A
Data Science & Big Data Analatics Mini Project Report submitted to
Savitribai Phule Pune University, Pune

Insurance Fraud Detection

In partial Fulfillment for the awards of Degree of Engineering in Computer Engineering

Submitted by

Mr. Chintan Gangurde T400240785
Mr. Tejas Kolhe T400240835
Mr. Nauman Kurkarni T400240838
Mr. Om Gunjikar T400240798

Under the Guidance of

Prof. Rubi Mandal

Department of Computer Engineering

DYP DPU
Dr. D. Y. Patil Unitech Society's
Dr. D. Y. Patil Institute of Technology
Pimpri, Pune - 411018.

2024-25

Certificate

This is to certify that,

Mr. Chintan Gangurde	T400240785
Mr. Tejas Kolhe	T400240835
Mr. Nauman Kurkarni	T400240838
Mr. Om Gunjikar	T400240798

have successfully completed the Mini project entitled " **Insurance Fraud Detection** " under my guidance in partial fulfillment of the requirements for the Third Year of Engineering in Computer Engineering under the Savitribai Phule Pune University during the academic year 2024-2025

Date : …….

Place:…….

Mrs.Rubi Mandal **Dr. Vinod Kimbahune**

Project Guide **HOD**

Dr. Nitin Sherje
Pincipal

Acknowledgements

With deep sense of gratitude we would like to thank all the people who have lit our path with their kind guidance. We are very grateful to these intellectuals who did their best to help during our project work.

It is our proud privilege to express a deep sense of gratitude to **Pro,Dr. Nitin Sherje** Principal of for his comments and kind permission to complete this project. We remain indebted to **Dr. Vinod Kimbahune,** H.O.D.Computer Engineering Department for his timely suggestion and valuable guidance.

The special gratitude goes to Mrs. Rubi Mandal excellent and precious guidance in completion of this work .We thanks to all the colleagues for their appreciable help for our working project. With various industry owners or lab technicians to help, it has been our endeavor throughout our work to cover the entire project work.

We are also thankful to our parents who provided their wishful support for our project completion successfully. And lastly we thank our all friends and the people who are directly or indirectly related to our project work.

Mr. Chintan Gangurde
Mr. Tejas Kolhe
Mr. Nauman Kurkarni
Mr. Om Gunjikar

Abstract

*The **Insurance Fraud Detection** system is a machine learning-based solution designed to assist insurance companies in identifying fraudulent claims. Insurance fraud has become a critical issue, causing substantial financial losses every year. Traditional manual methods of fraud detection are time-consuming, inefficient, and often fail to catch subtle fraud patterns.*

This project uses a dataset of historical insurance claims to analyze and classify whether a claim is genuine or fraudulent. Using Python and data science libraries such as pandas, numpy, seaborn, matplotlib, and scikit-learn, the project follows a pipeline of data preprocessing, exploratory data analysis, model building, and evaluation.

Various supervised machine learning algorithms—including Decision Trees, Random Forest, and Support Vector Machines—are implemented and compared based on their accuracy, precision, recall, and F1 score. The insights generated from visualizations help stakeholders understand common traits of fraudulent activities.

The primary aim of this project is to create a smart, scalable, and automated system that can alert insurers about suspicious claims, thereby improving operational efficiency and reducing losses. This project not only showcases the potential of data science in the insurance sector but also enhances our understanding of real-world applications of machine learning.

Keywords

Insurance Fraud, Machine Learning, Data Analytics, Classification, Python, Scikit-learn, Pandas, Data Preprocessing, Fraud Detection, Random Forest, Support Vector Machine, Data Visualization, Model Evaluation, Automation.

Table of Contents

List of Abbreviations

List of Abbreviations (Insurance Fraud Detection)

In technical project reports like this Movie Review Website, a **List of Abbreviations** is optional but very helpful for readers and evaluators. It provides a quick reference for all the abbreviations used throughout the report, especially for technical terms.

