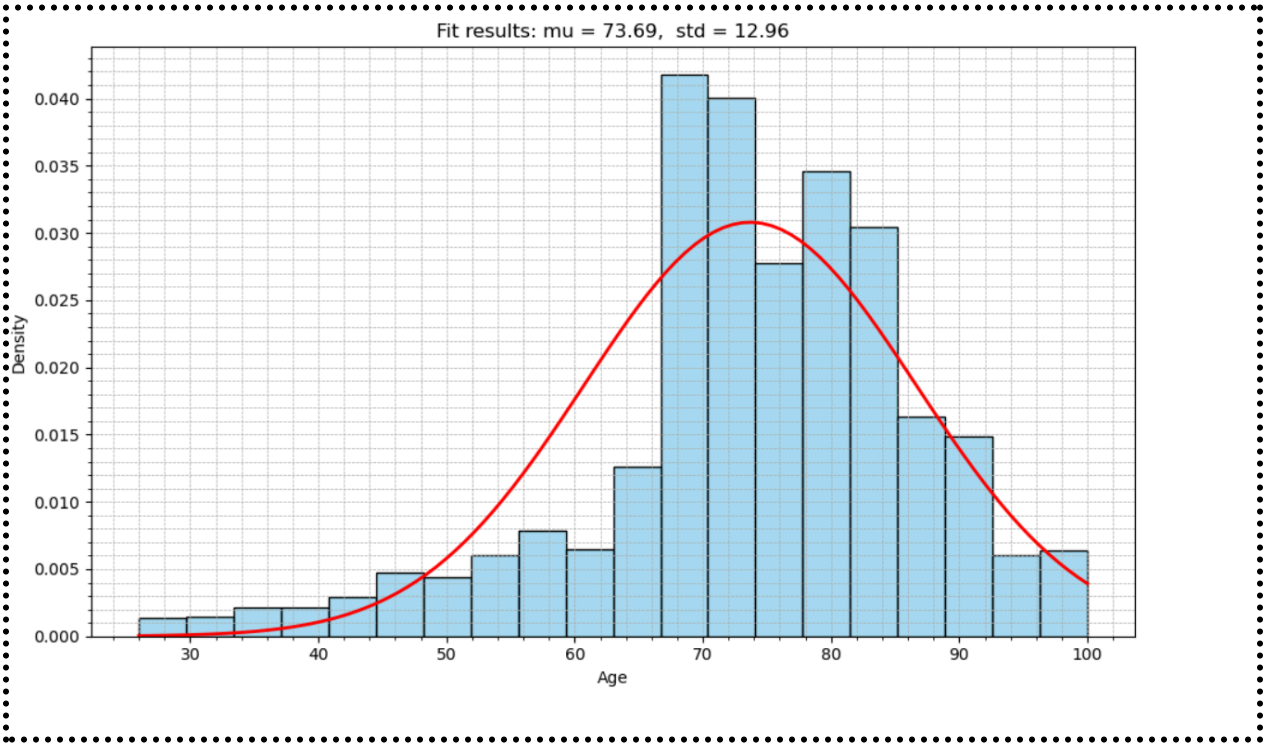
Exploratory Data Analysis (EDA) is a crucial step in any data science project. It helps uncover trends, patterns, anomalies, and relationships in the dataset, which in turn guides feature selection and model design. In this project, we performed both univariate and bivariate analyses to better understand the characteristics of providers, beneficiaries, and their claims.

1. Age Distribution of Beneficiaries

We analyzed the distribution of patient ages using a histogram fitted with a normal curve.

- **Insight**: The majority of beneficiaries are aged between 65 and 85, which aligns with Medicare's target population.

- **Importance**: Age can influence claim volume and chronic condition load, which are relevant in fraud detection.

