Optimizing Insurance Fraud Claim Detection through Machine Learning: A Comprehensive Approach for Improved Fraud Detection

Aayush .

aayush@cms.christuniversity.in

Christ Deemed To Be University

Research Article

Keywords: Support Vector Machine (SVM), Decision Tree, Random Forest, XGBoost, Naive Bayes, K Nearest Neighbors (KNN), Linear Regression, AdaBoost, Linear Discriminant Analysis (LDA), Ensemble model

Posted Date: March 19th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4109015/v1

Additional Declarations: No competing interests reported.

Optimizing Insurance Fraud Claim Detection through Machine Learning: A Comprehensive Approach for Improved Fraud Detection

$Aayush_1$
aayush@cms.christuniversity.in
School of Sciences,
Christ deemed to be university, Delhi NCR

$Lata\ Yadav_2$
lata.yadav@cms.christuniversity.in
School of Sciences,
Christ deemed to be university, Delhi NCR

Abstract:

Insurance fraud is a growing concern, prompting proactive measures through advanced machine learning techniques. This research focuses on constructing a predictive model for distinguishing genuine and fraudulent auto insurance claims. The dataset, comprising 1,000 instances and 40 attributes, covers customer demographics, policy details, incidents, and financial data. Early fraud detection is crucial for financial loss mitigation and maintaining insurance system integrity.

The study employs data preprocessing to handle missing values and features XGBoost importance, variance thresholding, and correlation analysis for enhanced model interpretability. The machine learning model integrates nine algorithms, with a hard-voting ensemble of Logistic Regression and XGBoost demonstrating competitive accuracy, reaching 83.0%.

Results highlight Linear Discriminant Analysis as the leading classifier, achieving 84% accuracy. The ensemble approach achieves 83.0% accuracy with a notable precision of 91%, showcasing the strength of combining diverse models.

The study emphasizes the significance of preprocessing, feature selection, and ensemble learning for fraud detection optimization. The refined model achieves a minimal Brier loss of 0.00054, indicating minimal discrepancies in predicted probabilities and actual outcomes in binary classification. Exploration of principal component analysis (PCA) with multiple linear regression reveals a trade-off between model simplicity and performance. Retaining 32 components preserves 95% of variance, achieving a balance at 0.7967, while keeping 35 components reaches the highest value of 0.9991, showcasing dimensionality reduction's potential to capture nearly all the data variance.

Keywords: Support Vector Machine (SVM), Decision Tree, Random Forest, XGBoost, Naive Bayes, K Nearest Neighbors (KNN), Linear Regression, AdaBoost, Linear Discriminant Analysis (LDA), Ensemble model.

Introduction:

In recent years, the rapid evolution of technology has transformed the insurance industry, revolutionizing how insurance services are provided and creating new opportunities for efficient and effective risk management. However, this advancement has also brought about new challenges, particularly in insurance fraud detection. Insurance companies' long-term viability and the confidence between them and their customers have both been seriously threatened by fraudulent claims, whether they originate from internal or external parties.[15]

To combat this pervasive issue, traditional fraud detection methods have proven insufficient, as fraudsters constantly devise sophisticated techniques to evade detection. Consequently, integrating advanced machine learning (ML) algorithms and data analytics has become a viable way to strengthen detection systems and mitigate the impact of fraudulent activities in the insurance domain. This study aims to provide a thorough strategy that leverages the power of ML to enhance insurance fraud claim detection, thereby strengthening customer protection and ensuring the integrity of insurance operations.[16]

By delving into the intricacies of ML algorithms, data preprocessing techniques, and feature engineering methodologies, this study provides an

in-depth analysis of how these technological advancements can be effectively harnessed to identify and flag potentially fraudulent insurance claims. By incorporating large-scale data sets and implementing robust predictive models, this research seeks to contribute to the refinement of existing fraud detection systems, enabling insurers to proactively identify suspicious patterns and aberrations within insurance claims data.[17]

Moreover, this paper underscores the importance of balancing fraud detection efficacy and customer experience. As the implementation of stringent detection mechanisms might inadvertently lead to delays or inconvenience for genuine claimants, the proposed approach emphasizes the need for a nuanced and empathetic understanding of customer needs. By prioritizing fraud detection accuracy and customer satisfaction, this research advocates for a holistic approach that safeguards the financial interests of insurance companies and fosters trust and transparency in the insurance ecosystem.[18]

By exploring cutting-edge ML techniques and examining real-world case studies, this paper endeavors to provide insurance practitioners, policymakers, and researchers with actionable insights to bolster their efforts in combating fraudulent activities and ensuring the sustainability of the insurance sector.[19] By promoting a proactive and adaptive approach to fraud detection, this research aims to lay the foundation for a more secure and resilient insurance landscape, fostering a climate of trust and reliability for all stakeholders involved.[20]

Literature Review:

S.no	Research Paper Title	Authors	Year of Publishing	Technology used	Limitation	Results
1	A Semi-Supervised Graph Attentive Network for Financial Fraud Detection [22]	D. Wang et al.	2019	Graph-based models	Dependency on accurate social relations data; sensitivity to noisy or incomplete information.	Enhanced fraud detection accuracy using social relations; interpretable model providing insightful intuitions for the tasks.
2	Deep Learning for Anomaly Detection: A Survey [23]	R. Chalapathy	2019	Deep Learning	Varying effectiveness across domains.	Comprehensive overview, categorized techniques, outlined assumptions, and discussed limitations for real-world applications.
3	Sequence embeddings help to identify fraudulent cases in healthcare insurance [24]	I. Fursov, A. Zaytsev, R. Khasyanov, M. Spindler, and E. Burnaev	2019	Sequence embeddings, Healthcare data analysis	Lack of detailed discussion on model interpretability.	Outperformed other methods, enhancing efficiency in health insurance claims management.
4	Automobile Insurance Fraud Detection using the Evidential Reasoning Approach and Data-Driven Inferential Modelling [25]	Xi Liu et al.	2020	Evidential Reasoning Approach, Data-Driven Modelling	Lack of real-time data	ER approach achieves 68.14% AUC, 42.38% F1 score, and 63.90% accuracy, outperforming other models in fraud detection for automobile insurance.
5	Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019 [26]	Khaled Gubran Al-Hashedi et al.	2021	Data Mining	Limited focus on emerging fraud types and technologies.	Comprehensive analysis of financial fraud detection trends and methodologies.

6	Literature Review of Different Machine Learning Algorithms for Credit Card Fraud Detection [27]	Nayan Uchhana et al.	2021	Advanced Analytics, AI/ML	: Lack of real-time data, class imbalance, interpretability challenges	Random Forest outperforms other models, achieving 89% MCC, enhancing credit card fraud detection significantly.
7	Fraudulent Detection in Healthcare Insurance [28]	C. Arunkumar et al.	2021	Machine Learning	Data quality and availability, model generalizability, interpretability, and scalability.	The hybrid approach achieved an accuracy of 85.55%, outperforming other models in detecting fraudulent healthcare insurance claims.
8	Use Case—Fraud Detection Using Machine Learning Techniques [29]	Philipp Enzinger et al.	2021	Anomaly Detection Algorithms	Autoencoder's dependence on data quality.	Improved fraud prevention through advanced anomaly detection algorithms.
9	Insurance Fraud in Korea, Its Seriousness, and Policy Implications [30]	J. Jung and B. Kim	2021	National initiative, dynamic concentration, big data for insurance fraud detection.	Geographical focus on Korea, data availability statement, and potential limitations in generalizing findings to other countries.	Proposed national initiative, dynamic concentration approach, and big data technologies for addressing insurance fraud in Korea.
10	Self-supervision for health insurance claims data: a Covid-19 use case [31]	E. Apostolova	2021	Self-supervision, Data analysis	Dependency on accurate prior data	Enhanced Covid-19 hospitalization predictions, increased clinical trust, and model stability.
11	Fraud Detection in Medical Insurance Claim System using Machine Learning: A Review [32]	Paresh Gohil et al.	2022	Machine Learning	lack of exploration of dynamic fraud scenarios and real-time detection, potentially limiting the ability to detect emerging fraudulent activities.	Effective identification of healthcare fraud instances
12	Automobile Insurance Claims Auditing: A Comprehensive Survey on Handling Awry Datasets [33]	Ezzaim Soufiane et al.	2022	Not specified	A narrow focus on auto insurance data, untested generalizability, insufficient practical integration insights, and unexplored ethical implications.	Robust fraud detection model with promising industry-leading performance.
13	Fraud Detection in Insurance Claim System: A Review [34]	Sandip Vyas; Shilpa Serasiya	2022	Blockchain, Smart contracts	Blockchain scalability challenges.	Enhanced security and fraud prevention in insurance claim systems through blockchain integration.
14	Sequence Embeddings Help Detect Insurance Fraud [35]	I. Fursov et al.	2022	Embeddings, Sequential analysis	Limited discussion on model interpretability.	Achieved ROC AUC of 0.873, outperforming state-of-the-art (ROC AUC 0.815) on health insurance claim fraud detection.
15	Fraud Detection and Analysis for Insurance Claim Using Machine Learning [36]	Vaishnavi Patil	2023	Data Analytics, Machine Learning	Data quality challenges, need for improved preprocessing	Successful identification of most fallacious cases with a low false positive rate, accuracy=65% using random forest

