# Insurance Cost Prediction

*Contents*

# INSURANCE COST PREDICTION


## Abstract

In this study, we develop a comprehensive machine learning framework to address four critical tasks within the insurance domain: premium calculation, high-risk customer identification, customer segmentation, and fraud detection. Utilizing a dataset encompassing demographic and behavioural features, we employ advanced ensemble techniques, specifically stacking ensembles, to improve model performance across all tasks.

For premium calculation, we implement a stacking regressor that combines the predictive capabilities of Random Forest Regressor, Gradient Boosting Regressor, Support Vector Regressor, and XGBoost Regressor. This ensemble model aims to accurately predict insurance premiums based on features such as age, sex, BMI, number of children, smoking status, and region.

To identify high-risk customers, we deploy a stacking classifier comprising Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, Support Vector Classifier, and XGBoost Classifier. This model differentiates between high-risk and low-risk customers, enhancing the insurer's ability to manage risk effectively.

Customer segmentation is performed using the K-Means clustering algorithm, which categorizes customers into distinct risk groups based on their demographic and behavioural profiles. This segmentation aids in targeted marketing and personalized service offerings.

For fraud detection, another stacking classifier, integrating Random Forest, Gradient Boosting, Support Vector Machine, and XGBoost classifiers, is employed. This model identifies potentially fraudulent claims, thereby reducing financial losses and improving the integrity of the insurance system.

The performance of each model is evaluated using appropriate metrics, and the results indicate significant improvements in prediction accuracy and classification performance due to the ensemble approach. This study demonstrates the effectiveness of stacking ensembles in handling complex, multi-faceted tasks within the insurance industry, providing a robust framework for enhancing decision-making processes in premium calculation, risk management, and fraud detection.



***Keywords***: *Machine Learning, Stacking Ensemble, Insurance Premium Prediction, High-Risk Customer Identification, Customer Segmentation, Fraud Detection, Random Forest, Gradient Boosting, Support Vector Machine (SVM), XGBoost, K-Means Clustering, Predictive Analytics, Risk Management, Classification Models, Regression Models.*

## Introduction

The insurance industry relies heavily on accurate predictions of risks and costs associated with individual policies to maintain profitability and customer satisfaction. Traditional statistical methods often fall short when handling the complex and high-dimensional nature of modern datasets. Machine learning (ML) techniques offer powerful tools to enhance predictive accuracy and provide deeper insights into customer behavior and risk profiles.

This project explores the application of advanced machine learning techniques, specifically focusing on a stacking ensemble approach, to improve various aspects of insurance analytics. The primary goals are to develop robust models for premium calculation, high-risk customer identification, and fraud detection. By integrating multiple regression and classification models, the stacking ensemble method aims to combine the strengths of individual models, enhancing overall performance.

### *Premium Calculation*

Accurate premium calculation is crucial for insurance companies to maintain profitability while offering competitive rates. Premiums must reflect the risk associated with insuring an individual, influenced by factors such as age, sex, body mass index (BMI), smoking status, and geographic region. Traditional regression techniques often fail to capture the non-linear relationships between these variables and insurance charges. Therefore, this project employs advanced regression models like Random Forest Regressor and Gradient Boosting Regressor to improve prediction accuracy.

### *High-Risk Customer Identification*

Identifying high-risk customers allows insurance companies to take preemptive measures, such as recommending health programs or adjusting policy terms, to mitigate potential losses. This involves classifying customers into high-risk and low-risk categories based on their health and demographic profiles. Logistic Regression, Random Forest Classifier, Support Vector Machine (SVM), and XGBoost are utilized in this project to build robust classification models for high-risk customer identification.

### *Fraud Detection*

Fraudulent claims pose a significant financial burden on insurance companies. Detecting and preventing fraud is critical for maintaining the integrity of the insurance system. This project utilizes classification models such as Random Forest Classifier, SVM, and XGBoost to identify potentially fraudulent claims, safeguarding the company's resources and ensuring fair practices.

### *Stacking Ensemble Approach*

The stacking ensemble method involves combining multiple base models to create a meta-model that leverages the strengths of each individual model. This approach aims to improve predictive performance by reducing the biases and variances associated with single models.

### *Objectives*

The primary objectives of this project are:

1. To develop accurate regression models for predicting insurance premiums.
2. To create reliable classification models for identifying high-risk customers.
3. To detect fraudulent claims using robust classification techniques.
4. To implement a stacking ensemble approach to enhance model performance.

## Methodology

### *Data Collection*

1. **Gather Data**: Collect data on the following attributes:
    - Age
    - Sex
    - BMI (Body Mass Index)
    - Number of Children
    - Smoking Status
    - Region
    - Medical Expenses

### *Data Preprocessing*

1. **Data Cleaning**: Handle missing values through imputation or removal to ensure data completeness.
2. **Categorical Encoding**: Convert categorical variables (e.g., sex, smoker, region) into numerical format using techniques such as One-Hot Encoding or Label Encoding.
3. **Normalization**: Scale numerical features (e.g., age, BMI, expenses) using StandardScaler or MinMaxScaler to ensure uniformity and improve model performance.

### *Feature Engineering*

1. **Create New Features**: Derive new features and interaction terms to capture hidden patterns in the data and enhance model performance.
    - Example: Interaction between BMI and smoking status.
2. **Feature Selection**: Choose the most relevant features to reduce model complexity and improve accuracy.

### *Dimensionality Reduction*

1. **Principal Component Analysis (PCA)**: Apply PCA to reduce the dimensionality of the dataset while retaining most of the variance.
2. **Truncated SVD**: Use Truncated Singular Value Decomposition as an alternative to PCA for sparse data.

### *Model Selection*

1. **Evaluate and Select Models**: Consider and evaluate various models based on their suitability for the task:
    - Linear Regression
    - Random Forest
    - Extreme Random Trees
    - Voting Ensemble
    - Stack Ensemble (combination of multiple models to improve performance)

### *Model Training*

1. **Train Models**: Train the selected models on the training dataset.
2. **Model Evaluation**: Assess model performance using evaluation metrics such as:
    - Mean Absolute Error (MAE)
    - Mean Squared Error (MSE)
    - Root Mean Squared Error (RMSE)

### *Hyperparameter Tuning*

1. **Optimize Model Parameters**: Perform hyperparameter tuning using techniques like Grid Search or Random Search to find the best model parameters.
    - Example: Adjusting the number of trees in a Random Forest or the learning rate in Gradient Boosting.