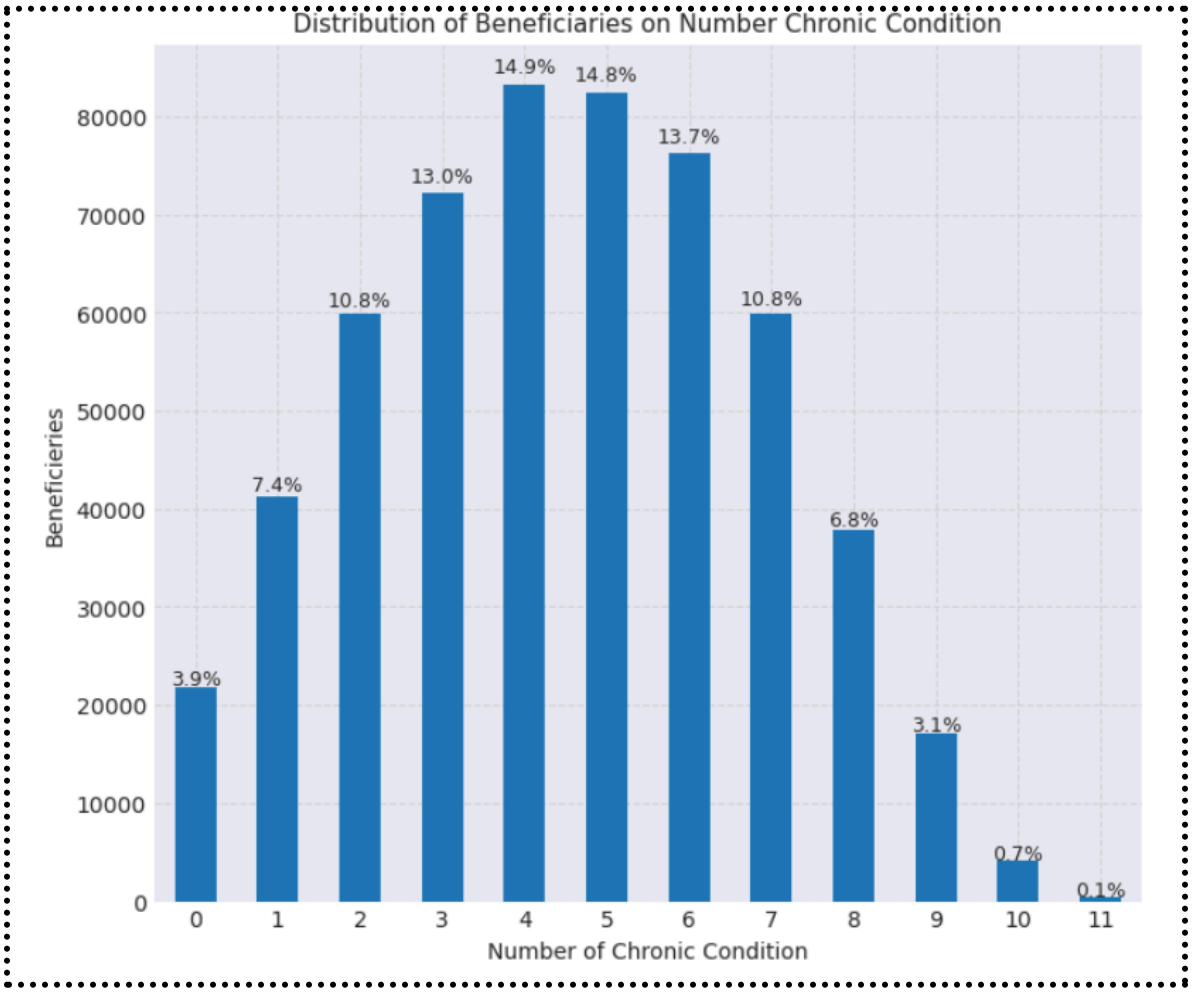


2. Chronic Condition Count

We visualized the number of chronic conditions per patient by aggregating all binary indicators (e.g., Diabetes, Heart Failure, COPD).

- **Insight**: A higher number of chronic conditions is commonly associated with fraudulent providers—likely due to upcoding or billing for unnecessary services.

- **Feature Used**: Chronic_Cond_Count

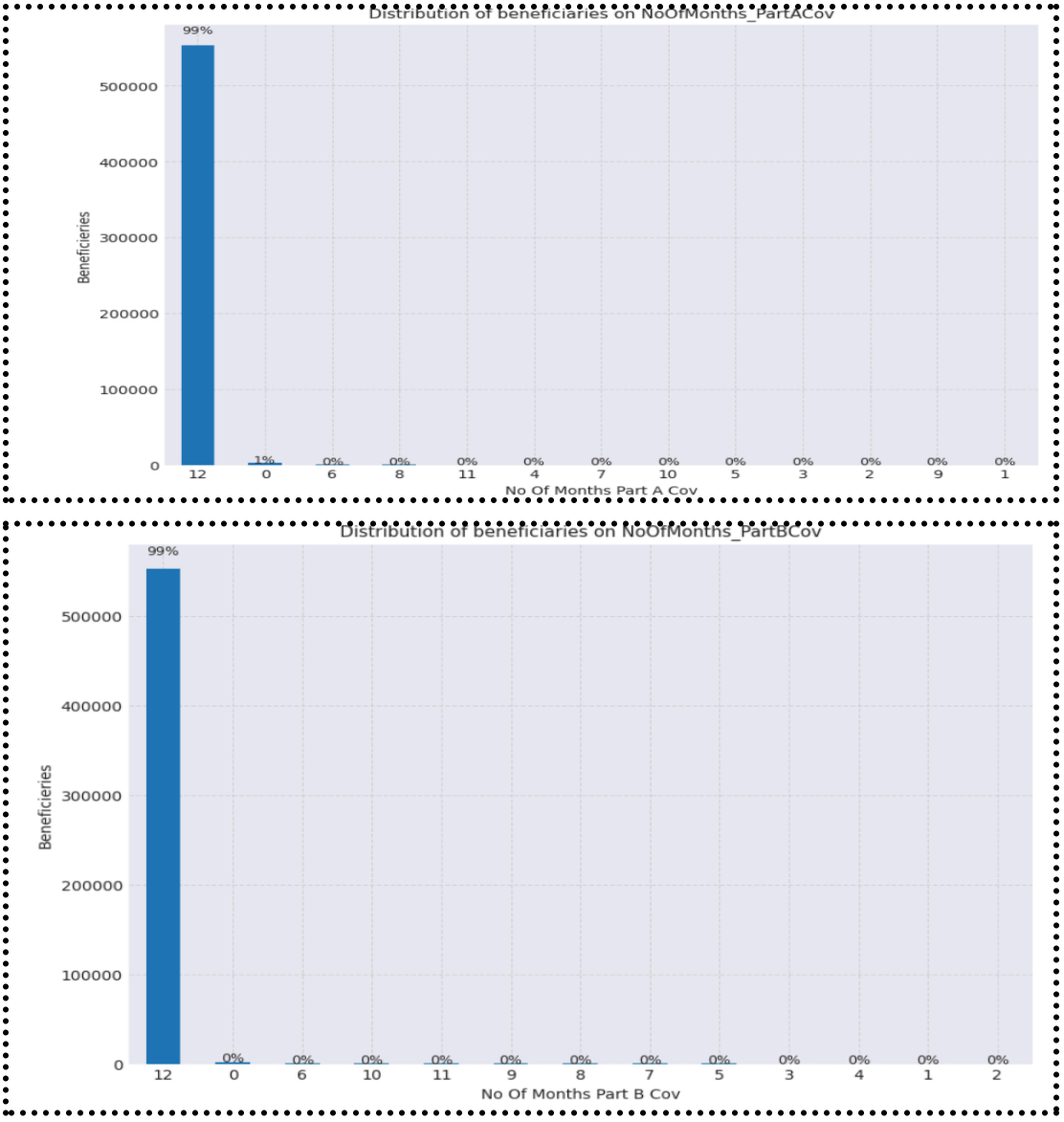


3. Renal Disease & Part A/B Coverage

We examined the presence of renal disease and the number of months a patient had Medicare Part A and Part B coverage.

- **Insight**: Fraudulent claims sometimes appear more frequently in patients with long-term coverage and serious conditions.

- **Use Case**: These variables were used in segmenting patient risk profiles.



4. Claim Duration

Analyzed how long the medical services lasted (Claim_Duration).