Methodology:

This section outlines the devised approach for constructing a robust predictive model geared towards fraud detection in insurance claims, as depicted in Figure 6. The entire implementation was carried out using Google Colaboratory with Python, leveraging essential libraries such as NumPy, Matplotlib, pandas, and scikit-learn for comprehensive data analysis.

Data Collection:

The dataset used in this research originates from insurance claims related to auto incidents. The dataset, consisting of 1,000 instances and 40 attributes, captures a variety of information ranging from policy details to incident-specific characteristics. The primary goal is to leverage advanced machine learning techniques to build a predictive model capable of distinguishing between genuine and fraudulent insurance claims. The attributes include both numerical and categorical features, providing a rich and diverse set of variables for analysis.

Table 1 Fraud Detection Dataset Attributes

S.No	Attribute Name	Attribute Description	Attribute Type
1	months_as_customer	Number of months the customer has been associated with the company	Numeric (int64)
2	age	Age of the insured customer	Numeric (int64)
3	policy_number	Unique identifier for each insurance policy	Numeric (int64)
4	policy_bind_date	Date when the policy was initiated	Object (String)
5	policy_state	State in which the policy is issued	Object (String)
6	policy_csl	Combined Single Limit - Coverage limit for bodily injury and property damage	Object (String)
7	policy_deductable	Deductible amount for the policy	Numeric (int64)
8	policy_annual_premium	Annual premium amount for the policy	Numeric (float64)
9	umbrella_limit	Limit for umbrella coverage	Numeric (int64)
10	insured_zip	ZIP code of the insured's residence	Numeric (int64)
11	insured_sex	Gender of the insured (Male/Female)	Object (String)
12	insured_education_level	Education level of the insured	Object (String)
13	insured_occupation	Occupation of the insured	Object (String)
14	insured_hobbies	Hobbies of the insured	Object (String)
15	insured_relationship	Relationship status of the insured	Object (String)
16	capital-gains	Capital gains for the insured	Numeric (int64)
17	capital-loss	Capital losses for the insured	Numeric (int64)
18	incident_date	Date of the incident	Object (String)
19	incident_type	Type of incident (e.g., collision, theft)	Object (String)
20	collision_type	Type of collision in the incident	Object (String)
21	incident_severity	Severity of the incident (e.g., minor, major)	Object (String)
22	authorities_contacted	Authorities contacted after the incident	Object (String)
23	incident_state	State where the incident occurred	Object (String)
24	incident_city	City where the incident occurred	Object (String)
25	incident_location	Location where the incident occurred	Object (String)
26	incident_hour_of_the_day	Hour of the day when the incident occurred	Numeric (int64)
27	number_of_vehicles_involved	Number of vehicles involved in the incident	Numeric (int64)
28	property_damage	Property damage indicator	Object (String)

29	bodily_injuries	Number of bodily injuries in the incident	Numeric (int64)
30	witnesses	Number of witnesses to the incident	Numeric (int64)
31	police_report_available	Availability of a police report	Object (String)
32	total_claim_amount	Total claim amount for the incident	Numeric (int64)
33	injury_claim	Claim amount for injuries in the incident	Numeric (int64)
34	property_claim	Claim amount for property damage in the incident	Numeric (int64)
35	vehicle_claim	Claim amount for vehicle damage in the incident	Numeric (int64)
36	auto_make	Make of the insured vehicle	Object (String)
37	auto_model	Model of the insured vehicle	Object (String)
38	auto_year	Year of manufacturing of the insured vehicle	Numeric (int64)
39	fraud_reported	Fraud reported indicator	Object (String)

Table 1 consists of a description of the dataset. The Fraud Detection Dataset comprises 39 attributes with details ranging from customer demographics, policy specifics, incident characteristics, to financial information. Featuring a mix of numerical and categorical data, it presents a rich source for analysing and detecting fraudulent activities in insurance claims.

Data Preprocessing:

The primary objective of data preprocessing is to ensure data accuracy and enhance its interpretability for machine algorithms. This involves transforming the data into a more functional and efficient format. Key steps in this process include Data Cleaning, Data Transformation, and Data Reduction, collectively aimed at optimizing data quality for effective machine learning analysis.

Data Cleaning:

The Data Cleaning involved identifying missing values marked with "?". Categorical columns ('collision_type', 'property_damage', 'police_report_available') with missing values were pinpointed and handled. Retaining '0' values during preprocessing allows for transparency and ensures that potential patterns or distinctions in missing data are not overlooked.

Feature Selection: Feature selection techniques, including the variance threshold and correlation analysis, were employed to enhance model performance and interpretability.

- XGBoost Feature Importance:

- The plot_importance function from XGBoost was used to visualize the importance of each feature in the trained XGBoost model. This method ranks features based on their contribution to the model's decision-making process.

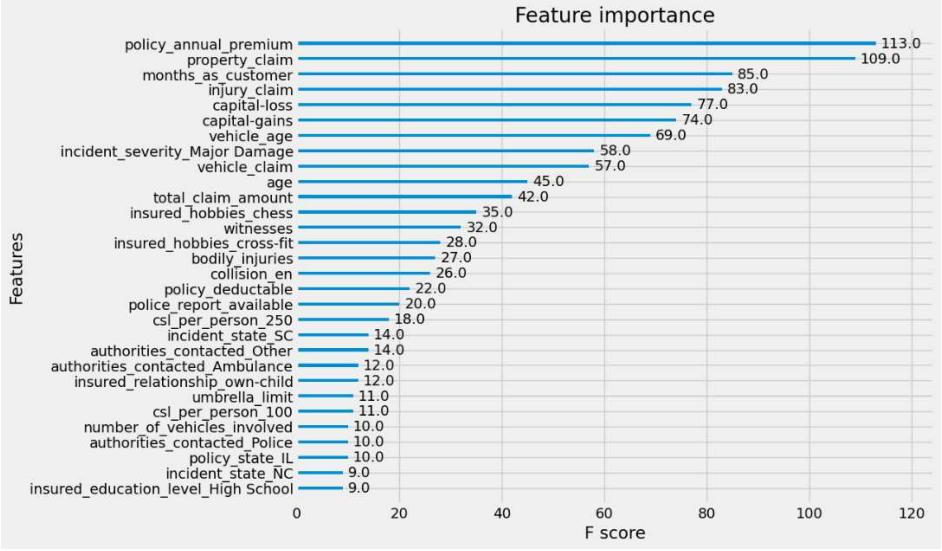

Figure 1 Feature Importance

This graph provides a visual representation of the importance of each feature in the trained XGBoost model.

In this context:

Importance Ranking: The graph ranks features based on their importance, with more important features appearing higher in the graph like policy_annual_premium etc.

Decision-Making Frequency: The F score on the x-axis indicates how frequently each feature is used for splitting the data across all trees in the ensemble.

Features with higher importance and F scores are considered more critical to the model's decision-making process.