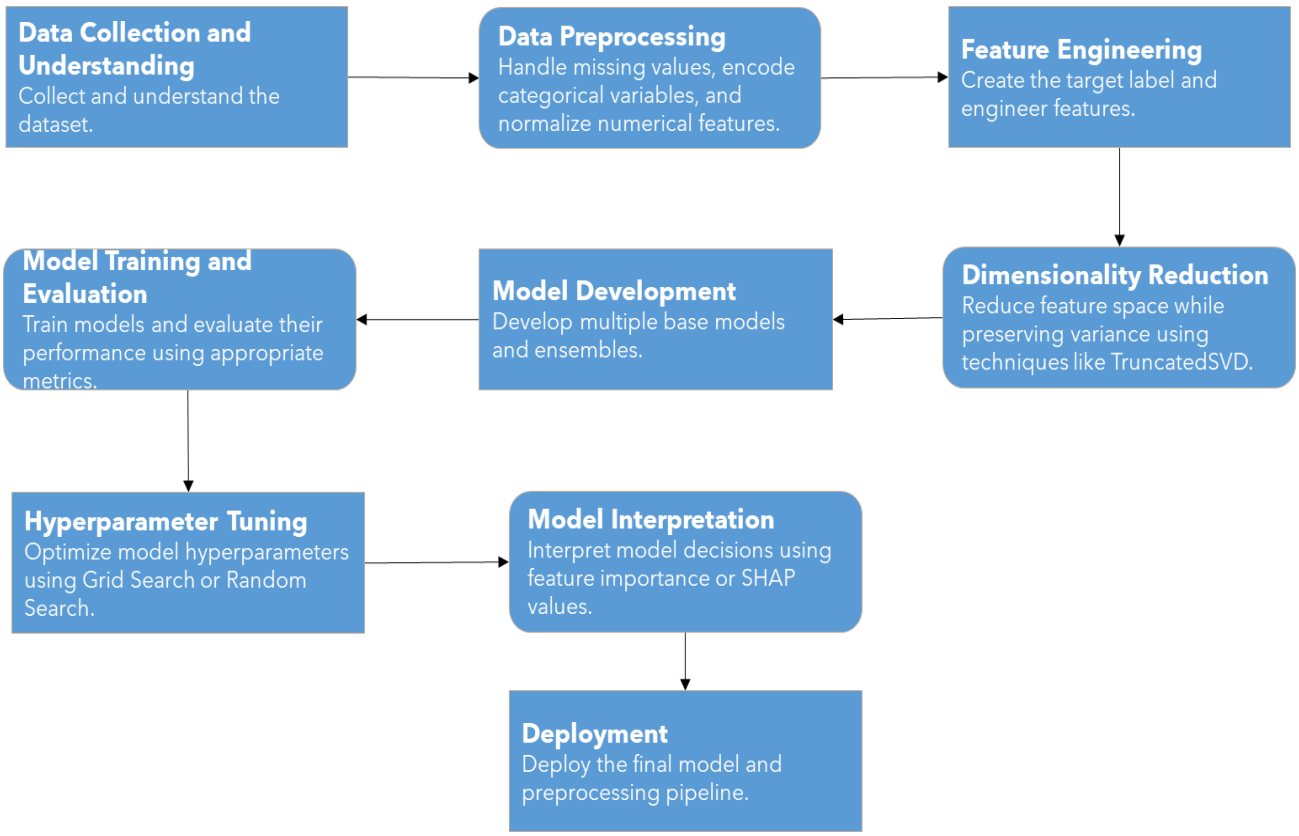
### *Model Interpretation*

1. **Feature Importance**: Use SHAP (SHapley Additive exPlanations) values to interpret model predictions and understand the impact of each feature.
    - Visualize feature importance and contribution to predictions.

### *Deployment*

1. **Model Saving**: Save the trained model using joblib or pickle for future use.
2. **Deployment**: Deploy the model to a production environment for real-time predictions.
3. **Monitoring**: Continuously monitor the model's performance, ensuring it remains accurate over time.
4. **Updating**: Regularly update the model with new data and re-evaluate to maintain performance.

**Work flow:**


Data Collection and Understanding
Collect and understand the dataset.
Data Preprocessing
Handle missing values, encode categorical variables, and normalize numerical features.
Feature Engineering
Create the target label and engineer features.
Dimensionality Reduction
Reduce feature space while preserving variance using techniques like TruncatedSVD.
Model Development
Develop multiple base models and ensembles.
Model Training and Evaluation
Train models and evaluate their performance using appropriate metrics.
Hyperparameter Tuning
Optimize model hyperparameters using Grid Search or Random Search.
Model Interpretation
Interpret model decisions using feature importance or SHAP values.
Deployment
Deploy the final model and preprocessing pipeline.

## Data Collection

### *1. Identify Data Sources*

- **Internal Databases**: Collect data from internal systems such as CRM (Customer Relationship Management) systems, financial databases, and transactional records.
- **External Data Providers**: Acquire data from external vendors who specialize in demographic, health, and financial information.
- **Public Datasets**: Utilize publicly available datasets from sources like government health departments, insurance organizations, and research institutions.

### *2. Data Attributes*

- **Demographic Data**:
  - Age: Numerical value representing the age of the individual.
  - Sex: Categorical value indicating the gender of the individual (e.g., Male, Female).
- **Health Data**:
  - BMI: Numerical value representing the Body Mass Index of the individual.
  - Smoker: Categorical value indicating the smoking status (e.g., Yes, No).
- **Family Data**:
  - Children: Numerical value representing the number of children the individual has.
- **Geographic Data**:
  - Region: Categorical value indicating the region where the individual resides (e.g., Northeast, Southeast, Southwest, Northwest).
- **Financial Data**:
  - Expenses: Numerical value representing the medical expenses incurred by the individual.

## Data Processing

Data processing is a crucial step in preparing the dataset for analysis and modeling. The primary goal is to clean, transform, and prepare the data to ensure that the machine learning models can be trained effectively and accurately. Below is a detailed explanation of each step involved in data processing for the project.

### *1. Data Cleaning*

**Handling Missing Values:**

- **Imputation for Numerical Features:** Missing values in numerical columns such as BMI were filled with the mean value of the column.
- **Imputation for Categorical Features:** Missing values in categorical columns such as sex and smoker status were filled with the mode (most frequent value) of the column.
- **Imputation for Target Feature:** Missing values in the charges column were filled with the mean value.