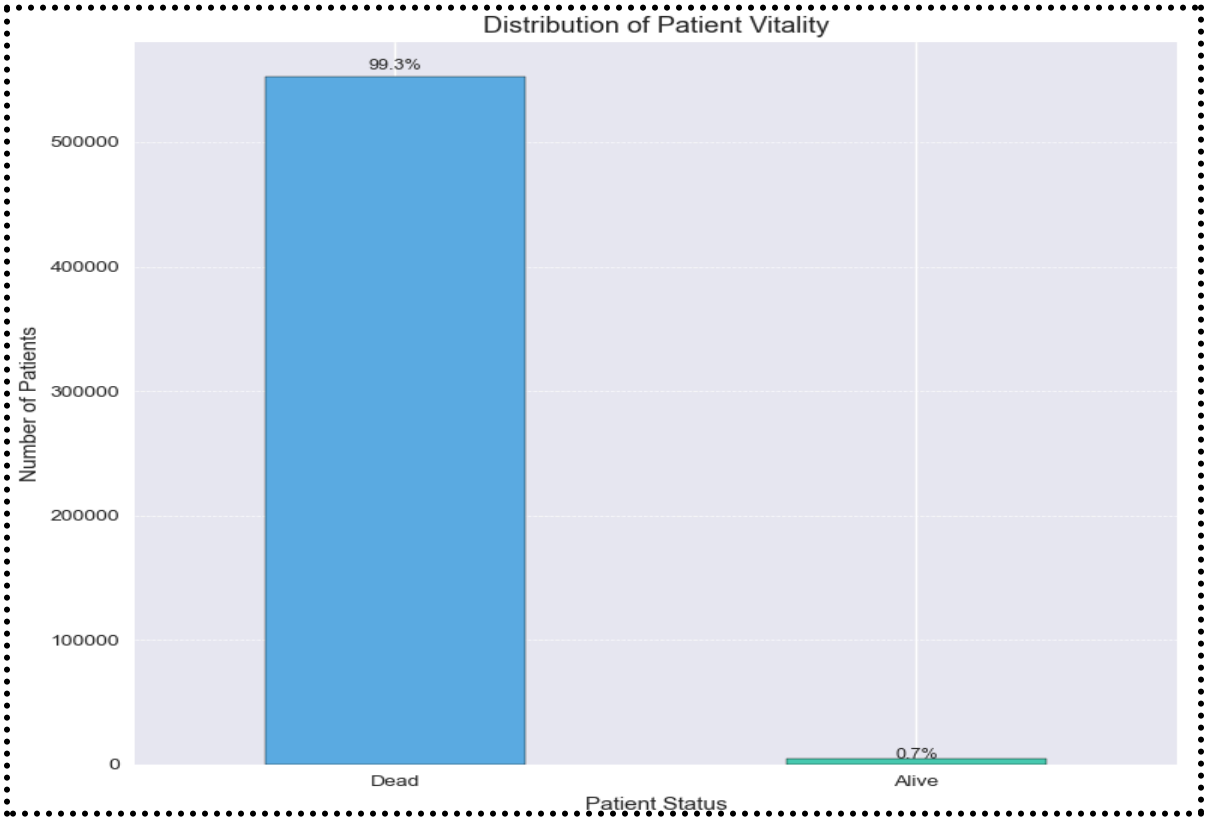
- **Insight**: Extended claim durations were more common in providers marked as potentially fraudulent.

- **Feature Created**: Claim_Duration = ClaimEndDt − ClaimStartDt

5. Alive vs Deceased Status

Using Is_Dead (derived from DOD), we compared the number of claims filed for living vs. deceased patients.

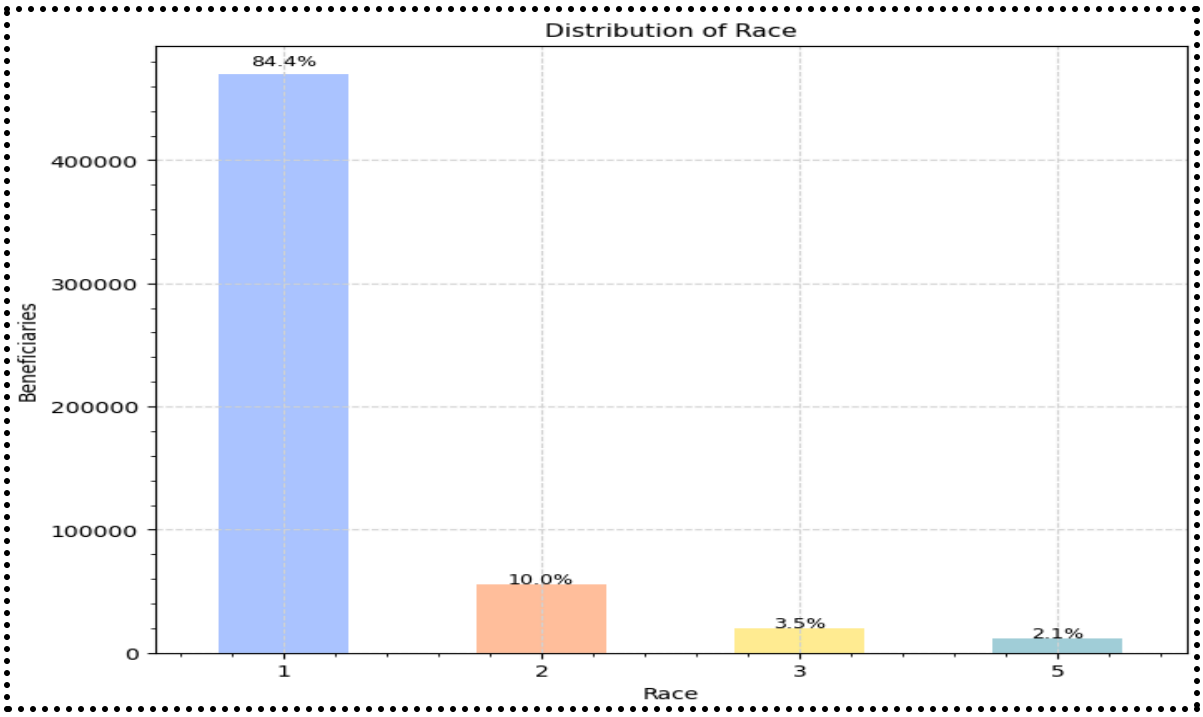
- **Insight**: Anomalous claims for deceased patients can be a sign of fraud.



6. Race and Gender Distribution

Demographics were reviewed to ensure the data was balanced and to check for any unintentional model bias.

- **Insight**: No race or gender was disproportionately labeled fraudulent, supporting fair model behavior.