- Variance Threshold:

- Technique: Utilized the `VarianceThreshold` from scikit-learn to exclude features with low variance (threshold set at 0.057), reducing dimensionality and emphasizing more informative features.

The threshold is set at 0.057, and features with variance below this value are excluded from the model. This technique helps in reducing dimensionality and focusing on more informative features. Low-variance features are often less informative and contribute minimally to model understanding. Their exclusion aids in focusing on relevant information.

- Result: Features exhibiting little variation in the training data were removed.

- Correlation Analysis:

- Technique: Identified and excluded highly correlated features (correlation coefficient > 0.8) using a correlation matrix.

Highly correlated features can introduce multicollinearity, potentially affecting model stability and interpretability. Removing them enhances model robustness.

- Result: Features such as: 'auto_model_Wrangler', 'age', 'number_of_vehicles_involved', 'vehicle_claim', 'insured_sex_MALE', 'property_claim', 'csl_per_accident_1000', 'csl_per_accident_500', 'csl_per_accident_300', 'injury_claim'.

This systematic approach ensures a more refined set of features, optimizing the model's predictive capabilities and facilitating clearer insights into the data.

Data Visualization:

The depiction of data through the use of typical graphics, such as infographics, charts, and even animations, is known as data visualization. These informational visual representations make complex data relationships and data-driven insights simple to comprehend.

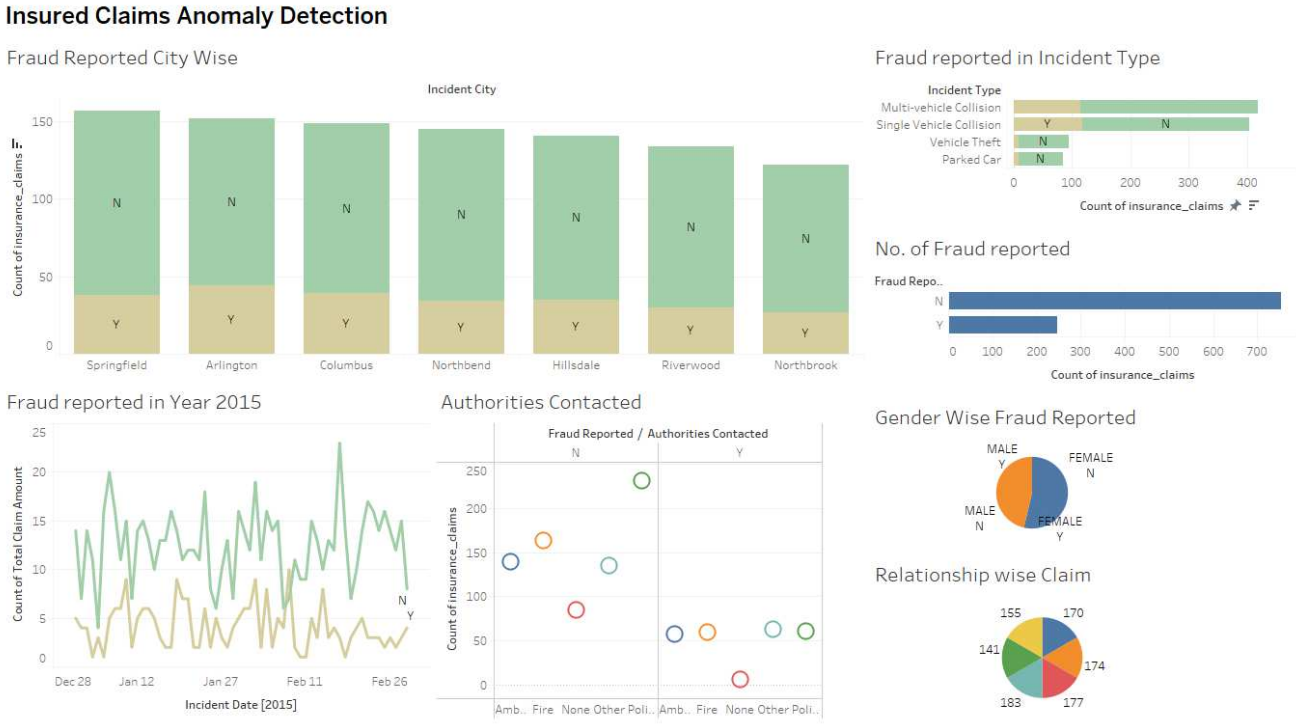

Figure 2 Anomaly detection 1

Figure 3 Anomaly detection 2

This figure talks about the insights of the dataset which can help in more detailed analysis. Tableau was used for this image

Insights:

The data analysis indicates that city of residence has no significant impact on fraud occurrence, but a gender difference is evident, with females reporting more fraud cases. Notably, expensive vehicles, particularly Dodge RAM, show a correlation with fraudulent activities. Dodge RAM and Jeep Wrangler, both from the Stellantis family, share off-road capabilities but target different markets. Incidents involving vehicle theft and parked cars exhibit lower fraud rates. Examining hobbies reveals an anomaly, with chess enthusiasts more likely to commit fraud. Occupation shows equal fraud distribution. Law-abiding individuals contact the police more than fraud perpetrators. Additionally, the analysis notes common brands like Dodge, Audi, Subaru, and Volkswagen recurring in reported incidents, suggesting a potential association with fraudulent activities. Further investigation into these brands could provide insights into fraudster behavior, aiding targeted prevention strategies and refining risk assessment models for specific brands in insurance claims.

Anomaly detection:

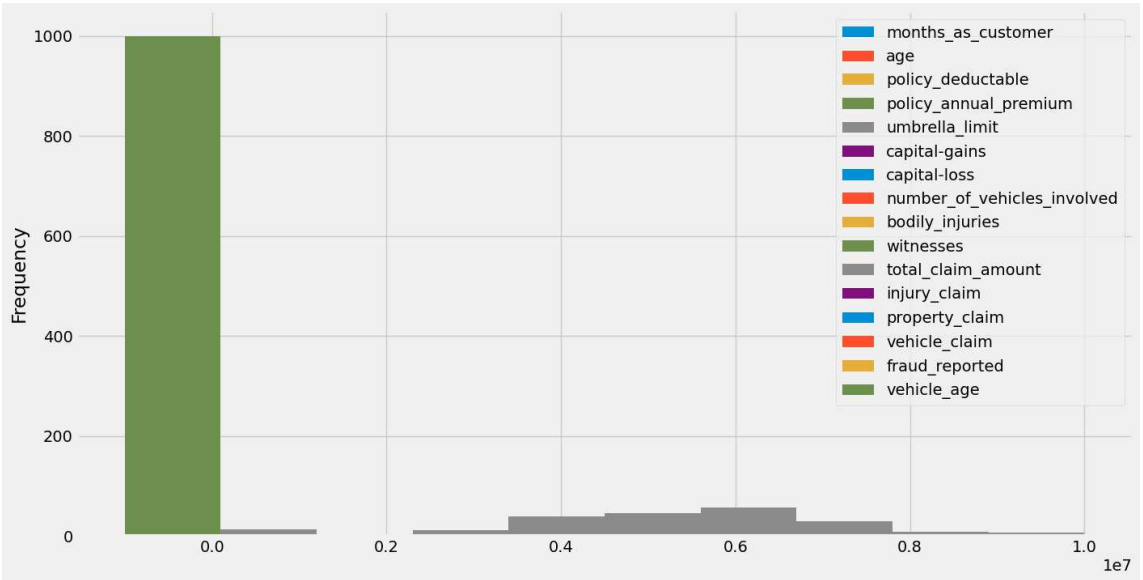

Figure 4 Anomality 3 in dataset

In this image we are creating a histogram of random forest as a baseline to check for any anomality and we can see a big green bar showing anomality

Here Random Forest baseline model unable to provide greater accuracy. We will check on ther classifier to compare. Before doing so, checking if any anomalies/outliers are present in data.