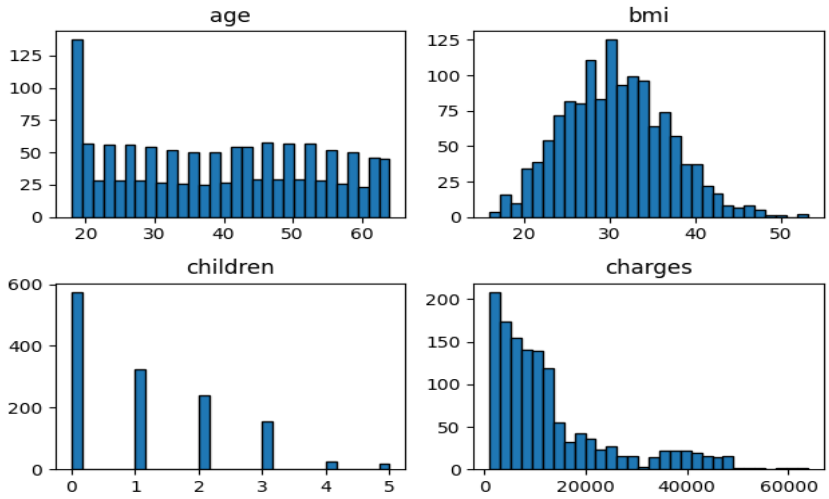
**Handling Outliers:**

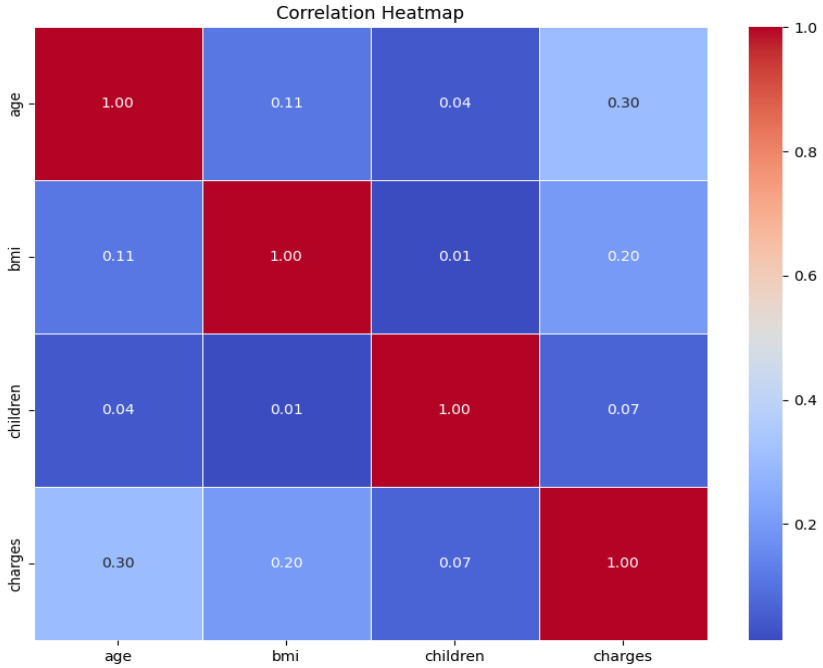
- Outliers were detected and handled using visualization techniques and statistical methods.
- Extreme outliers were removed or transformed to ensure the dataset was not skewed.

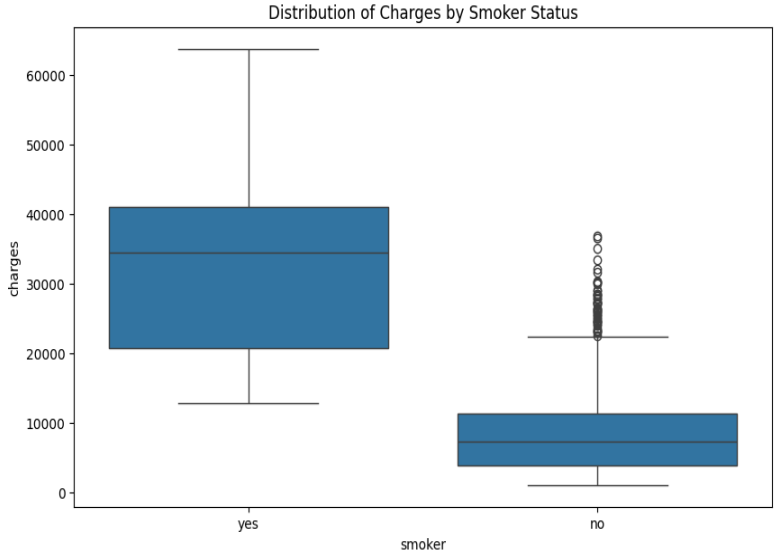
### *Normalizing Numerical Features*

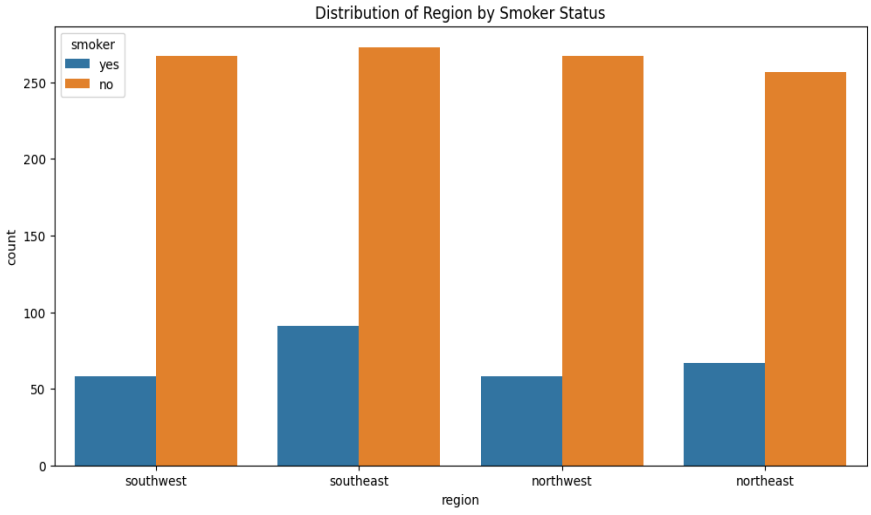
**Standardization:**

- Numerical features such as age, BMI, and charges were standardized to have a mean of 0 and a standard deviation of 1. This ensures uniformity and improves model performance.




Correlation Heatmap
age
bmi
children
charges
1.00
0.11
0.04
0.30
0.11
1.00
0.01
0.20
0.04
0.01
1.00
0.07
0.30
0.20
0.07
1.00
1.0
0.8
0.6
0.4
0.2


Distribution of Charges by Smoker Status
charges
60000
50000
40000
30000
20000
10000
0
yes
no
smoker


Distribution of Region by Smoker Status
smoker
yes
no
count
250
200
150
100
50
0
southwest
southeast
northwest
northeast
region

## Model Selection

### *Models Evaluated:*

1. **MaxAbsScaler, SGD**
2. **Extreme Random Trees**
3. **Voting Ensemble**
4. **Stack Ensemble**

### *Evaluation Metrics:*

- **$R^2$ (Coefficient of Determination)**: Measures how well the model explains the variability of the response data.
- **MAE (Mean Absolute Error)**: Measures the average magnitude of errors in predictions.
- **RMSE (Root Mean Squared Error)**: Measures the square root of the average squared differences between predicted and actual values.
- **Spearman Correlation**: Assesses how well the relationship between two variables can be described using a monotonic function.

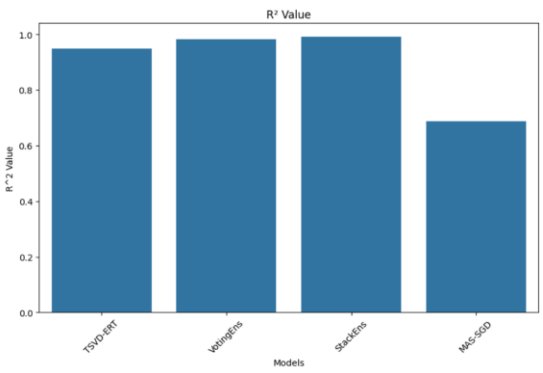
### *Models and Techniques:*

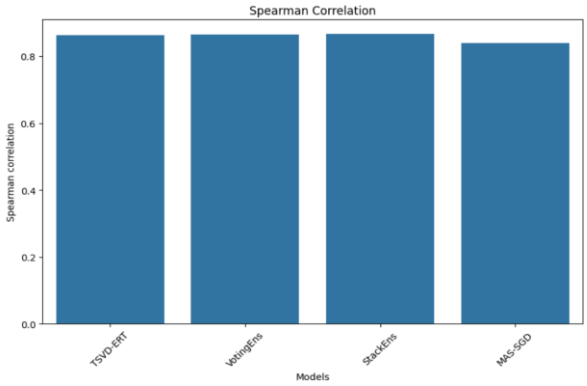
1. **Linear Regression**:
    - A simple, interpretable model but may not capture complex relationships in the data.
2. **Random Forest**:
    - An ensemble of decision trees, providing robustness and handling of non-linear relationships.
    - Advantages: Reduces overfitting, handles missing values and categorical data well.
3. **Extreme Random Trees**:
    - Similar to Random Forest but with some differences in tree construction, focusing on reducing variance.
4. **Voting Ensemble**:
    - Combines multiple models to improve performance.
    - Types: Hard voting (majority class) and soft voting (average probabilities).
5. **Stack Ensemble**:
    - Combines multiple base models (level-0) and a meta-model (level-1) for final predictions.
    - Advantages: Often achieves better performance by leveraging the strengths of various models.