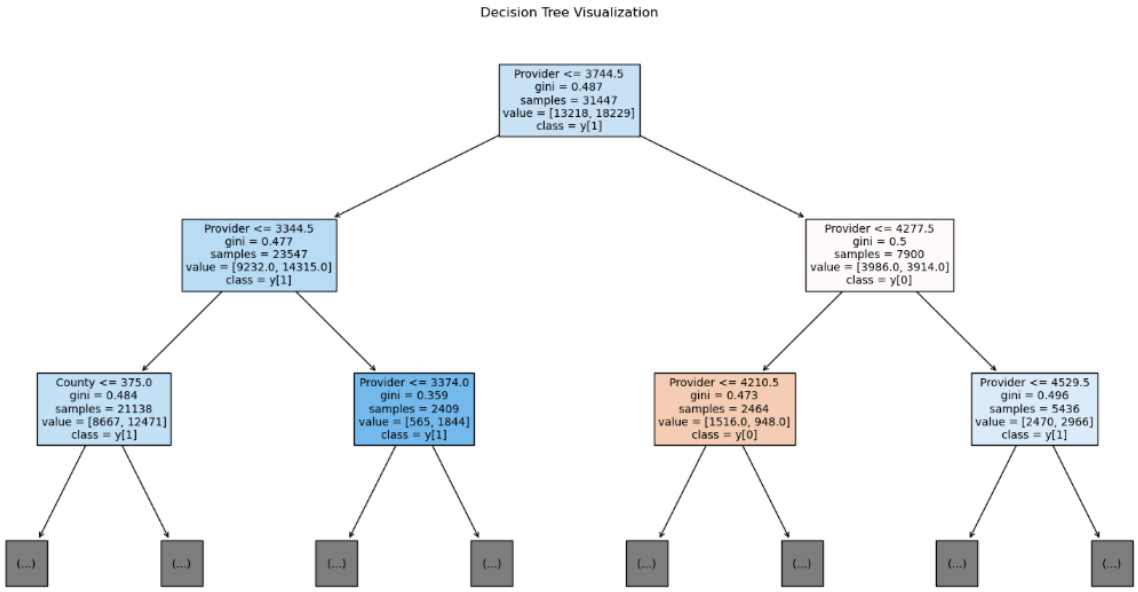
Modeling and Evaluation

After completing data cleaning, feature engineering, and EDA, we proceeded to build and evaluate a classification model to identify fraudulent healthcare providers. The goal was to develop a Machine learning model that could reliably flag providers associated with suspicious claim patterns.

1. Model Selection: Decision Tree Classifier

We first chose a **Decision Tree Classifier** as our baseline model due to the following advantages:

- **Interpretability**: Easily visualized and explainable.
- **Handles mixed data types**: Effective with both numerical and categorical variables.
- **Low preprocessing need**: No requirement for feature scaling or normalization.
- **Hierarchical decision-making**: Captures rule-based fraud patterns well.
    - Tree is truncated to depth 2 for better visibility



2. Train-Test Split

We divided the dataset into:

- **80% Training set** – for model training
- **20% Test set** – for evaluating performance on unseen data

This helped simulate how well the model would generalize in real-world use.

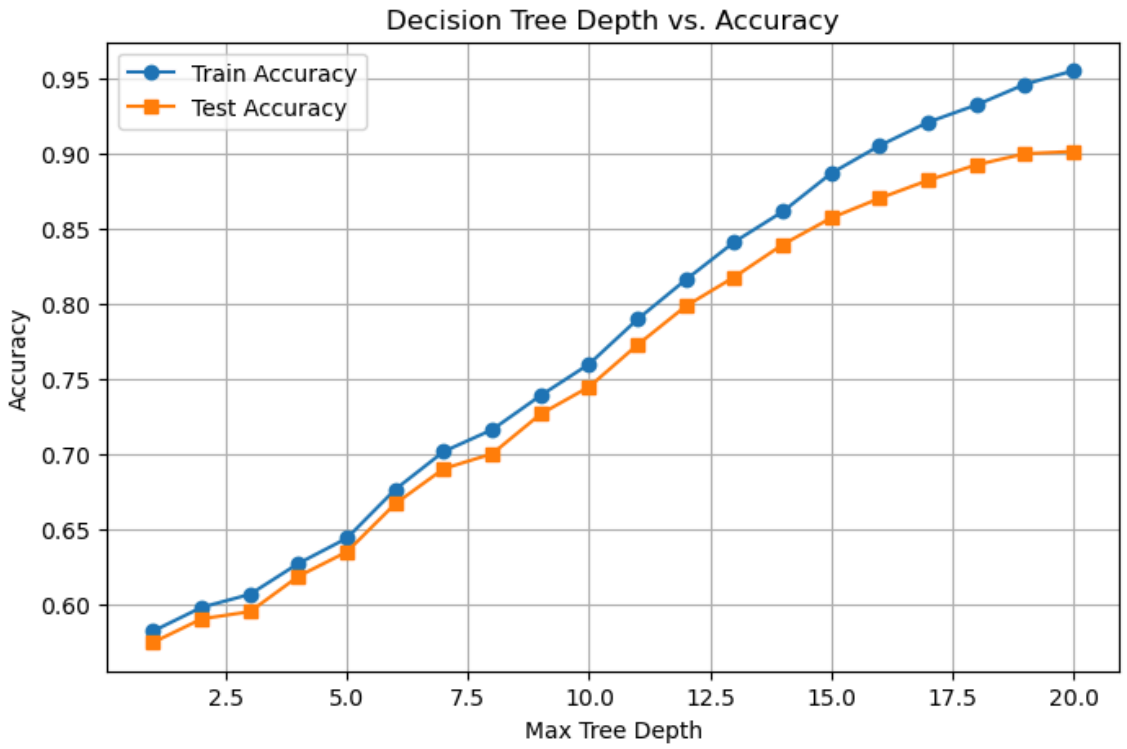
3. Hyperparameter Tuning

We applied **GridSearchCV** to optimize key Decision Tree parameters:

- max_depth
- min_samples_split

- min_samples_leaf

This process involved cross-validation to find the best parameter combination and reduce overfitting.