For this project, the list may include standard web development abbreviations such as:

- ML – Machine Learning
- DB – Database
- CSV – Comma Separated Values
- UI – User Interface
- API – Application Programming Interface
- SVM – Support Vector Machine
- DT – Decision Tree
- RF – Random Forest

These abbreviations are widely recognized, but still must be listed for clarity and consistency.

As a best practice:

The dataset was loaded into the program using Pandas, a Python library for data manipulation and analysis. Exploratory Data Analysis (EDA) was then performed to understand the distribution of claims and detect anomalies. Various classification algorithms, including Decision Tree (DT) and Random Forest (RF), were used to train the model.

List of Figures

Chapter 1

Introduction

1.1 Overview
1.2 Aim/Motivation
1.3 Objective
1.4 Organization of Report

1.1 Overview

Insurance fraud is a significant and growing issue in the financial sector. Fraudulent insurance claims can lead to major financial losses and affect the trust between insurers and clients. Traditional methods of detecting such fraud are often manual, time-consuming, and prone to human error. This project aims to develop an intelligent, data-driven system that can detect potentially fraudulent insurance claims using machine learning algorithms.

The project uses a publicly available dataset and Python-based data science tools to analyze, train, and evaluate models that can classify claims as fraudulent or non-fraudulent. The solution includes steps such as data preprocessing, visualization, model building, and performance comparison, providing a strong foundation for automated fraud detection.

1.2 Aim/Motivation

The aim of this project is to build a reliable fraud detection system that assists insurance companies in identifying suspicious claims early. The motivation comes from the increasing number of fraud cases reported in the insurance domain and the potential of machine learning to discover hidden patterns in large datasets. Automating this process can save time, reduce losses, and improve overall operational efficiency.

1.3 Objective

- To develop a classification model capable of identifying fraudulent insurance claims.

- To analyze the dataset using Exploratory Data Analysis (EDA) and extract meaningful insights.
- To implement and compare machine learning algorithms like Decision Tree, Random Forest, and SVM.
- To evaluate model performance using metrics like accuracy, precision, recall, and F1 score.
- To present visual outputs for better understanding of fraud trends and model results.

1.4 Organization of Report

This report is divided into several chapters to systematically present the entire project process:

Chapter 1: Provides an overview of the project, aim, objectives, and motivation.

Chapter 2: Reviews existing literature and similar systems in fraud detection.

Chapter 3: States the problem and the impact of insurance fraud.

Chapter 4: Lists the hardware and software requirements.

Chapter 5: Describes the system design, block diagrams, GUI, and displays output screenshots.

Chapter 6: Concludes the project findings and outlines future enhancement

Chapter 2

Literature Survey

Reference	Key Findings	Relevance to Project
Kaggle Dataset (Ravalsmit, 2020)	Provided a real-world insurance claims dataset with labeled fraud cases.	Served as the primary dataset for training and evaluating fraud detection models.
Scikit-learn Documentation (2021)	Offered implementation of key classification algorithms like DT, RF, and SVM.	Used for building and comparing machine learning models for fraud detection.
Chen et al. (2017)	Emphasized the effectiveness of data mining in financial fraud detection.	Justified the use of data science techniques for accurate fraud identification.
IBM Fraud Detection Research (2020)	Highlighted the financial impact of undetected insurance fraud.	Motivated the development of automated fraud detection systems.
Seaborn & Matplotlib Docs	Provided data visualization tools for analysis and pattern recognition.	Used to analyze trends and visualize relationships between features and fraud labels.
Research Paper – Sharma et al. (2019)	Compared accuracy of ML models in identifying fraudulent insurance claims.	Helped in selecting and validating the most effective classification models.
Reference	Key Findings	Relevance to Project
Kaggle Dataset (Ravalsmit, 2020)	Provided a real-world insurance claims dataset with labeled fraud cases.	Served as the primary dataset for training and evaluating fraud detection models.

Chapter 3

Problem Statement

A precise articulation of the problem: insurance companies lose millions to fraudulent claims. There is a need for a reliable and automated method to detect these claims using historical data.