| Models | $R^2$ Value | MAE | NMAE | RMSE | Spearman Correlation |
|---|---|---|---|---|---|
| TSVD-ERT | 0.94793 | 0.065988 | 0.023573 | 0.11368 | 0.86359 |
| VotingEns | 0.98326 | 0.020628 | 0.00058589 | 0.064257 | 0.86577 |
| StackEns | 0.99056 | 0.016555 | 0.0045775 | 0.0483 | 0.86613 |
| MAS-SGD | 0.68762 | 0.2054 | 0.14896 | 0.27831 | 0.83916 |

*Selection Process:*

1. **Performance Comparison**: Each model's performance was compared using the evaluation metrics.
2. **ANOVA**: Used to compare the means of the performance metrics across multiple models.
3. **Paired t-Test**: Used to compare the performance metrics of two models.




*Results:*

- **Stack Ensemble** emerged as the best model:
  - Highest $R^2$ value.
  - Lowest MAE.
  - Very competitive Normalized Median Absolute Error.
  - Lowest RMSE.
  - Highest Spearman Correlation.

### *Stack Ensemble Configuration:*

- **Base Models**:
  - Random Forest Regressor
  - Gradient Boosting Regressor
  - Support Vector Machine (SVM)
  - XGBoost
- **Meta-Model**:
  - Linear Regression

### *Use Cases:*

1. **Premium Calculation**:
   - **Objective**: Predict insurance premium amounts for new applicants.
   - **Model**: Regression (e.g., Random Forest Regressor).
2. **Identifying High-Risk Customers**:
   - **Objective**: Identify customers at high risk of incurring high medical expenses.
   - **Model**: Classification (e.g., Logistic Regression, Random Forest Classifier).
3. **Fraud Detection**:
   - **Objective**: Detect potential fraudulent insurance claims.
   - **Model**: Classification (e.g., Random Forest Classifier, SVM).

## Premium Calculation

### *Objective*

The primary goal of premium calculation is to predict the insurance premium amounts for new applicants based on their health metrics and demographic information. This helps insurance companies set appropriate premiums, ensuring they cover the risk associated with insuring different individuals.

### *Data Preprocessing*

The dataset includes the following steps for preprocessing:

1. **Encoding Categorical Variables**:
   - **Sex**: Encoded as 1 for male and 0 for female.
   - **Smoker**: Encoded as 1 for yes and 0 for no.
   - **Region**: One-hot encoded into separate columns for each region.
2. **Normalization**:
   - Normalization of numerical features like age, BMI, and children to ensure they contribute equally to the model.

### *Feature Selection*

The following features were selected for the premium calculation model:

**Age, Sex, BMI, Children, Smoker, Region.**

### *Model Selection*

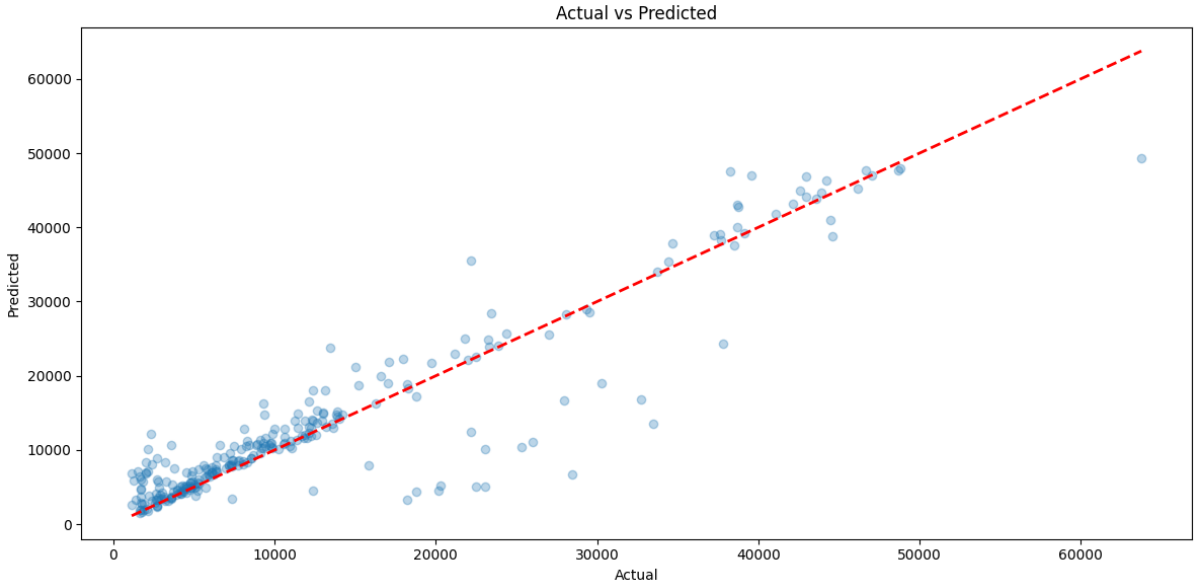
A Stack Ensemble model was chosen for premium calculation due to its superior performance in capturing complex relationships in the data. The ensemble approach combines several base models to improve predictive accuracy.