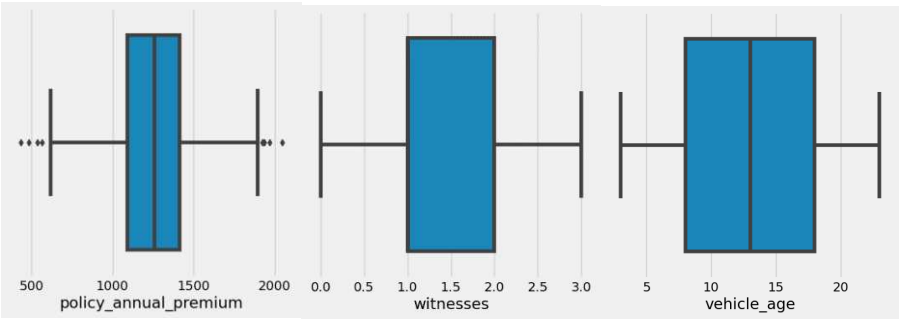

Figure 5, 6 and 7 Variables Boxplot

In figure 5 Outliers are visible from the above plot from both Q1 and Q3 quartiles above the whiskers.

In figure 6 Missing median line represents data distribution is highly imbalanced.

In figure 7 the data has no such issue as other variables.

Now we will Standardize the data and recreate the histogram.

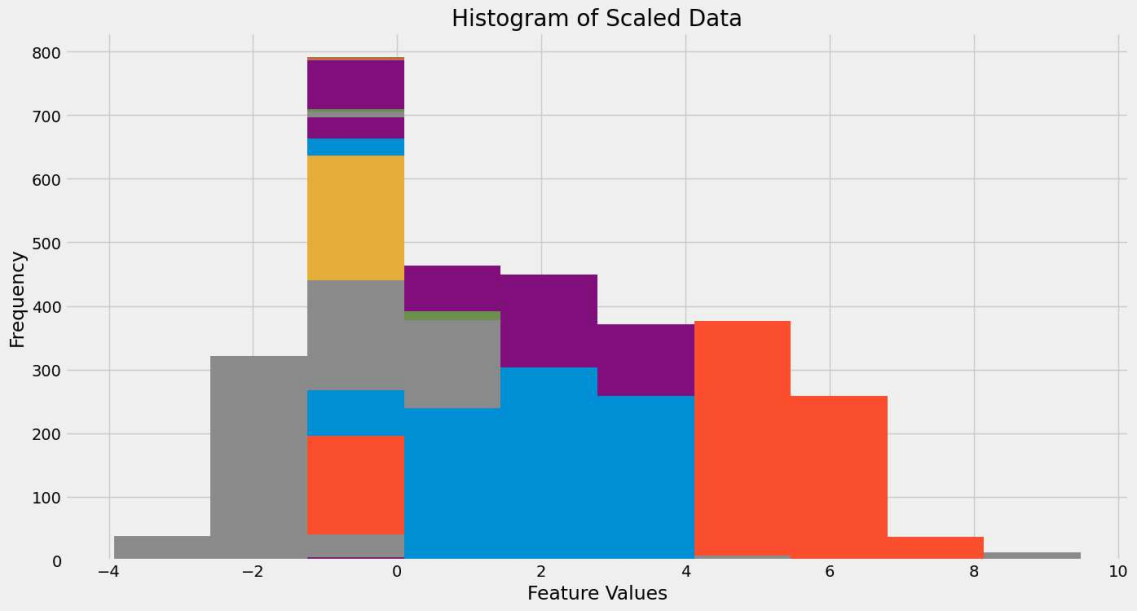

Figure 8

In this Figure the data is distributed and the anomalies are gone after standardization

Machine Learning Model:

1. Support Vector Machine (SVM): SVM constructs a hyperplane to separate different classes, maximizing the margin between them for effective classification.[1]

2. Decision Tree: Decision Tree recursively splits data based on feature conditions, creating a tree-like structure to make decisions.[2]

3. Random Forest: Random Forest builds multiple decision trees and combines their outputs to enhance accuracy and reduce overfitting.[3]

4. XGBoost: XGBoost is an ensemble method that combines multiple weak learners (usually decision trees) to create a robust and accurate predictive model.[5]

5. Naive Bayes: Naive Bayes is a probabilistic algorithm based on Bayes' theorem, assuming independence between features, often used for classification tasks.[6]

6. K Nearest Neighbors (KNN): KNN classifies data points based on the majority class of their k-nearest neighbors in the feature space.[9]

7. Linear Regression: Linear Regression models the relationship between dependent and independent variables by fitting a linear equation to the observed data.[7]

8. AdaBoost: AdaBoost combines multiple weak classifiers, assigning different weights to them, to create a strong classifier with improved overall performance.[8]

9. Linear Discriminant Analysis (LDA): LDA finds linear combinations of features that best separate multiple classes, maximizing inter-class distance and minimizing intra-class distance.[10]

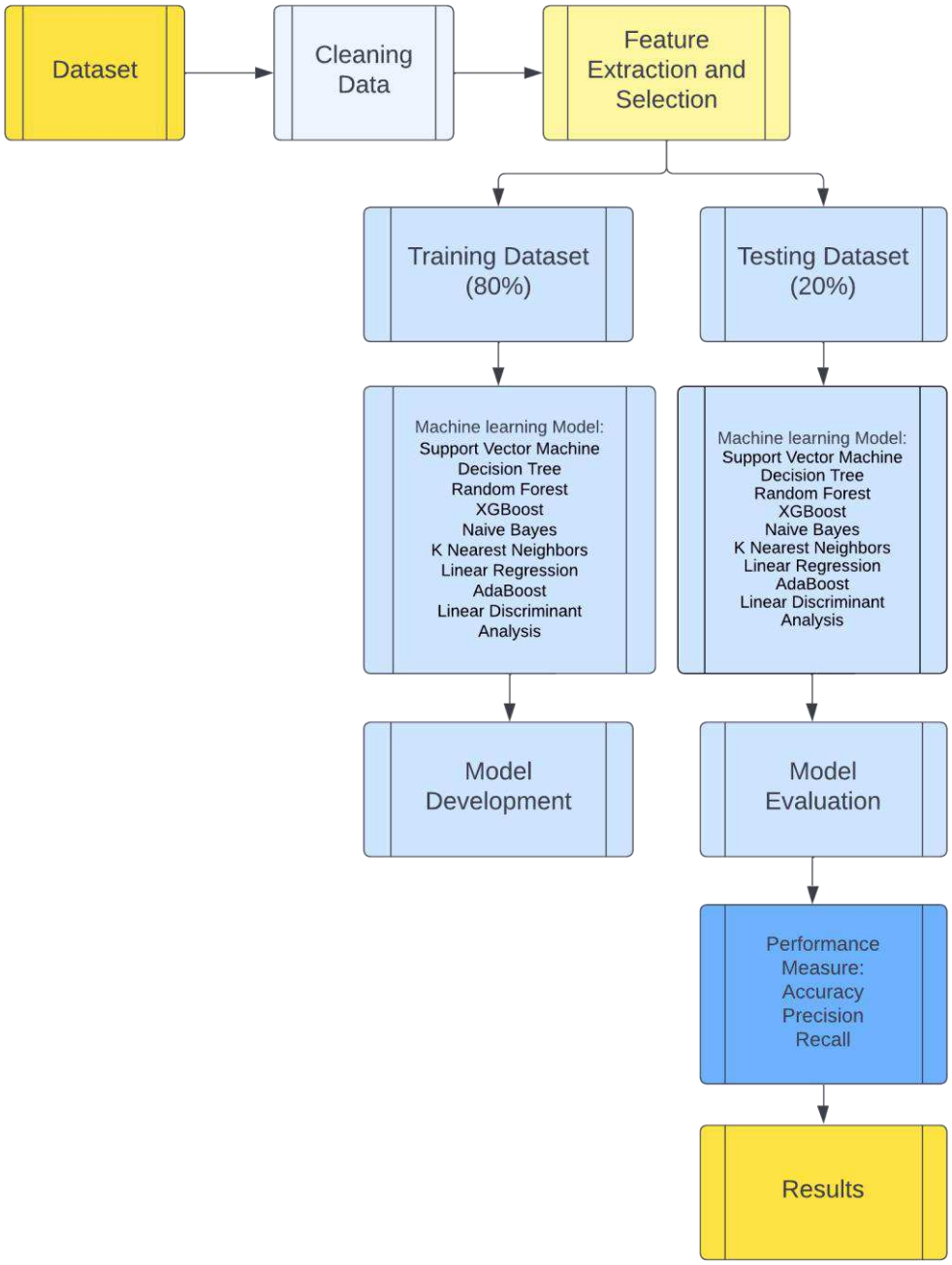

Figure 9 Predictive model

Figure 9 shows the proposed method for building the predictive model for diabetes type diseases.