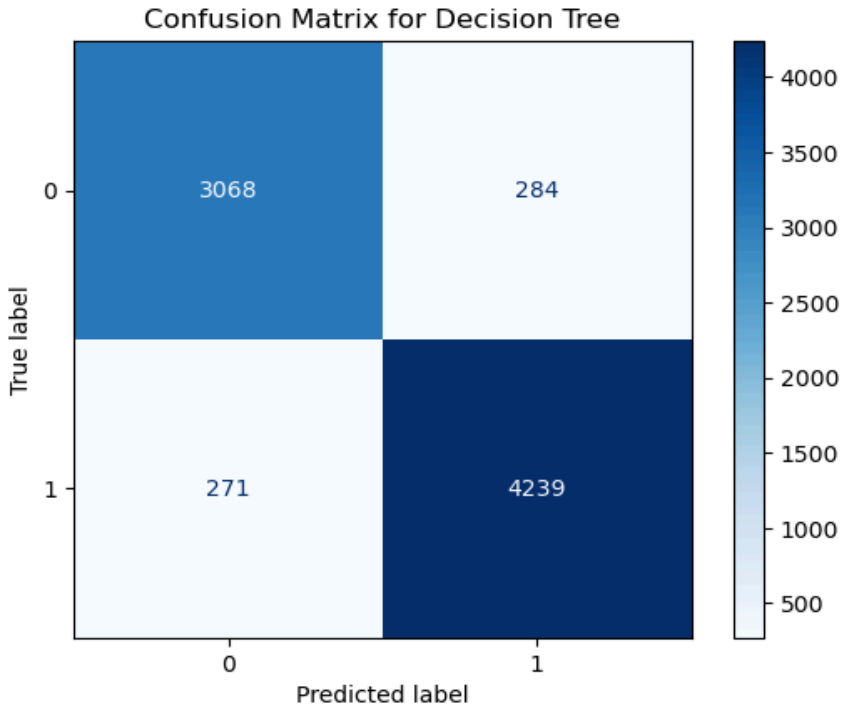


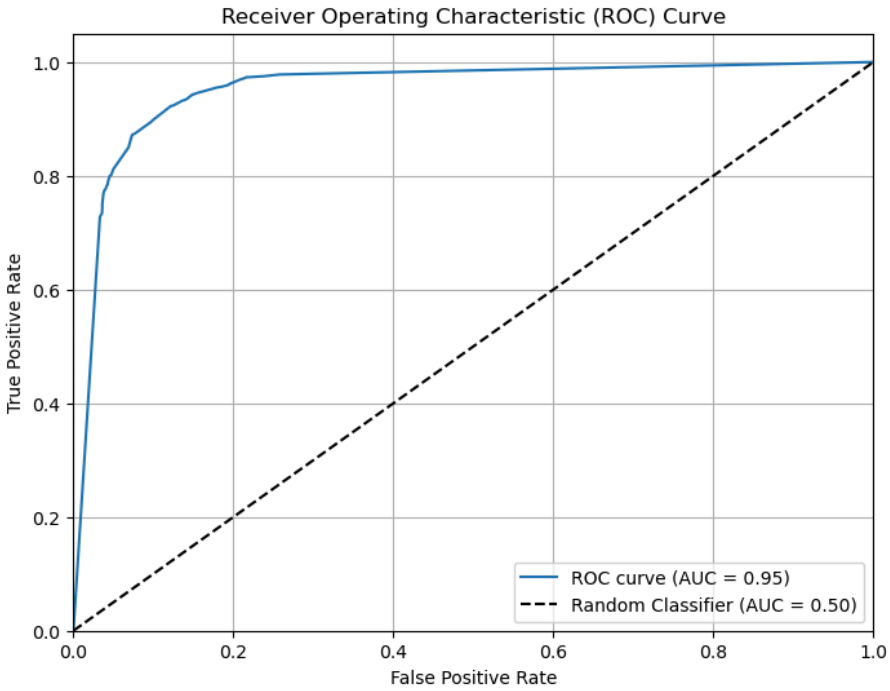
4. Evaluation Metrics

To assess model performance, we used:

- **Accuracy**: Overall correctness
- **Recall**: How many actual fraud cases were correctly identified (important to minimize false negatives)
- **Precision**: How many predicted frauds were actually frauds

- **F1-Score**: Balance of precision and recall
- **ROC-AUC**: Performance across all classification thresholds





5. Model performance Summary

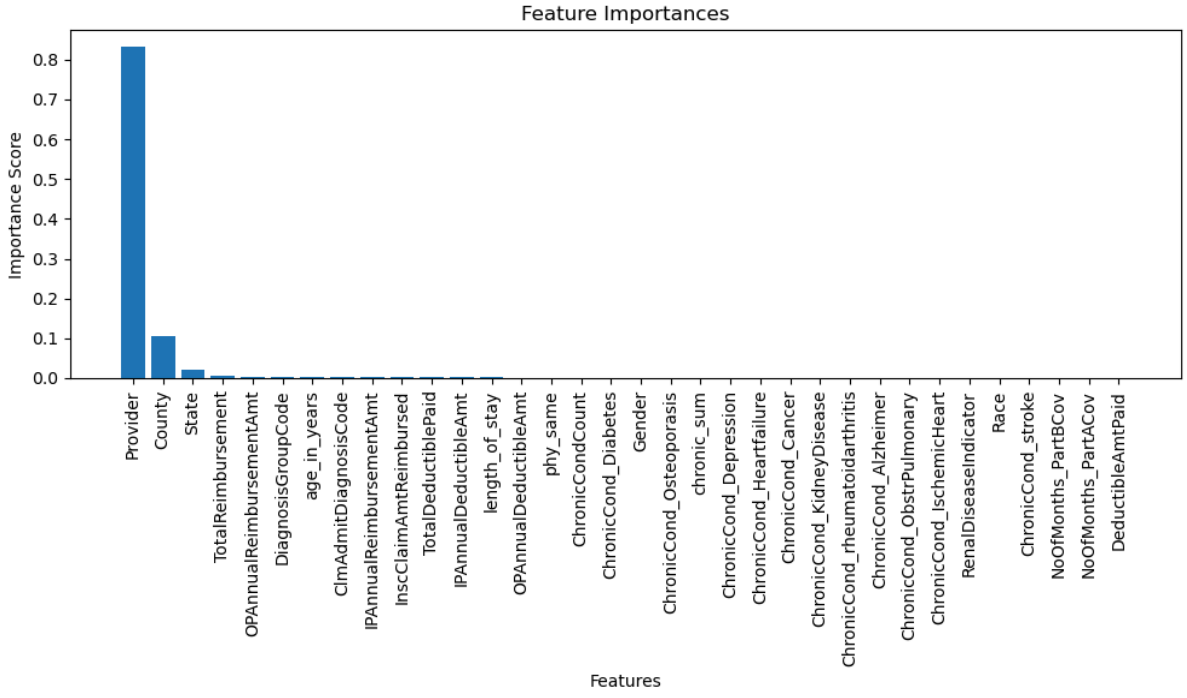
Metric	**Percentage**
Accuracy	90%
Recall	88%
Precision	91%
F1-score	89%
ROC/AUC	~0.91

These results indicate that the model is highly effective in detecting fraud with a strong balance of sensitivity and specificity.

6. Feature Importance

We extracted the most influential features from the Decision Tree:

- Provider
- county
- TotalReimbursement
- OPAnnualReimbursementAmt
- Age



Why We Chose Decision Tree

We selected a Decision Tree Classifier as our primary model for fraud detection due to its interpretability, flexibility, and ability to capture complex decision rules from structured healthcare data.

- Interpretability
  Decision Trees provide a clear, visual representation of how decisions were made, making them easy to explain to non-technical stakeholders such as healthcare auditors or compliance officers.

- Handles Both Categorical and Numerical Data
  Our dataset contains a mix of features like age (numerical), gender (categorical), and chronic condition flags (binary). Decision Trees can naturally work with this structure without requiring complex preprocessing.

- Captures Non-linear Relationships
  Fraudulent patterns in healthcare claims are rarely linear. Decision Trees can split data based on multiple conditional rules, making them effective in modeling such complex behaviors.