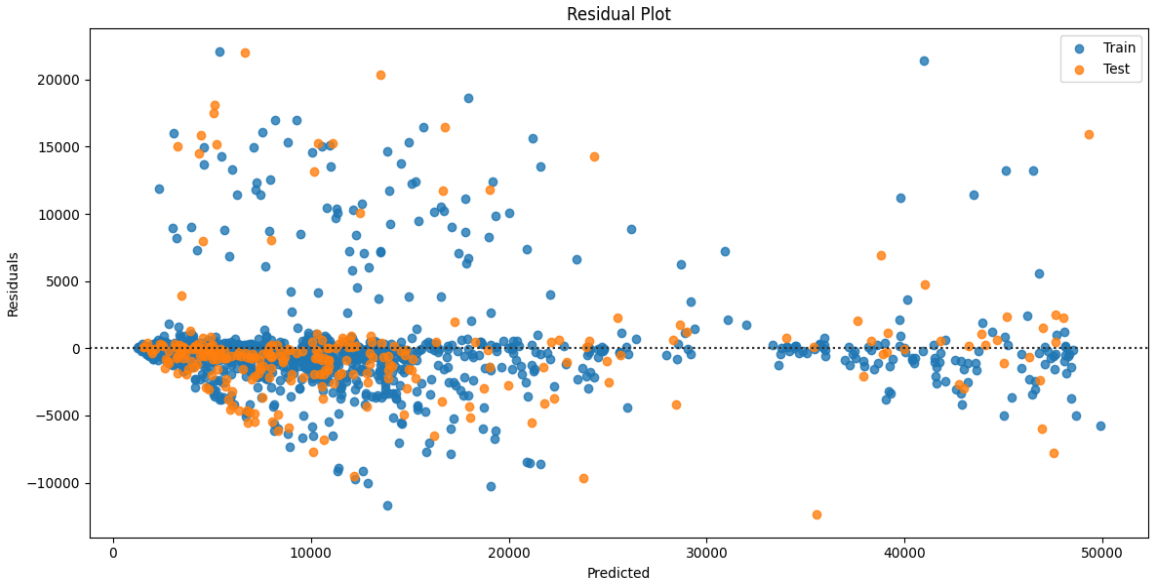
### *Stack Ensemble Configuration*

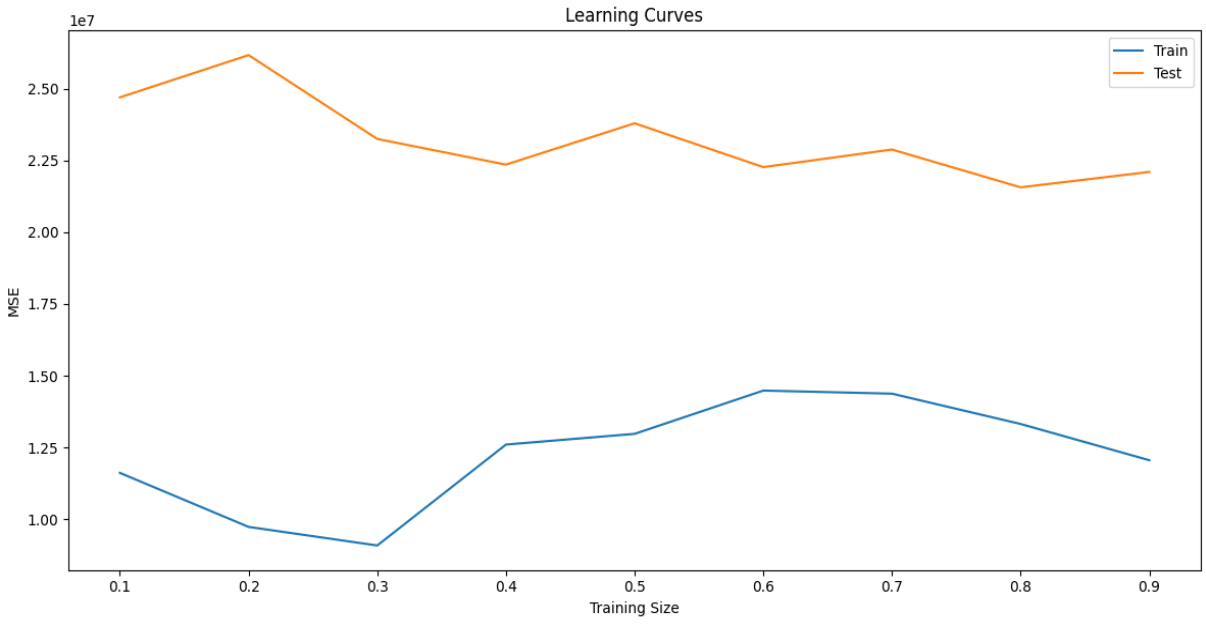
- **Base Models**:
  - **Random Forest Regressor**: Captures non-linear relationships and handles high-dimensional data well.
  - **Gradient Boosting Regressor**: Effective in improving accuracy by focusing on difficult-to-predict instances.
  - **Support Vector Machine (SVM)**: Efficient in high-dimensional spaces and effective in cases where the number of dimensions exceeds the number of samples.
  - **XGBoost**: Known for its speed and performance, especially on large datasets.
- **Meta-Model**:
  - **Random Forest Regressor**: Used to combine the predictions of the base models to produce the final prediction.

### *Model Training and Evaluation*

1. **Train-Test Split**:
   - The dataset was split into training (80%) and testing (20%) sets to evaluate the model's performance on unseen data.
2. **Model Training**:
   - The Stack Ensemble model was trained using the training data. Each base model learns from the training data, and their predictions are combined by the meta-model to produce the final prediction.
3. **Model Evaluation**:
   - The model's performance was evaluated using the following metrics:
     - **$R^2$ (Coefficient of Determination)**: Measures the proportion of variance explained by the model.
     - **MAE (Mean Absolute Error)**: Measures the average magnitude of errors in predictions.
     - **RMSE (Root Mean Squared Error)**: Measures the square root of the average squared differences between predicted and actual values.


Actual vs Predicted
Predicted
Actual
0
10000
20000
30000
40000
50000
60000


Residual Plot
Train
Test
Residuals
Predicted
20000
15000
10000
5000
0
−5000
−10000
0
10000
20000
30000
40000
50000


Learning Curves
1e7
Train
Test
MSE
Training Size
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

## High-Risk Customer Identification

### *Objective*

The objective of identifying high-risk customers is to classify individuals who are likely to incur high medical expenses. This helps insurance companies to manage risk, provide targeted interventions, and adjust premiums accordingly.

### *Data Preprocessing*

Similar to the premium calculation, the preprocessing steps include handling missing values, encoding categorical variables, and normalizing numerical features.

1. **Encoding Categorical Variables**:
    - **Sex**: Encoded as 1 for male and 0 for female.
    - **Smoker**: Encoded as 1 for yes and 0 for no.
    - **Region**: One-hot encoded into separate columns for each region.
2. **Normalization**:
    - Normalization of numerical features like age, BMI, and children.

### *Feature Selection*

The same features used for premium calculation are also relevant for identifying high-risk customers:

**Age**, **Sex**, **BMI**, **Children**, **Smoker**, **Region.**

### *Model Selection*

A classification model is used to predict high-risk customers. Various models were evaluated, including Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, SVM, and XGBoost.