Who: This problem primarily affects insurance companies, claim officers, and fraud investigators who are responsible for verifying the legitimacy of insurance claims. Fraudulent claims result in significant financial losses and administrative burden. It also indirectly impacts genuine policyholders by leading to higher premium costs. Detecting fraud effectively is essential for maintaining trust in the insurance process.

What: Insurance fraud involves individuals submitting false or misleading claims to obtain unentitled benefits. With increasing volumes of data, it becomes challenging to manually analyze and detect suspicious activities. Most existing systems lack the intelligence to spot complex fraud patterns. Hence, there is a strong need for an automated, data-driven solution to identify fraud accurately.

When: The issue occurs every time a claim is filed, especially in high-volume environments where insurers must process thousands of claims daily. Over time, fraudsters develop more sophisticated techniques, making it harder to detect fraud with traditional rule-based systems. Therefore, it is critical to address this problem proactively and continuously as part of the claim lifecycle.

Where: Fraud detection challenges arise across various departments in insurance companies, particularly during claim processing, underwriting, and customer verification. It can occur in any type of insurance—health, auto, or life—making it a widespread problem. The lack of centralized fraud detection systems across organizations often leads to inefficiencies in identifying and addressing these cases.

Why: satisfaction, operational costs, and the insurer's reputation. Automating fraud detection with machine learning reduces manual workload and improves accuracy. An effective detection system can flag suspicious claims early, allowing faster response and better decision-making, thereby benefiting both insurers and genuine customers.

Chapter 4

Software Requirements Specification

4.1 Hardware Requirements

To implement and test the Insurance Fraud Detection system effectively, the following hardware resources are required:

- **Processor:** Dual Core or higher (Intel i3/i5, AMD Ryzen 3 or above) for running data models efficiently.
- **RAM:** Minimum 4 GB (8 GB recommended) to handle large datasets and parallel processing tasks.
- **Storage:** At least 20–50 GB of free disk space for storing datasets, libraries, logs, and project files.
- **Graphics Support:** Basic GPU support (optional) for faster computations, especially if deep learning is considered.
- **Network:** Stable internet connection for downloading libraries, datasets, and documentation.

4.2 Software Requirements

This project uses Python and its ecosystem of data science libraries for processing and analyzing data. The following software components are needed:

- **Operating System:** Windows 10/11, Linux (Ubuntu), or macOS
- **Programming Language:** Python 3.7 or above
- **IDE/Text Editor:** Jupyter Notebook (via Anaconda), VS Code, or PyCharm
- **Libraries/Packages:**
    - pandas – for data manipulation
    - numpy – for numerical computations
    - matplotlib & seaborn – for visualizations
    - scikit-learn – for implementing ML algorithms
- **Browser:** Google Chrome / Firefox for any visual output or interface preview
- **Version Control:** Git and GitHub for collaboration and backup (optional but recommended)
- **Virtual Environment:** Anaconda or venv for managing dependencies