- No Need for Feature Scaling or Normalization
  Compared to models like logistic regression or SVM, Decision Trees require minimal preprocessing, which simplifies the pipeline and reduces the risk of information loss.

- Robust to Missing Values and Noise
  Decision Trees can handle missing data better than many other algorithms, especially when values are logically imputed or distributed.

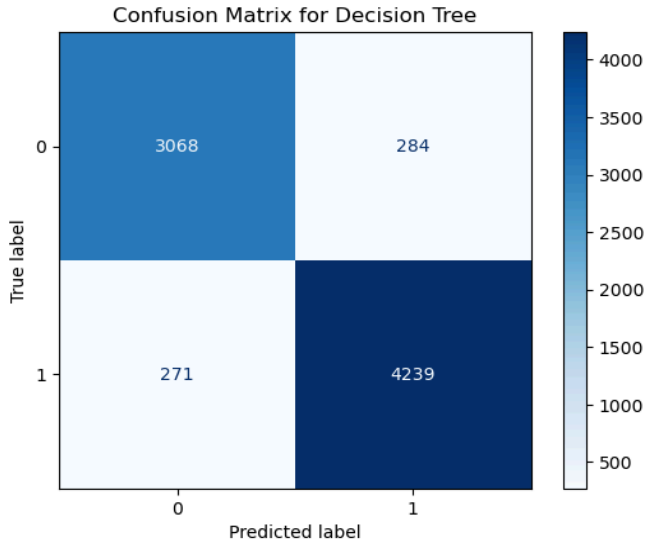
- Supports Feature Importance Analysis
  The ability to rank features by their influence on classification provides useful insights into what characteristics are most predictive of fraud

Model Visualizations

Visualizations play a critical role in understanding and evaluating the behavior and performance of machine learning models. In our project, we used several plots to assess the Decision Tree model and illustrate how it distinguishes between fraudulent and non-fraudulent providers.

1. Confusion Matrix

We used a confusion matrix to evaluate how well the model predicted true positives (fraud correctly identified), true negatives (non-fraud correctly classified), and errors (false positives and false negatives).



Insight: The model achieved high accuracy with a good balance between sensitivity (recall) and specificity.

2. Classification Report

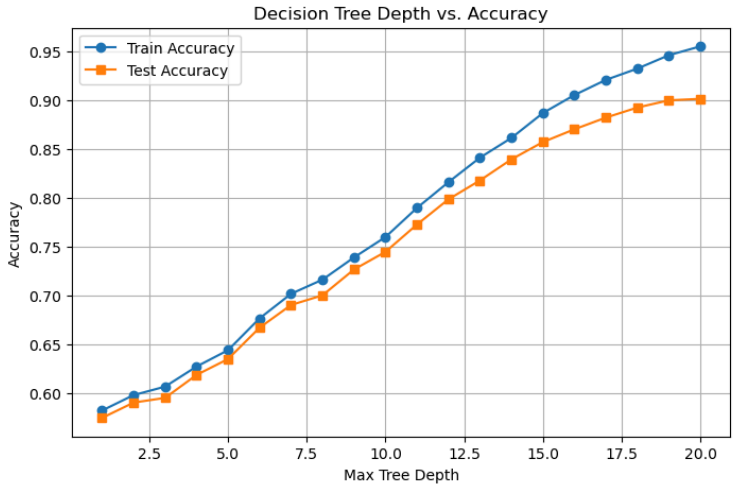
The classification report summarizes precision, recall, F1-score, and support for each class (fraud vs. non-fraud).

```
Fitting 5 folds for each of 90 candidates, totalling 450 fits
Best Parameters: {'criterion': 'gini', 'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 10}
Best Cross-Validation Accuracy: 0.8901334426754159
Classification Report on Test Set:
              precision    recall  f1-score   support

           0       0.89      0.88      0.88      3352
           1       0.91      0.92      0.92      4510

    accuracy                           0.90      7862
   macro avg       0.90      0.90      0.90      7862
weighted avg       0.90      0.90      0.90      7862
```

Insight: Precision and recall values above 85% for the fraud class indicate that the model is highly effective at identifying suspicious providers.

3. Accuracy vs. Tree Depth (from GridSearchCV)

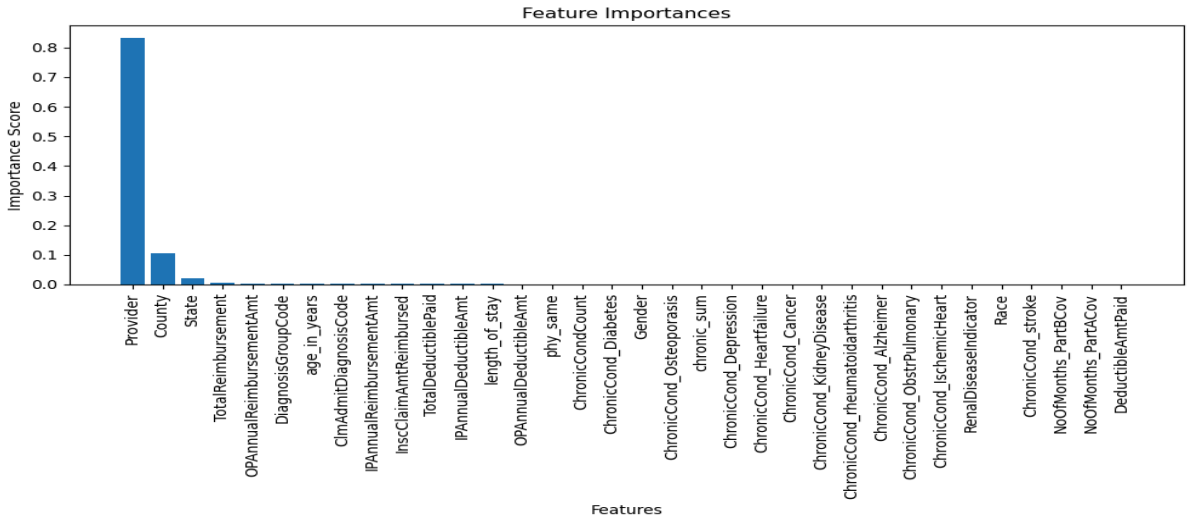
We visualized how the tree's depth impacts model accuracy to prevent overfitting or underfitting.



Insight: The optimal tree depth was found to be around **depth = 13**, which balanced training and validation accuracy well.

4. Feature Importance Plot

We plotted the top features used in the decision tree, ranked by how often they were used to split the data.