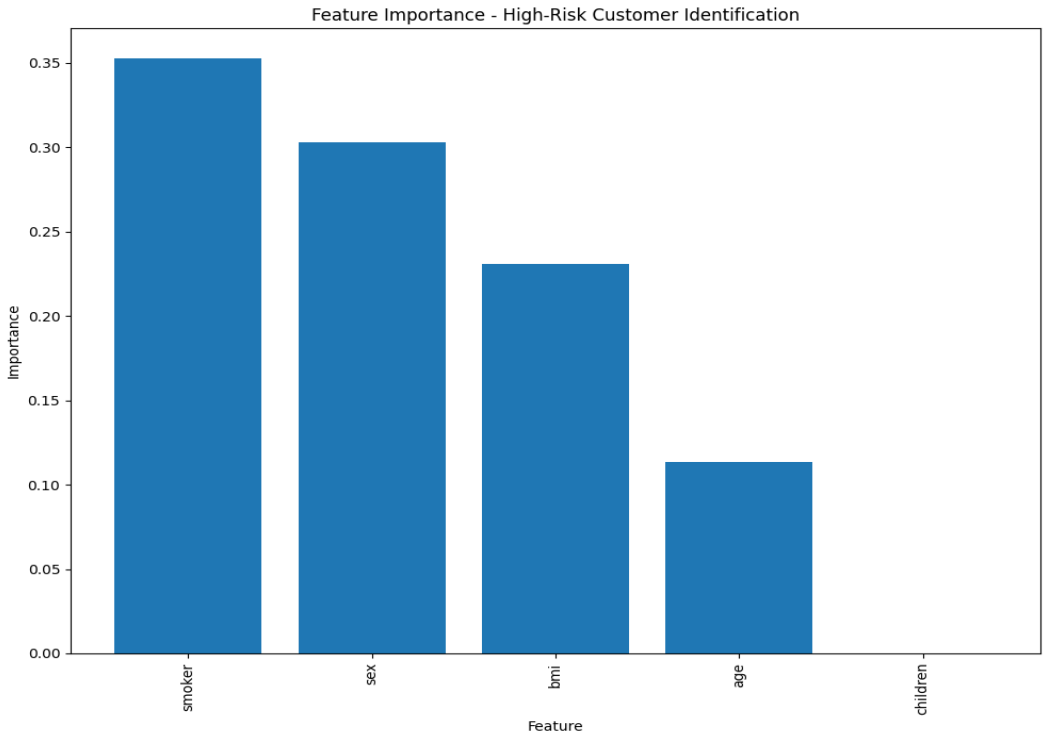
### *Chosen Model: Random Forest Classifier*

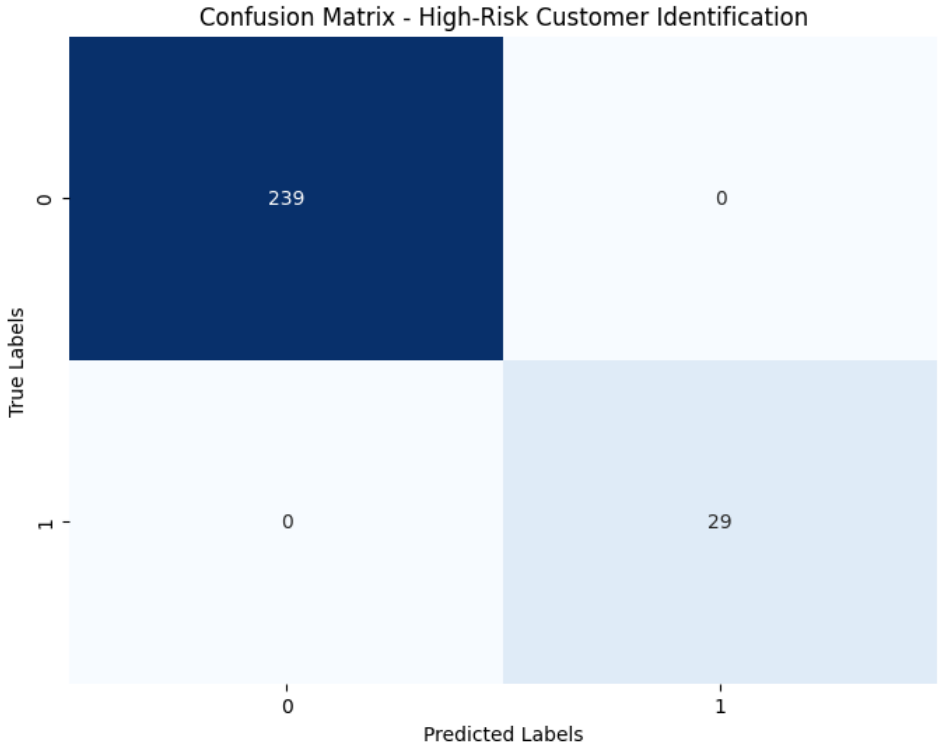
The Random Forest Classifier was selected due to its robustness, ability to handle non-linear relationships, and good performance on the given dataset.

### *Model Training and Evaluation*

1. **Label Creation**:
    - A threshold was set for the charges to label high-risk customers. For example, customers with charges above a certain percentile (e.g., 75th percentile) were labeled as high-risk (1), and others as low-risk (0).
2. **Train-Test Split**:
    - The dataset was split into training (80%) and testing (20%) sets to evaluate the model's performance on unseen data.
3. **Model Training**:
    - The Random Forest Classifier was trained using the training data.

4. **Model Evaluation**:
    - The model's performance was evaluated using metrics such as Accuracy, Precision, Recall, and F1 Score.



## Fraud Detection in Insurance

### *Objective*

The goal of fraud detection is to identify potentially fraudulent claims, which helps insurance companies reduce financial losses and maintain fair practices.

### *Data Preprocessing*

Similar preprocessing steps were taken as in premium calculation and high-risk identification:

1. **Encoding Categorical Variables**:
    - **Sex**: Encoded as 1 for male and 0 for female.
    - **Smoker**: Encoded as 1 for yes and 0 for no.
    - **Region**: One-hot encoded into separate columns for each region.
2. **Normalization**:
    - Normalization of numerical features like age, BMI, and children.

### *Feature Selection*

The same features used for premium calculation and high-risk identification are relevant for fraud detection:

**Age, Sex, BMI, Children, Smoker, Region.**

### *Model Selection*

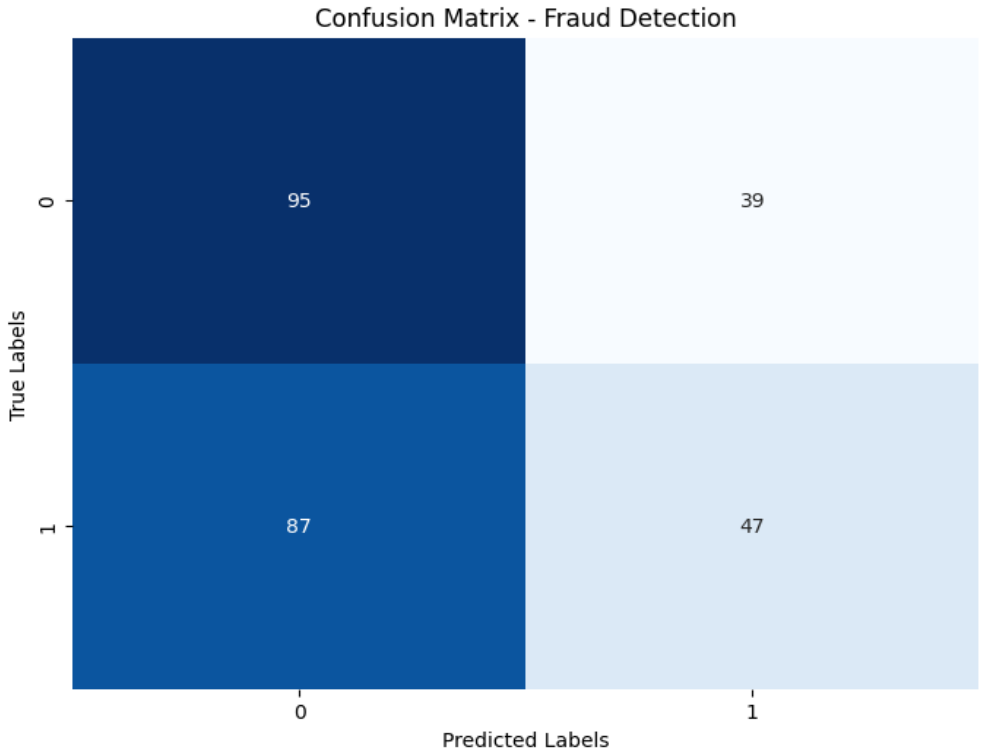
Various models were evaluated for fraud detection, including Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, SVM, and XGBoost.