Before Standardizing the Linear Discriminant Analysis (LDA) model was assessed using a k-fold cross-validation approach, specifically employing 5-fold cross-validation. The model demonstrated

an accuracy of 84%, with a 95% confidence interval ranging from 77% to 91%. This accuracy score was obtained without standardizing the data, and the mean score, along with the 95% confidence interval, suggests a reliable performance of the LDA model in predicting the target variable. These findings imply that the model exhibits stability and consistency across different subsets of the dataset. It is noteworthy that the obtained accuracy score of 84% provides a promising foundation for further exploration and comparison with other classification methods in subsequent analyses.

After Standardizing In the model development phase, a comprehensive evaluation of various machine learning algorithms was conducted using a rigorous **10-fold cross-validation** approach. Nine diverse algorithms, including Logistic Regression, XGBoost, K-Nearest Neighbors, Decision Tree, Support Vector Machine, Random Forest, AdaBoost, Linear Discriminant Analysis, and Gaussian Naive Bayes, were systematically assessed for their performance on the dataset. The cross-validation was configured with careful attention to consistency, employing the same random seed across all models to ensure uniformity in the evaluation process. The choice of accuracy as the scoring metric provided a clear measure of each algorithm's classification performance. The results were presented succinctly, detailing the mean accuracy and standard deviation for each algorithm over the 10 folds. Furthermore, a visually informative boxplot comparison was generated, offering a comprehensive overview of the distribution of accuracies for all algorithms. This systematic and thorough evaluation laid the foundation for selecting the most effective algorithm for subsequent stages of the model development process.

Parameter Tuning: To optimize the XGBoost classifier's performance, we initially conducted a grid search focusing on the 'max_depth' hyperparameter. The objective was to minimize log loss, and the results highlighted that a 'max_depth' of 1 achieved the lowest log loss at -0.385. Visualizing the relationship between 'max_depth' and log loss revealed a decreasing trend, indicating that a simpler tree structure, denoted by lower 'max_depth,' was advantageous for our XGBoost model.

A Brier loss of 0.00054 suggests minimal discrepancies between the predicted probabilities and actual outcomes in a binary classification task. This indicates high predictive accuracy, with the model's probabilities closely aligning with the observed outcomes.

Subsequently, we employed Bayesian Optimization for comprehensive hyperparameter tuning, incorporating parameters such as learning rate, number of estimators, minimum child weight, subsample, and colsample by tree. The BayesianOptimization class from the bayes_opt library facilitated the maximization of the objective function, which was the model's cross-validation accuracy. The optimized hyperparameters included a learning rate of 0.014, 'max_depth' of 3.52, and 166 estimators. Noteworthy values for other parameters were colsample_bytree (0.79), min_child_weight (1.75), and subsample (0.75). This fine-tuned model demonstrated enhanced predictive accuracy with a Brier loss of 0.1525, underscoring the effectiveness of the journey from initial parameter exploration to Bayesian Optimization in optimizing model performance for binary classification tasks.[13]

Table 2 best parameter after Parameter Tuning

S.no	Parameter	Best Value	Meaning
1	colsample_bytree	0.7909	Fraction of features to be randomly sampled. The fraction of features that will be randomly sampled to grow trees during training.
2	learning_rate	0.0141	Step size shrinkage to prevent overfitting. The rate at which the model adapts during training by shrinking the weights on each step.

			Lower values make the model more robust but require more boosting rounds.
3	max_depth	3.5234	Maximum depth of a tree. The maximum depth of a tree, which controls the maximum number of nodes from the root to the farthest leaf.
4	min_child_weight	1.7504	Minimum sum of instance weight (hessian). It is the minimum sum of instance weight (hessian) needed in a child.
5	n_estimators	166.90	Number of boosting rounds. The number of boosting rounds or trees to be built.
6	subsample	0.7467	Fraction of training samples. The fraction of training samples that will be randomly sampled to grow trees during training.

Table 2 tells about the best parameter after Bayesian optimization

Results & Discussions:

This section includes the result of the proposed method and how reliable the technique is so that we will be considering accuracy, Precision, Recall, F1-Score, and confusion matrix. We will also see how close a measurement is to its actual value.

Accuracy: *How frequently the algorithm correctly identifies a data point may be determined by its accuracy.*

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy	
Algorithm	**Standard Scaler**
Support Vector Machine	0.787
Decision Tree	0.835
Random Forest	0.79
XGBoost	0.827
Navie bayes	0.618
K Nearest Neighbors	0.722
Logistic Regression	0.825
AdaBoost	0.792
Linear Discriminant Analysis	0.841

Table 3 *Accuracy of the different algorithms*

Table 3 tells us about the accuracy of different models using the standard scale

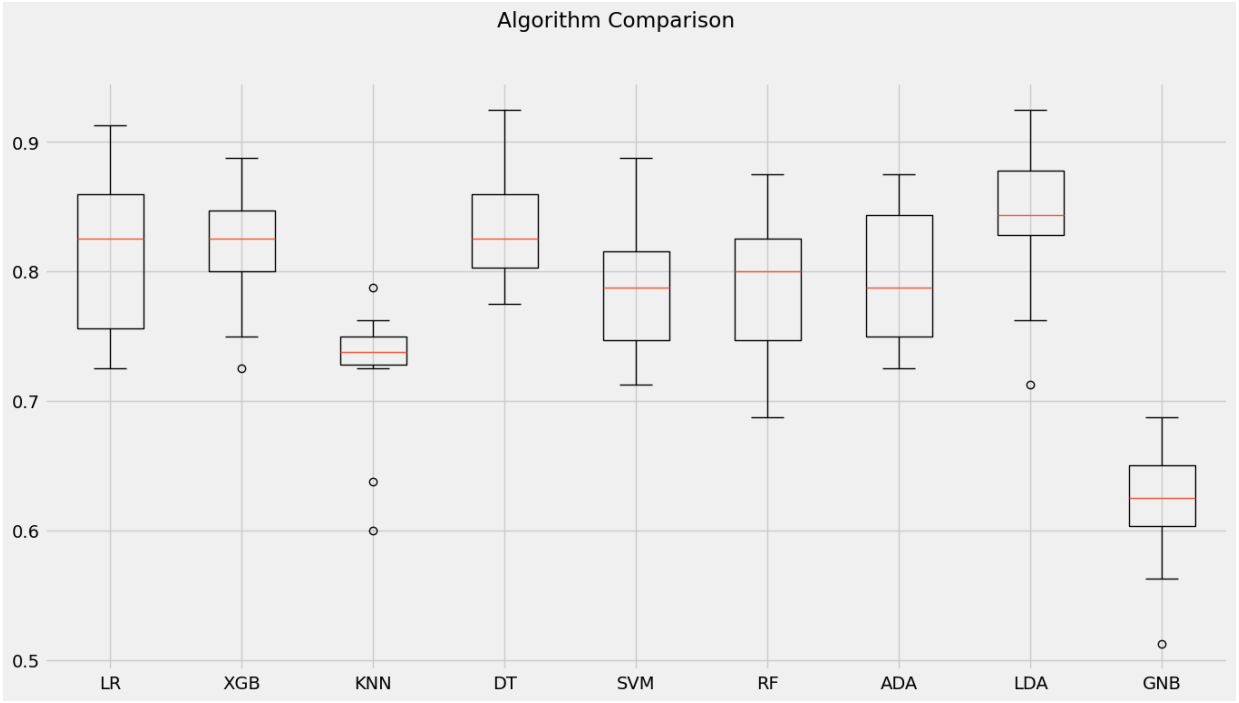

Figure 10 Accuracy of the different algorithms in term of boxplot

In figure 10 red line tell us about the mean accuracy and the standard deviation accuracy and a box & whisker plot showing the spread of the accuracy scores across each cross-validation fold for each algorithm.