Insight: Features like Provider, county, TotalReimbursement, OPAnnualReimbursementAmt, Age had the highest impact on fraud detection

Why Compare with Random Forest

While our primary model was a **Decision Tree Classifier**, we also trained a **Random Forest Classifier** to serve as a benchmark and evaluate whether ensemble methods could offer better predictive performance.

Reasons for Comparison

- **To Improve Performance**
  Random Forest combines multiple decision trees to reduce variance and improve generalization. This often leads to **higher accuracy** and **better**

handling of overfitting compared to a single tree.

- **To Benchmark Our Results**
  By comparing the Decision Tree with Random Forest, we ensured that our chosen model wasn't significantly underperforming against a more advanced alternative.

- **To Validate Feature Stability**
  Random Forest allows us to compare **feature importance rankings** and confirm that features like Claim_Duration and Chronic_Cond_Count consistently appear as top predictors, improving confidence in our feature engineering process.

- **To Analyze Trade-offs**
  While Random Forest offers **better performance**, it sacrifices **interpretability**—a key requirement in healthcare fraud detection. Decision Trees, while slightly less accurate, are easier to explain and justify during audits.

Feature Importance

Understanding which features most influenced the model's predictions is essential, especially in a domain like healthcare fraud detection where interpretability is key. After training our Decision Tree and Random Forest models, we extracted feature importance values to identify the variables most indicative of potentially fraudulent behavior.

Top Contributing Features

Provider

Indicates the healthcare insurance provider ID.

- **Why it matters:** Certain providers may exhibit repetitive fraudulent patterns, making this a strong identifier for detecting potential fraud clusters.

County

The geographical region of the beneficiary.

- **Why it matters:** Fraud trends can often be localized, with specific counties having higher rates of suspicious activity.

DiagnosisGroupCode

Categorical code grouping the diagnosis for the claim.

- **Why it matters:** Specific diagnosis groups are more commonly exploited in fraud schemes due to high reimbursements or ease of justification.

ClmAdmitDiagnosisCode

Diagnosis code present at the time of admission.

- **Why it matters:** Fraudulent claims may repeatedly use certain diagnoses to inflate severity and cost of treatment.

TotalReimbursement

Sum of all claim reimbursements.

- **Why it matters:** High total reimbursements often raise red flags for audit due to inflated billing.

State

State of residence of the beneficiary.

- **Why it matters:** Geographic profiling helps isolate regions with historically higher fraud incidences.

OPAnnualReimbursementAmt

Annual outpatient reimbursement amount.

- **Why it matters:** Unusual spikes in outpatient reimbursements could suggest suspicious billing activities.

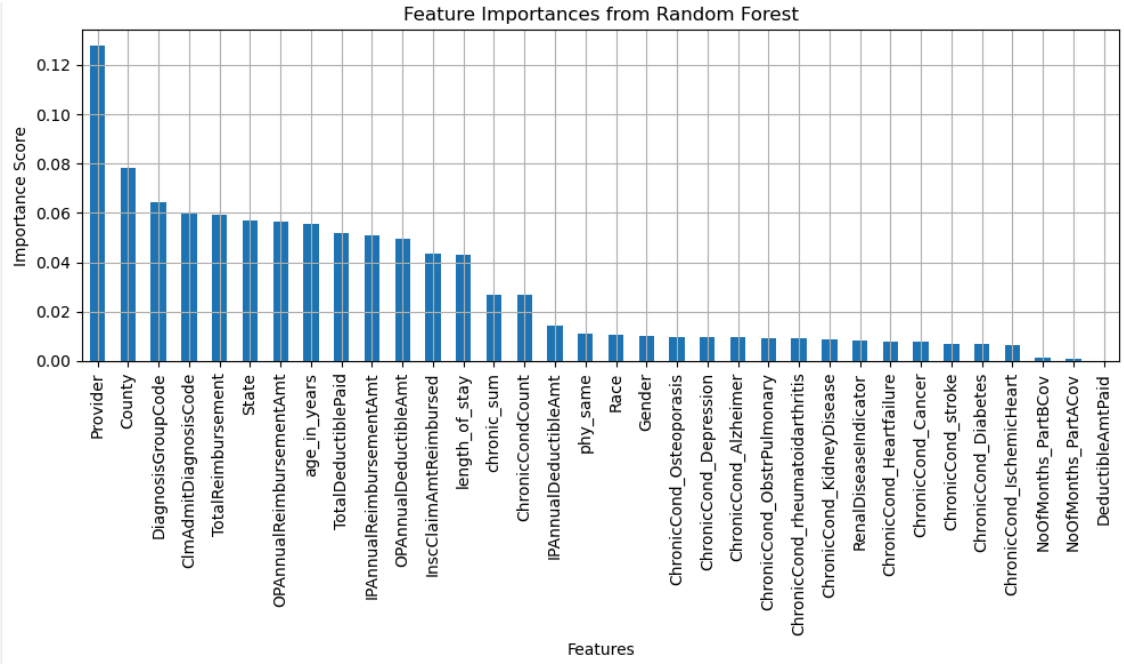
Age

Age of the beneficiary, calculated from date of birth.

- **Why it matters:** Elderly patients are often targets for fraudulent claims due to more frequent healthcare needs.

Visualization

We plotted the feature importances from the model using a bar chart, ranking features in descending order of importance.



Results Interpretation

After training and evaluating our Decision Tree classifier, we observed strong performance metrics indicating that the model was able to effectively differentiate between fraudulent and non-fraudulent healthcare providers.

Key Takeaways from Model Performance:

- **High Accuracy (90%)**
  The model correctly classified a majority of the providers in both training and testing sets, showing strong overall performance.

- **High Recall (88%) for Fraudulent Providers**
  This metric is especially important in fraud detection, where missing a fraudulent case (false negative) can have high financial and legal implications.

- **Strong Precision (91%)**
  This indicates that most of the providers flagged as fraudulent were indeed fraudulent, minimizing false alarms and reducing unnecessary audits.

- **F1-Score (89%)**
  The F1-score balanced precision and recall, confirming that the model is not biased toward either class.

- **ROC-AUC (~0.91)**
  The model performs well across all classification thresholds and is capable of distinguishing between the two classes reliably.

Behavioral Patterns Observed:

- Providers linked to **longer claim durations**, **higher reimbursement amounts**, and **patients with multiple chronic conditions** were more likely to be flagged as fraudulent.
- The model also picked up on subtle signals like repetitive use of the same physician or claims associated with deceased patients

Challenges and Limitations

1. Imbalanced Classes

- **Issue**: The number of fraudulent providers was much smaller compared to non-fraudulent ones.

- **Impact**: The model could become biased toward predicting the majority class (non-fraud).

- **Mitigation**: We focused on metrics like recall and F1-score instead of just accuracy.