Chapter 5

System Design and Result

5.1 Project Block Diagram

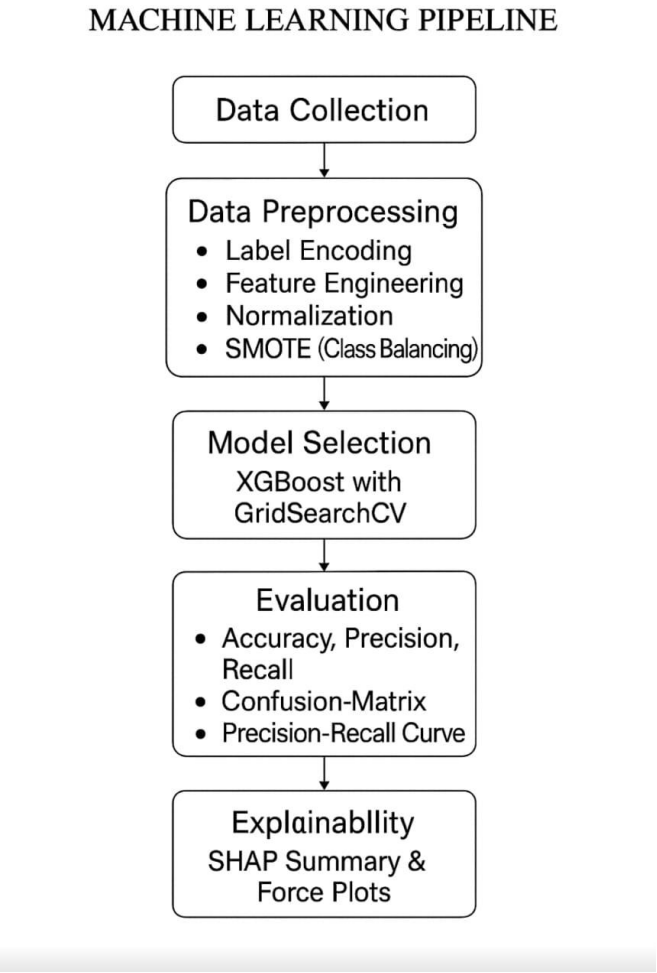

Figure 1.1 : System Block Diagram of Insurance Fraud Detection

5.2 GUI of Working System

While this project is mainly console-based and analytical, the results are presented using graphical outputs and model performance metrics. The following are examples of output screens:

- **Fig 1.2:** Sample view of the dataset after preprocessing
- **Fig 1.3:** Boxplot showing feature-wise distribution of insurance claims based on fraud labels.
- **Fig 1.4:** SHAP summary plot showing the impact of features on model output for fraud detection

- **Fig 1.5:** Confusion matrix of the best-performing model

These outputs help visualize fraud trends and evaluate the effectiveness of each algorithm.

Result:

```
# ----------------------------
# 4. Define Features and Labels
# ----------------------------
X = df.drop("fraud_flag", axis=1)
y = df["fraud_flag"]

# ----------------------------
# 5. Scaling and SMOTE
# ----------------------------
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X_scaled, y)

# ----------------------------
# 6. Train/Test Split
# ----------------------------
X_train, X_test, y_train, y_test = train_test_split(
    X_resampled, y_resampled, test_size=0.25, stratify=y_resampled, random_state=42
)

# ----------------------------
# 7. XGBoost + GridSearchCV
# ----------------------------
xgb_clf = xgb.XGBClassifier(
    eval_metric='logloss',
    random_state=42,
```

Figure 1.2 view of the code while training the dataset

📊 Classification Report:

	precision	recall	f1-score	support
0	0.88	0.88	0.88	102
1	0.88	0.88	0.88	102
accuracy			0.88	204
macro avg	0.88	0.88	0.88	204
weighted avg	0.88	0.88	0.88	204