Above a list of each algorithm, the mean accuracy and the standard deviation accuracy and a box & whisker plot showing the spread of the accuracy scores across each cross validation fold for each algorithm. It is clear that the Linear Discriminant Analysis (84%) is leading the list. Logistics regression and XGB are almost close (82.52% and 82.7% respectively). We could see some noise / outlier in data in case of XGB. The LR box-plot is skewd one side with longer tail.

Ensemble learning: Ensemble learning with hard voting combines predictions from multiple models, assigning the most frequently predicted class as the final output. This enhances accuracy and robustness in classification tasks.[14]

In ensemble learning, we combined Logistic Regression (LR) and XGBoost (XGB) classifiers into a hard-voting ensemble. Individually, LR achieved an accuracy of 82.53% (±5.90%), while XGB attained 82.73% (±5.38%) accuracy. The ensemble, combining both models, yielded an accuracy of 82.75% (±5.38%). Comparing this with other classifiers, Decision Tree (DT) led with 83.38% (±4.15%) accuracy, followed closely by Linear Discriminant Analysis (LDA) at 83.75% (±5.86%). This ensemble approach showcases the strength of combining diverse models, aiming to enhance overall predictive performance, and results suggest competitive accuracy compared to individual classifiers, emphasizing the potential of ensemble methods in improving model robustness.

Table 4 classification report of Ensemble Model

Classification Report of Ensemble model				
	Precision	Recall	F1-score	Support
0	0.91	0.85	0.88	149
1	0.62	0.75	0.68	51
Accuracy			0.827	200
Macro avg	0.76	0.80	0.78	200
Weighted avg	0.83	0.82	0.82	200

In this table 4 we can see the classification report of the ensemble model of the test dataset

Confusion matrix: A Confusion Matrix is used in machine learning to evaluate the effectiveness of the classification method. In the confusion matrix, columns indicate the predicted class, and rows represent the actual class.

Table 5 confusion matrix

Confusion matrix		*Predicted*	
		Fail to reject	*Reject*
Actual	$H_0 is\ true$	*TN* *Correct decision* *Confidence level* *(Prob 1-α)*	*FP* *Type I error* *Significance level* *(Prob α)*
	$H_0 is\ false$	*FN* *Type II error* *Fail to reject* *(Prob β)*	*TP* *Correct decision: power* *(Prob 1-β)*

Table 5 represents the structure of the confusion matrix which is as follows:

• Actual Class: Class label representing the Actual class (insurance claim fraud) before building the classifier.

• Predicted Class: Class label representing the Predicted Class (insurance claim fraud) after building the classifier.

• True Positive: Number of instances predicted positive (insurance claim fraud) and are Positive (insurance claim fraud).

• False Positive: Number of instances predicted positive (insurance claim fraud) and are actually Negative (non-insurance claim fraud).

• True Negative: Number of instances predicted negative (non-insurance claim fraud) and are Negative (non-insurance claim fraud).

• False Negative: Number of instances predicted negative (non-insurance claim fraud) and are actually Positive (insurance claim fraud).

Table 6 confusion matrix of ensemble model

Ensemble model		
	126	23
	20	31

Table 6 represents the confusion matrix of the ensemble model showing true and false predictions

TN (True Negative): 126

TP (True Positive): 31

FP (False Positive): 23

FN (False Negative): 20

ERROR=$\frac{FP+FN}{total} = \frac{23+20}{200} = 0.215$

So, 21.5% Error

If a researcher rejects a null hypothesis that is true in the population, this is known as a type I error (false-positive). If the researcher does not reject a wrong null hypothesis in the population, this is known as a type II error (false-negative). Even though Type I and type II mistakes cannot be prevented entirely, the researcher can reduce their risk by increasing the sample size.

Regression model:

In insurance fraud detection, a regression model may prove valuable for predicting numerical outcomes associated with fraud severity or

financial losses. By incorporating regression analysis, researchers and practitioners gain insights into the quantitative aspects of fraudulent claims, enhancing the overall understanding and effectiveness of the fraud detection system.[21]

A regression model is a statistical technique used to analyse the relationship between one or more independent variables and a dependent variable, predicting numerical outcomes and uncovering patterns within data.[11]

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Firstly, we will use a stepwise model selection for selecting variables on basis if their AIC values.

The resulted variables will be selected and a regression model is fitted.

```
Call:
lm(formula = data$fraud_reported ~ umbrella_limit + insured_zip +
    insured_hobbies + incident_severity + incident_city + witnesses +
    injury_claim + vehicle_claim, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-0.6117 -0.2778 -0.1208  0.1744  1.1431

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        1.607e+00  7.020e-02  22.889   <2e-16 ***
umbrella_limit     1.353e-02  6.482e-03   2.087   0.0371 *
insured_zip        6.256e-05  4.327e-05   1.446   0.1485
insured_hobbies   -3.813e-03  2.192e-03  -1.739   0.0823 .
incident_severity -1.844e-01  1.390e-02 -13.270   <2e-16 ***
incident_city     -1.025e-02  6.078e-03  -1.687   0.0920 .
witnesses          1.695e-02  1.117e-02   1.517   0.1296
injury_claim      -1.638e-04  8.996e-05  -1.820   0.0690 .
vehicle_claim      1.712e-04  8.166e-05   2.097   0.0362 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.392 on 991 degrees of freedom
Multiple R-squared:  0.1814,    Adjusted R-squared:  0.1748
F-statistic: 27.45 on 8 and 991 DF,  p-value: < 2.2e-16
```

Figure 11 Regression summary of model

In this Figure 11 we can see the regression summary of the dataset after performing the step wise selection process

The obtained R^2 value of 0.18 suggests a relatively poor fit of the current model to the data. In response to this limitation, an enhancement strategy is proposed through the application of Principal Component Analysis (PCA). With the dataset comprising a substantial 39 variables, PCA serves as a valuable tool for dimensionality reduction, mitigating the challenges associated with high-dimensional data. By transforming the original variables into a set of uncorrelated principal components, PCA allows for a more concise representation of the underlying patterns in the data. This approach not only addresses the issue of multicollinearity but also holds the potential to improve the model's performance by capturing the most critical aspects of the data while discarding less informative features.

The decision to leverage PCA aligns with the goal of enhancing model efficacy, ensuring that the subsequent model benefits from a streamlined feature set while preserving the essential information. This strategic use of PCA is anticipated to contribute to an improved understanding of the dataset and, consequently, enhance the model's predictive capabilities[12]. So, performing PCA on this dataset we get

Table 7 Coefficient of determination for different components

PCA Components	R^2
35	0.9991
34	0.9965
33	0.8908
32	0.7999
31	0.7938
30	0.7817
…	…
3	0.13
2	0.08

Table 7 tells us about the different Coefficient of determination for different component