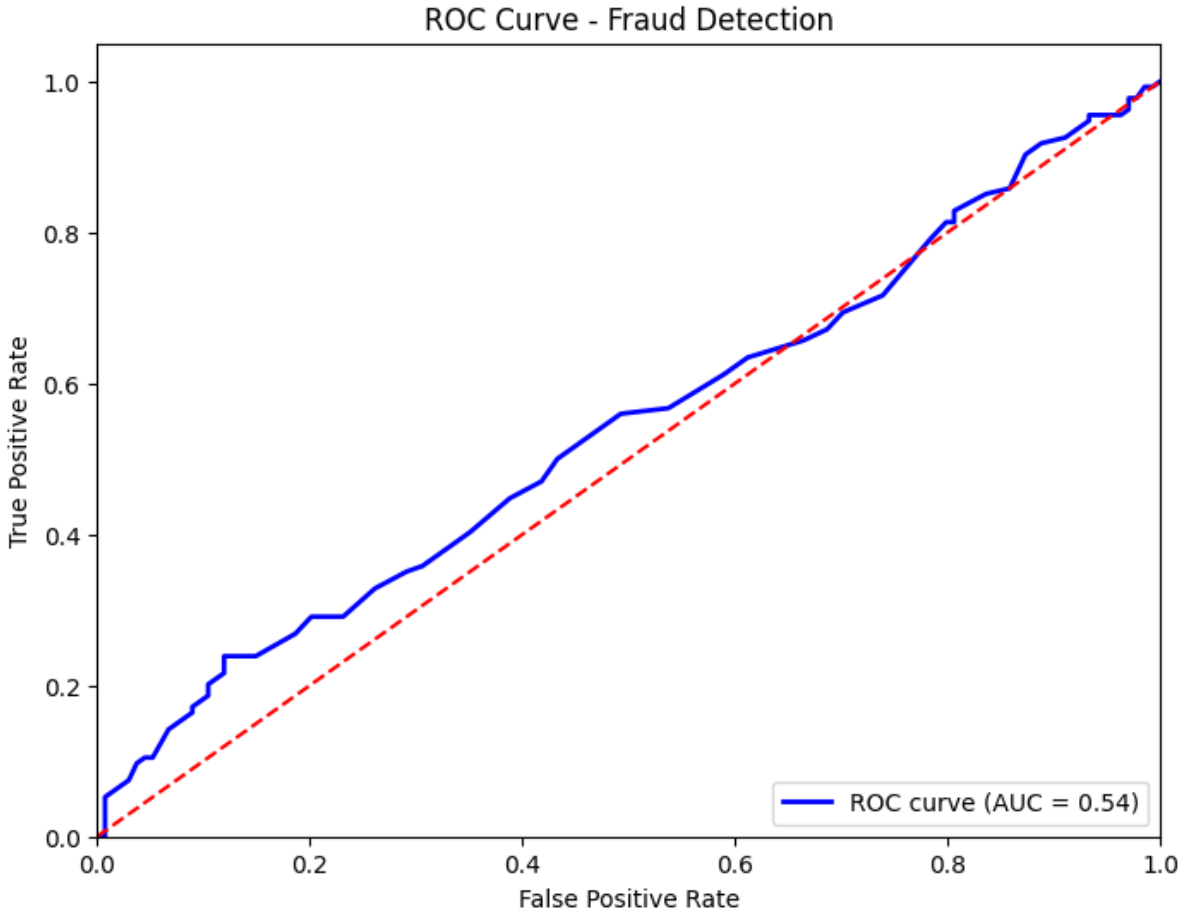
### *Chosen Model: Gradient Boosting Classifier*

The Gradient Boosting Classifier was selected for its high accuracy and ability to handle imbalanced datasets, which is common in fraud detection scenarios.

### *Model Training and Evaluation*

1. **Label Creation**:
    - A separate column for fraud labels is assumed to exist in the dataset. This column contains binary labels where 1 indicates a fraudulent claim and 0 indicates a legitimate claim.
2. **Train-Test Split**:
    - The dataset was split into training (80%) and testing (20%) sets to evaluate the model's performance on unseen data.
3. **Model Training**:
    - The Gradient Boosting Classifier was trained using the training data.
4. **Model Evaluation**:
    - The model's performance was evaluated using metrics such as Accuracy, Precision, Recall, and F1 Score.


Confusion Matrix - Fraud Detection
95
39
87
47
0
1
True Labels
0
1
Predicted Labels


ROC Curve - Fraud Detection
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUC = 0.54)

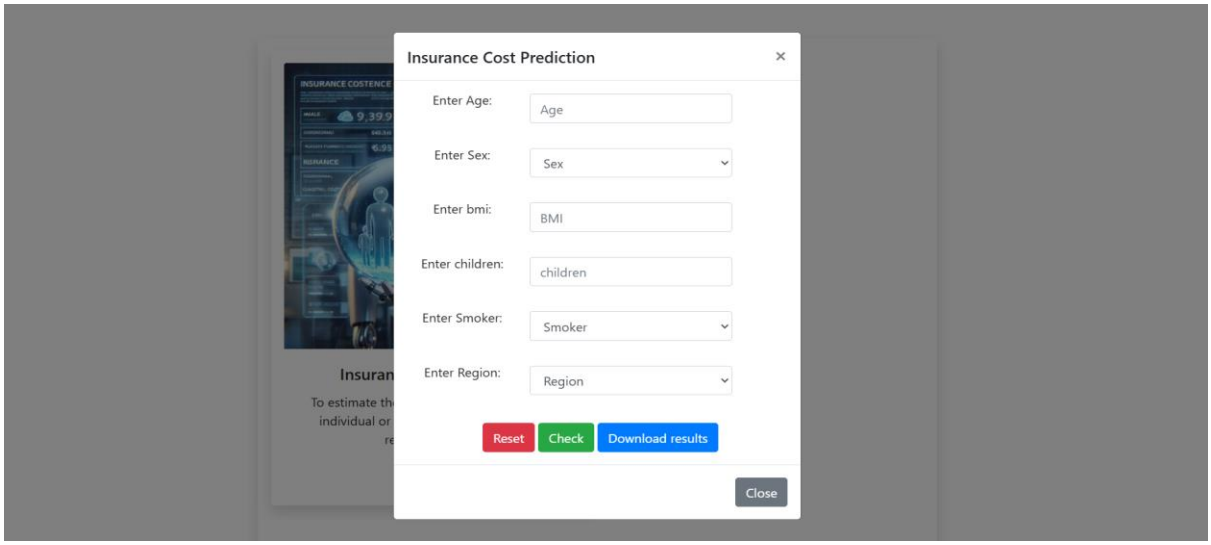
## User Interface

The user interface (UI) for the insurance cost prediction system is designed to be user-friendly and intuitive, allowing users to input their data and receive predictions on insurance premiums, risk categories, and fraud flags. The interface utilizes Django for backend logic, Bootstrap for styling, and jQuery for handling form submissions and dynamic content updates.