📉 Confusion Matrix:

```
[[90 12]
 [12 90]]
```

✅ Accuracy: 0.8823529411764706

Figure 1.3 Confusion matrix of the best-performing model

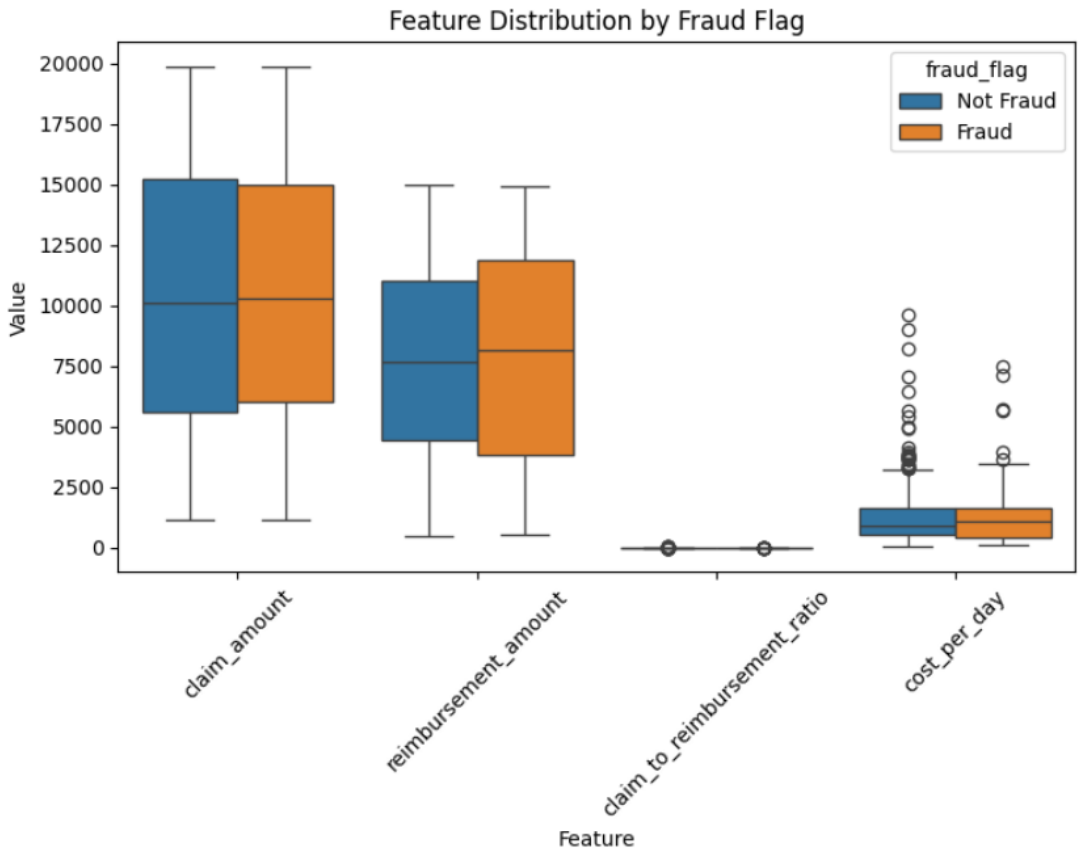

Figure 1.4 Boxplot showing feature-wise distribution of insurance claims based on fraud labels.