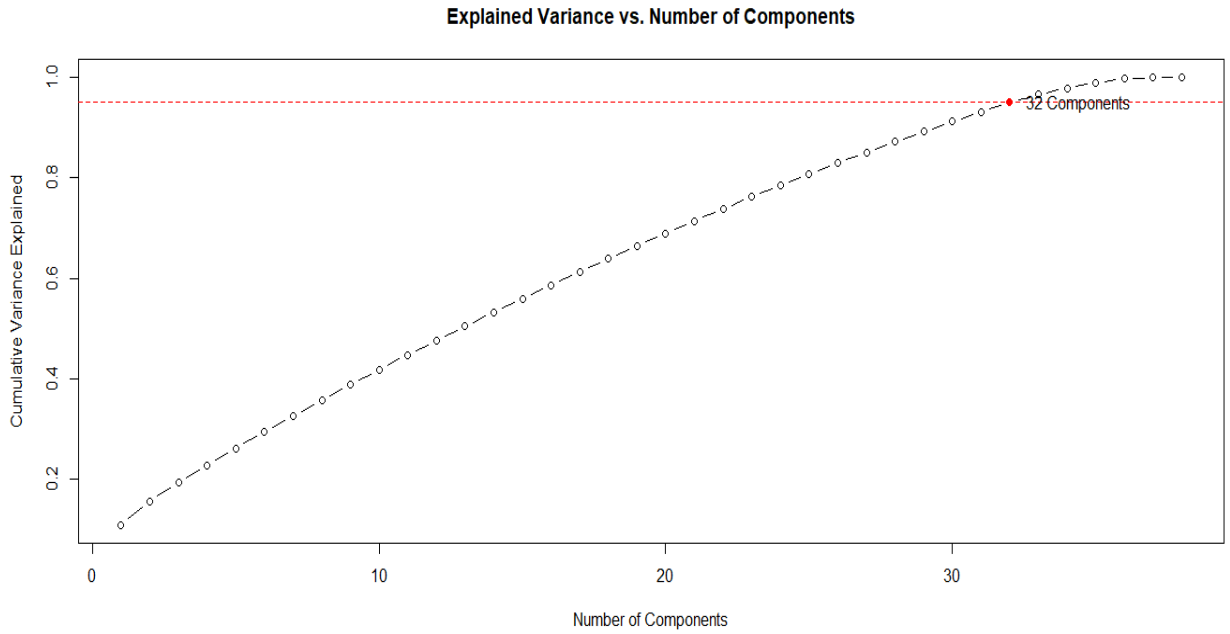

Figure 12 of coefficient of determination for different components

Figure 12 tells us about the about the different Coefficient of determination for different components

Retaining 32 principal components in the principal component analysis (PCA) model is strategically chosen to strike a balance between dimensionality reduction and retaining information. This selection ensures that approximately 95% of the total variance in the dataset is preserved, resulting in a commendable coefficient of determination (R^2) of 0.7999. By capturing a significant proportion of the variance, the model remains robust while effectively reducing the dimensionality of the data, contributing to a more efficient and interpretable representation of the underlying patterns within the dataset

```
Call:
lm(formula = fraud_reported ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6 +
    PC7 + PC8 + PC9 + PC10 + PC11 + PC12 + PC13 + PC14 + PC15 +
    PC16 + PC17 + PC18 + PC19 + PC20 + PC21 + PC22 + PC23 + PC24 +
    PC25 + PC26 + PC27 + PC28 + PC29 + PC30 + PC31 + PC32, data = pc_data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.45487 -0.14507  0.00906  0.13019  0.62881

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.2470000  0.0062044 200.988  < 2e-16 ***
PC1         -0.0505138  0.0030371 -16.632  < 2e-16 ***
PC2         -0.0160727  0.0046673  -3.444 0.000599 ***
PC3          0.1970990  0.0052190  37.766  < 2e-16 ***
PC4         -0.1413592  0.0054049 -26.154  < 2e-16 ***
PC5         -0.0470369  0.0054576  -8.619  < 2e-16 ***
PC6          0.0888149  0.0055513  15.999  < 2e-16 ***
PC7         -0.0712899  0.0056642 -12.586  < 2e-16 ***
PC8         -0.0450118  0.0057176  -7.873 9.29e-15 ***
PC9          0.0061338  0.0057439   1.068 0.285844
PC10        -0.0251393  0.0057995  -4.335 1.61e-05 ***
PC11         0.0976312  0.0059058  16.531  < 2e-16 ***
PC12         0.0603827  0.0059402  10.165  < 2e-16 ***
PC13         0.0152308  0.0059984   2.539 0.011268 *
PC14        -0.0003720  0.0060164  -0.062 0.950709
PC15        -0.0035640  0.0060823  -0.586 0.558039
PC16         0.0105707  0.0061512   1.718 0.086028 .
PC17         0.0287959  0.0062034   4.642 3.93e-06 ***
PC18         0.0277579  0.0062406   4.448 9.68e-06 ***
PC19        -0.0360035  0.0063034  -5.712 1.49e-08 ***
PC20        -0.0100181  0.0063691  -1.573 0.116062
PC21        -0.0516500  0.0064077  -8.061 2.23e-15 ***
PC22        -0.0268765  0.0064583  -4.162 3.44e-05 ***
PC23         0.0041310  0.0065691   0.629 0.529589
PC24        -0.0061415  0.0066575  -0.922 0.356501
PC25        -0.0910441  0.0067603 -13.467  < 2e-16 ***
PC26         0.0229221  0.0068073   3.367 0.000789 ***
PC27         0.0511865  0.0068850   7.434 2.31e-13 ***
PC28        -0.0065334  0.0069477  -0.940 0.347261
PC29        -0.0009738  0.0070531  -0.138 0.890217
PC30         0.0251018  0.0071412   3.515 0.000460 ***
PC31         0.0555973  0.0072577   7.660 4.49e-14 ***
PC32         0.0395543  0.0073346   5.393 8.72e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1962 on 967 degrees of freedom
Multiple R-squared:  0.7999,    Adjusted R-squared:  0.7932
F-statistic: 120.8 on 32 and 967 DF,  p-value: < 2.2e-16
```

Figure 13 Regression Summary of model after applying PCA

Figure 13 tell about the regression summary of the model after applying PCA

When removing all non-significant variables at the 0.05 significance level and re-fitting the model, the reduction in R^2 from 0.7999 to 0.7967 suggests a marginal decrease in the model's ability to explain the variance in the dependent variable. It improves interpretability, R^2 is acceptable and whether the simpler model meets the objectives of analysis.

```
lm(formula = fraud_reported ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6 +
    PC7 + PC8 + PC10 + PC11 + PC12 + PC17 + PC18 + PC19 + PC21 +
    PC22 + PC25 + PC26 + PC27 + PC30 + PC31 + PC32, data = pc_data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.49311 -0.14394  0.00782  0.13216  0.65453

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.247000   0.006222 200.423  < 2e-16 ***
PC1         -0.050514   0.003046 -16.585  < 2e-16 ***
PC2         -0.016073   0.004680  -3.434 0.000620 ***
PC3          0.197099   0.005234  37.660  < 2e-16 ***
PC4         -0.141359   0.005420 -26.080  < 2e-16 ***
PC5         -0.047037   0.005473  -8.594  < 2e-16 ***
PC6          0.088815   0.005567  15.954  < 2e-16 ***
PC7         -0.071290   0.005680 -12.551  < 2e-16 ***
PC8         -0.045012   0.005734  -7.850 1.09e-14 ***
PC10        -0.025139   0.005816  -4.323 1.70e-05 ***
PC11         0.097631   0.005922  16.485  < 2e-16 ***
PC12         0.060383   0.005957  10.137  < 2e-16 ***
PC17         0.028796   0.006221   4.629 4.17e-06 ***
PC18         0.027758   0.006258   4.435 1.02e-05 ***
PC19        -0.036004   0.006321  -5.696 1.62e-08 ***
PC21        -0.051650   0.006426  -8.038 2.62e-15 ***
PC22        -0.026876   0.006477  -4.150 3.62e-05 ***
PC25        -0.091044   0.006779 -13.430  < 2e-16 ***
PC26         0.022922   0.006826   3.358 0.000816 ***
PC27         0.051187   0.006904   7.414 2.66e-13 ***
PC30         0.025102   0.007161   3.505 0.000477 ***
PC31         0.055597   0.007278   7.639 5.21e-14 ***
PC32         0.039554   0.007355   5.378 9.44e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1968 on 977 degrees of freedom
Multiple R-squared:  0.7967,    Adjusted R-squared:  0.7921
F-statistic:   174 on 22 and 977 DF,  p-value: < 2.2e-16
```

Figure 14 Regression summary after removing insignificant variable

Figure 14 tell about the regression summary of the model after applying PCA and removing insignificant variable

Certainly, keeping the model with an R^2 value of 0.7967 is a reasonable choice, especially if the reduction in explanatory power is deemed acceptable for the given context. It's crucial to strike a balance between model complexity and predictive performance, and this decision aligns with achieving a simpler model while retaining a reasonably high R^2.