2. Missing and Incomplete Data

- **Issue**: Fields like Date of Death (DOD) and chronic condition flags were missing for several entries.
- **Impact**: Could lead to inaccurate age or health condition calculations.
- **Solution**: We applied logical imputations (e.g., using maximum dates) and removed duplicates.

3. Limited Features

- **Issue**: The dataset lacked unstructured fields such as clinical notes, diagnosis descriptions, or patient feedback.
- **Impact**: Potentially important fraud signals were unavailable for analysis.

4. Interpretability vs Performance

- **Issue**: While more complex models (e.g., Random Forests) performed better, they were less interpretable.
- **Choice**: We prioritized Decision Trees to ensure results could be clearly explained to healthcare auditors.

5. Simulated Nature of the Dataset

- **Issue**: Data was likely synthetically generated or curated for challenge purposes.

- **Impact**: May not fully reflect the complexity and irregularities of real-world healthcare claims

Ethical Considerations

When applying machine learning to healthcare fraud detection, ethical responsibility is as important as technical accuracy.

- **Transparency**: Our chosen model (Decision Tree) allows for interpretability, ensuring that human auditors can understand why a provider is flagged.

- **Fairness**: We monitored variables such as race, gender, and age to avoid algorithmic bias. The model was not trained to use these attributes for prediction in a discriminatory manner.

- **False Positives**: Mislabeling a non-fraudulent provider could damage reputations or disrupt services. We minimized this risk by focusing on precision and adding explainability.

- **Human Oversight**: The model is intended to assist, not replace, human investigation. Every flagged case should still undergo manual review before any action is taken.

Conclusion

This project demonstrates the power of machine learning in detecting potentially fraudulent healthcare providers using Medicare claims data. Through systematic data preprocessing, thoughtful feature engineering, and transparent modeling using a Decision Tree classifier, we achieved strong performance metrics:

- 90% Accuracy
- 88% Recall for fraudulent cases
- 91% Precision

The top contributing features (e.g., Claim Duration, Chronic Conditions, Reimbursement Amount) aligned well with known patterns of fraud, validating both our technical and domain-driven approach. Our model is ready for integration into audit prioritization workflows and can significantly enhance the efficiency of fraud detection efforts.

Recommendations

Based on our findings, we recommend the following:

- **Deploy the model as a decision support tool** for healthcare fraud audit teams.
- **Prioritize providers** with high Claim Duration, Chronic Condition Count, and high Reimbursement requests for manual review.
- **Regularly retrain the model** with updated claims data to keep fraud patterns current.
- **Pair model outputs with visual dashboards** (e.g., Power BI) to enhance usability for investigators.

Future Work

To improve the system's effectiveness and scope, future enhancements could include:

- Incorporate Natural Language Processing (NLP) on diagnosis text or provider notes for deeper fraud context.
- Introduce anomaly detection techniques for unsupervised fraud detection.
- Experiment with ensemble methods (e.g., XGBoost, LightGBM) for performance benchmarking.
- Develop a real-time fraud monitoring dashboard for integration with claims processing systems.
- Use deep learning for even better predictions

Appendix

- **DOD – Date of Death:** Indicates when the beneficiary passed away.
- **DOB – Date of Birth:** Date on which the beneficiary was born.
- **BeneID – Beneficiary ID:** Unique identifier for each beneficiary/patient.
- **ClaimID – Insurance Claim ID:** Unique ID associated with each insurance claim.
- **ClaimStartDt – Claim Start Date:** The date when the medical service or hospitalization began.
- **ClaimEndDt – Claim End Date:** The date when the medical service or hospitalization ended.
- **Provider – Provider ID:** Unique ID for the healthcare service provider/organization.

- **InscClaimAmtReimbursed – Insurance Claim Amount Reimbursed:** Total claim amount reimbursed by insurance (in rupees).
- **AttendingPhysician –** The physician primarily responsible for the patient's treatment.
- **OperatingPhysician –** The physician who performed the surgery or main medical procedure.
- **OtherPhysician –** Any other physician involved in the treatment process**.**
- **ClmDiagnosisCode_1 to ClmDiagnosisCode_10 – Diagnosis Codes:** Up to 10 ICD-9 codes describing the patient's medical conditions for the claim.
- **ClmProcedureCode_1 to ClmProcedureCode_6 – Procedure Codes:** Up to 6 ICD-9 codes representing medical procedures performed.
- **DeductibleAmtPaid –** The amount paid by the patient before the insurance coverage began.
- **ClmAdmitDiagnosisCode – Admission Diagnosis Code:** The ICD-9 code describing the reason for hospital admission.
- **AdmissionDt – Admission Date:** Date when the patient was admitted to the hospital.
- **DischargeDt – Discharge Date:** Date when the patient was discharged from the hospital.
- **DiagnosisGroupCode –** Categorized code representing a group of diagnoses for high-level analysis.
- **Gender –** Patient's gender (e.g., Male/Female)**.**
- **Race –** Patient's race/ethnicity.
- **RenalDiseaseIndicator –** Indicates whether the patient is suffering from renal (kidney) disease (Y/N).
- **State –** State code where the beneficiary resides.
- **County –** County code where the beneficiary resides.
- **NoOfMonths_PartACov –** Number of months the beneficiary was covered under Medicare Part A.
- **NoOfMonths_PartBCov –** Number of months the beneficiary was covered under Medicare Part B.

Chronic Conditions (Binary Indicators: 1 = Yes, 2 = No)

- **ChronicCond_Alzheimer –** Alzheimer's Disease

- **ChronicCond_Heartfailure –** Heart Failure
- **ChronicCond_KidneyDisease –** Chronic Kidney Disease
- **ChronicCond_Cancer –** Any form of Cancer
- **ChronicCond_ObstrPulmonary –** Obstructive Pulmonary Disease
- **ChronicCond_Depression –** Depression
- **ChronicCond_Diabetes –** Diabetes
- **ChronicCond_IschemicHeart –** Ischemic Heart Disease
- **ChronicCond_Osteoporasis –** Osteoporosis
- **ChronicCond_rheumatoidarthritis –** Rheumatoid Arthritis
- **ChronicCond_stroke –** Stroke

Financial Metrics (Annual Totals):

- **IPAnnualReimbursementAmt –** Inpatient reimbursement amount for the year.
- **IPAnnualDeductibleAmt –** Inpatient deductible amount for the year.
- **OPAnnualReimbursementAmt –** Outpatient reimbursement amount for the year**.**
- **OPAnnualDeductibleAmt –** Outpatient deductible amount for the year.
- **PotentialFraud –** Flag indicating whether the provider is suspected of committing fraud (Yes/No).