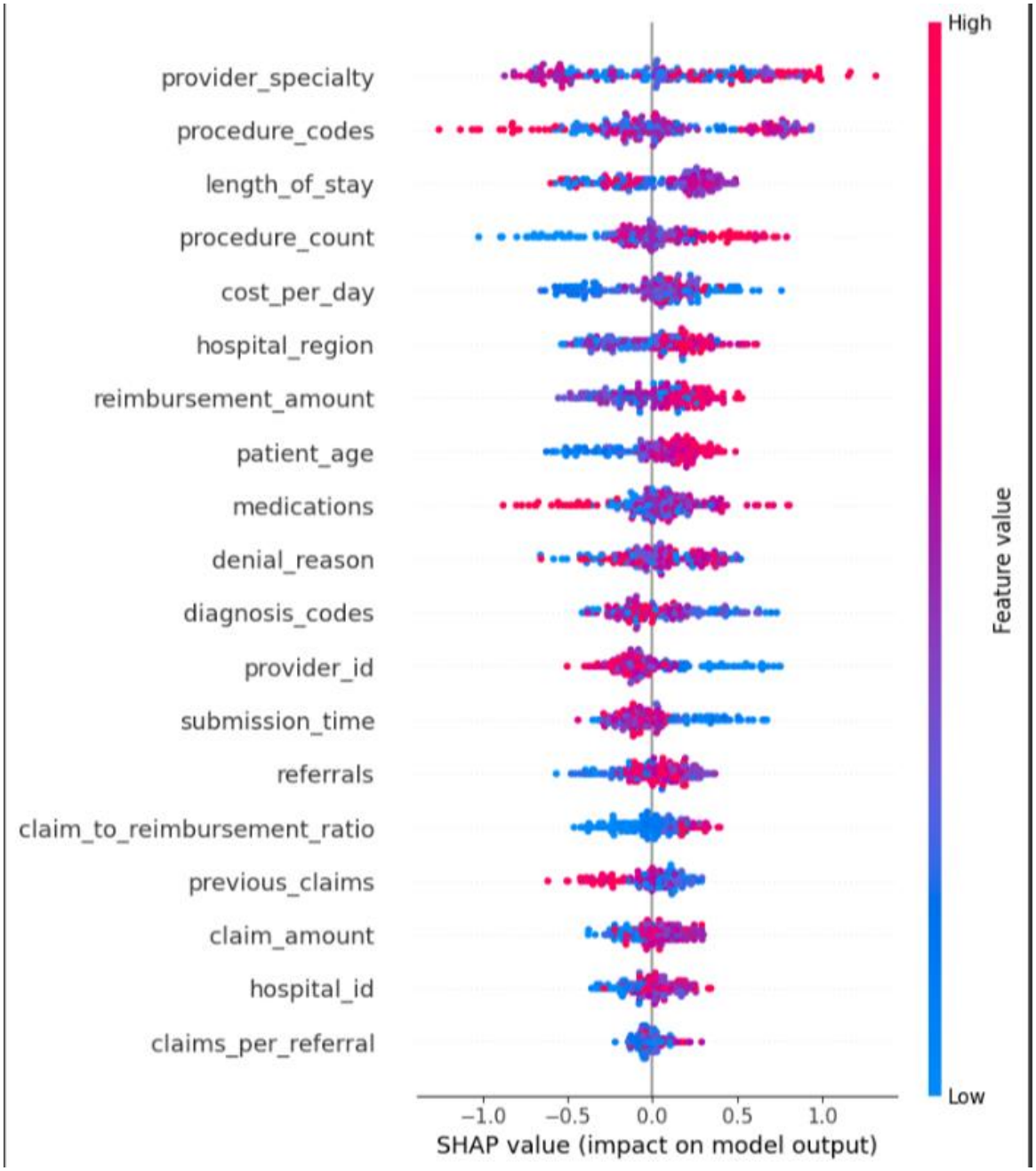

Figure 1. 5 SHAP summary plot showing the impact of features on model output for fraud detection

Chapter 6

Conclusion and Future Scope

The Insurance Fraud Detection project successfully demonstrates the application of machine learning techniques in identifying fraudulent insurance claims. By using a structured pipeline of data preprocessing, visualization, model training, and evaluation, the system provides a reliable method for detecting anomalies in claim data.
Among the algorithms implemented, models like Random Forest and Support Vector Machine showed promising results in terms of accuracy and precision. Visual tools such as count plots and correlation maps helped gain deeper insights into data distribution and patterns commonly associated with fraud.
This project highlights how data science can be effectively used to automate and improve traditional fraud detection processes, ultimately reducing financial losses for insurance companies and speeding up claim verification.

Future Scope:

- **Web Interface Integration:**
  The model can be integrated into a web-based dashboard to allow insurance personnel to upload and analyze claims in real time.
- **Real-Time Prediction API:**
  A RESTful API can be developed to process claims on the fly and return fraud predictions instantly.
- **Deep Learning Models:**
  More complex neural networks or ensemble methods can be tested to improve detection accuracy.
- **Big Data Handling:**
  Integration with distributed platforms like Hadoop or Spark could allow the system to scale for large volumes of claim data.
- **Advanced Feature Engineering:**
  More sophisticated feature extraction methods, such as using time-series data or customer behavior history, can further enhance prediction quality.

References

1. **https://scikit-learn.org**
2. **https://pandas.pydata.org**
3. **https://matplotlib.org**
4. **https://seaborn.pydata.org**
5. **Kaggle Dataset (Ravalsmit)**
6. **Python Official Website –** https://www.python.org
7. **W3Schools – HTML, CSS, and JavaScript Tutorials –** https://www.w3schools.com
8. **MDN Web Docs – Web Development Resources –** https://developer.mozilla.org
9. **Bootstrap Documentation –** https://getbootstrap.com
10. **Stack Overflow – Community Help and Solutions –** https://stackoverflow.com
11. **GeeksforGeeks – Django and Web Development Tutorials –** https://www.geeksforgeeks.org
12. **YouTube – Django Full Stack Tutorial (Programming with Mosh, Corey Schafer, etc.)**
13. **SQLite Official Site –** https://www.sqlite.org