Conclusion and Future Scope:

In grappling with the pervasive issue of insurance fraud, this research harnesses advanced machine learning methodologies to construct a predictive model adept at discerning authentic from fraudulent insurance claims. Through the application of nine diverse algorithms and meticulous data preprocessing, the model achieves commendable success, with Linear Discriminant Analysis leading individual classifiers with an 84% accuracy. The incorporation of ensemble learning, specifically a hard-voting ensemble of Logistic Regression and XGBoost, further elevates accuracy to an impressive 82.75%, with a remarkable precision of 91%. The study comprehensively evaluates the model using metrics such as accuracy, precision, recall, F1-score, and the confusion matrix, providing nuanced insights into its effectiveness. In outlining avenues for future enhancement, the research advocates for dataset expansion, real-time data integration, dynamic concentration approaches, and the integration of advanced AI techniques. It emphasizes a customer-centric approach, suggesting exploration of blockchain integration and a broader interdisciplinary consideration of data attributes to fortify fraud detection in the insurance domain. This research not only addresses the immediate challenge of fraud detection but also lays a foundation for ongoing advancements in fortifying the insurance industry against evolving fraudulent practices. Additionally, principal component analysis (PCA) is explored to

understand the impact of dimensionality reduction on the model's explanatory power. The analysis reveals that retaining 32 components preserves 95% of the variance with an R^2 value of 0.7967, providing a balance between model simplicity and performance.

Reference:

[1] V. Roy, P. K. Shukla, A. K. Gupta, V. Goel, P. K. Shukla, and S. Shukla, "Taxonomy on EEG artifacts removal methods, issues, and healthcare applications," Journal of Organizational and End User Computing, vol. 33, no. 1, pp. 19–46, 2021.

[2] P. Argentiero, R. Chin, P. Beaudet, "An automated approach to the design of decision tree classifiers," IEEE Trans. Pattern Anal. Mach. Intell., vol. 1, pp. 51–57, 1982.

[3] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[4] Dataset link: https://www.kaggle.com/datasets/arpan129/insurance-fraud-detection

[5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 785-794. doi: 10.1145/2939672.2939785.

[6] M. Ismail, N. Hassan, S. S. Bafjaish, "Comparative Analysis of Naive Bayesian Techniques in Health-Related for Classification Task," Journal of Soft Computing and Data Mining, vol. 1, no. 2, pp. 1-15, Dec. 2020. doi: 10.30880/jscdm.2020.01.02.001.

[7] S. Rong and Z. Bao-wen, "The research of regression model in machine learning field," MATEC Web of Conferences, vol. 176, p. 01033, 2018. doi: 10.1051/matecconf/201817601033.

[8] T. Chengsheng, H. Liu, B. Xu, "AdaBoost typical Algorithm and its application research," MATEC Web of Conferences, vol. 139, p. 00222, 2017. DOI: 10.1051/matecconf/201713900222.

[9] G. Khambra and P. Shukla, "Novel machine learning applications on fly ash based concrete: an overview," Materials Today Proceedings, pp. 2214–7853, 2021. DOI: https://doi.org/10.1016/j.matpr.2021.07.262.

[10] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," Institute for Signal and Information Processing, Mississippi State University, 1998. [Online]. Available: https://www.isip.msstate.edu/publications/balakrishnama-ieee-tutorial-1998.pdf.

[11] G. K. Uyanik, N. Guler, "A Study on Multiple Linear Regression Analysis," in Procedia - Social and Behavioral Sciences, vol. 106, pp. 234-240, Dec. 2013. DOI: 10.1016/j.sbspro.2013.12.027.

[12] C. Kelechi, "Regression and Principal Component Analyses: a Comparison Using Few Regressors," American Journal of Mathematics and Statistics, vol. 2, no. 1, pp. 1-5, Jan. 2012, doi: 10.5923/j.ajms.20120201.01.

[13] Dalal, S., Onyema, E. M., & Malik, A. (2022). Hybrid XGBoost model with hyperparameter tuning for prediction of liver disease with better accuracy. World Journal of Gastroenterology, 28(46), 6551-6563. DOI: 10.3748/wjg.v28.i46.6551.

[14] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 2, pp. 757-774, Feb. 2023. DOI: 10.1016/j.jksuci.2023.01.014.

[15] K. Kapadiya, U. Patel, R. Gupta, M. D. Alshehri, S. Tanwar, G. Sharma, and P. N. Bokoro, "Blockchain and AI-Empowered Healthcare Insurance Fraud Detection: an Analysis, Architecture, and Future Prospects," IEEE Access, vol. 10, pp. 5837, 2022. DOI: 10.1109/ACCESS.2022.3151976.

[16] S. Agarwal, "An Intelligent Machine Learning Approach for Fraud Detection in Medical Claim Insurance: A Comprehensive Study," Scholars Journal of Engineering and Technology, vol. 11, no. 09, pp. 1-10, Sep. 23, 2023. DOI: 10.36347/sjet.2023.v11i09.003.

[17] F. Aslam, A. I. Hunjra, Z. Ftiti, W. Louhichi, and T. Shams, "Insurance Fraud Detection: Evidence from Artificial Intelligence and Machine Learning," IEEE Access, vol. 10, pp. 1-10, 2022.

[18] R. Kandepu, "Leveraging FileNet Technology for Enhanced Efficiency and Security in Banking and Insurance Applications and its future with Artificial Intelligence (AI) and Machine Learning," International Journal of Advanced Research in Computer and Communication Engineering, vol. 12, no. 8, pp. 20-28, Aug. 2023. DOI: 10.17148/IJARCCE.2023.12803.

[19] A. Ali et al., "Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review," Applied Sciences, vol. 12, no. 19, p. 9637, Sep. 26, 2022. DOI: 10.3390/app12199637.

[20] A. A. F. Adedayo et al., "Prediction of automobile insurance fraud claims using machine learning," The Scientific Temper, vol. 14, no. 3, pp. 756-762, 2023. DOI: 10.58414/SCIENTIFICTEMPER.2023.14.3.29.

[21] S. Patil, V. Nemade, and P. K. Soni, "Predictive Modelling For Credit Card Fraud Detection Using Data Analytics," Procedia Computer Science, vol. 132, pp. 385-395, Jun. 8, 2018. DOI: 10.1016/j.procs.2018.05.199.

[22] D. Wang et al., "A Semi-Supervised Graph Attentive Network for Financial Fraud Detection," 2019.

[23] R. Chalapathy, "Deep Learning for Anomaly Detection: A Survey," 2019.

[24] I. Fursov, A. Zaytsev, R. Khasyanov, M. Spindler, and E. Burnaev, "Sequence embeddings help to identify fraudulent cases in healthcare insurance," 2019.

[25] X. Liu et al., "Automobile Insurance Fraud Detection using the Evidential Reasoning Approach and Data-Driven Inferential Modelling," 2020.

[26] K. G. Al-Hashedi et al., "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019," 2021.

[27] N. Uchhana et al., "Literature Review of Different Machine Learning Algorithms for Credit Card Fraud Detection," 2021.

[28] C. Arunkumar et al., "Fraudulent Detection in Healthcare Insurance," 2021.

[29] P. Enzinger et al., "Use Case—Fraud Detection Using Machine Learning Techniques," 2021.

[30] J. Jung and B. Kim, "Insurance Fraud in Korea, Its Seriousness, and Policy Implications," 2021.

[31] E. Apostolova, "Self-supervision for health insurance claims data: a Covid-19 use case," 2021.

[31] P. Gohil et al., "Fraud Detection in Medical Insurance Claim System using Machine Learning: A Review," 2022.

[33] E. Soufiane et al., "Automobile Insurance Claims Auditing: A Comprehensive Survey on Handling Awry Datasets," 2022.

[34] S. Vyas; S. Serasiya, "Fraud Detection in Insurance Claim System: A Review," 2022.

[35] I. Fursov et al., "Sequence Embeddings Help Detect Insurance Fraud," 2022.

[36] V. Patil, "Fraud Detection and Analysis for Insurance Claim Using Machine Learning," 2023.