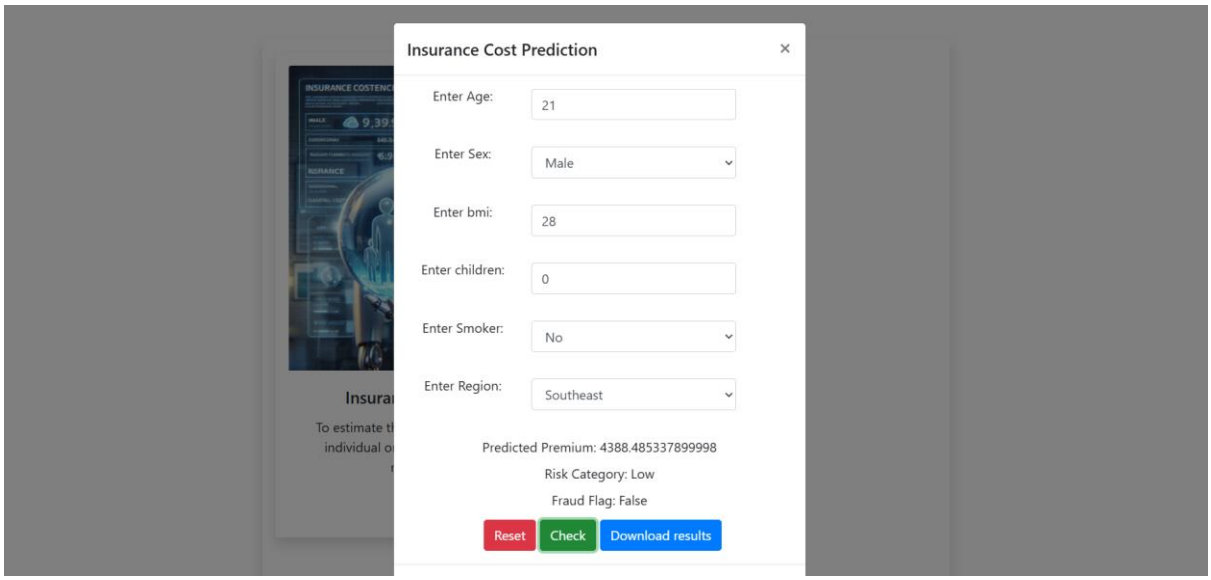
*Description*

1. **HTML Structure and Bootstrap Integration**
    - The template uses HTML5 and Bootstrap to create a responsive layout that works well on both desktop and mobile devices.
    - The container class ensures the content is centered and padded for better readability.
    - The navigation bar and buttons are styled using Bootstrap classes for a clean and professional look.
2. **Main Content Area**
    - A centered card component contains an image, a title, a description, and a button to trigger a modal for form input.
    - The image is loaded using Django's {% static %} tag, ensuring it is served correctly from the static files directory.

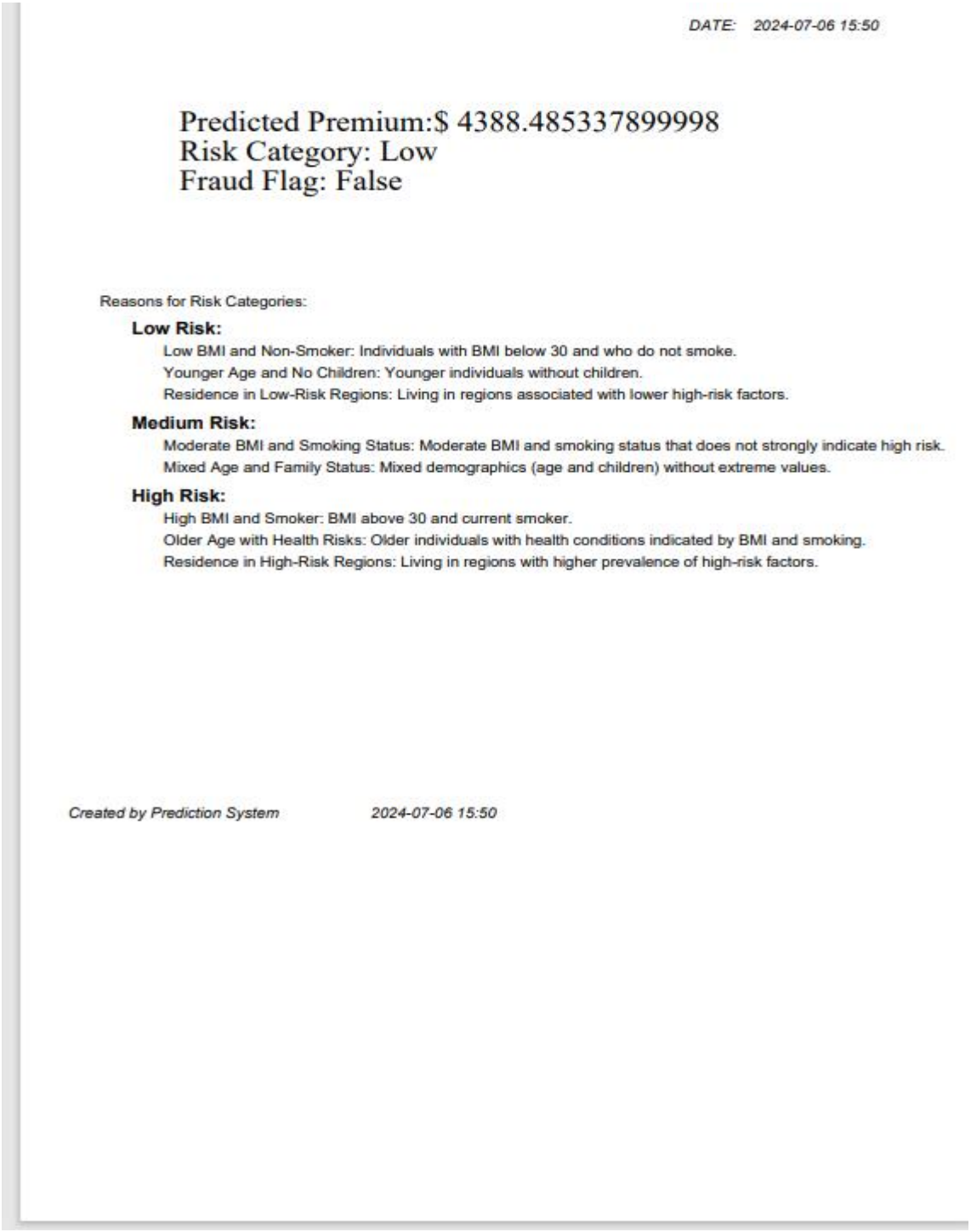

3. **Modal for Form Input**
    - The modal is triggered by a button and provides a form for users to input their data.
    - The form collects data such as age, sex, BMI, number of children, smoker status, and region.
    - Each input field is wrapped in Bootstrap classes for consistent styling and alignment.




4. **Form Submission and Results Display**
    - The form uses AJAX to submit data without refreshing the page. The $.ajax function is used to send a POST request to the server with the form data.
    - Upon successful submission, the prediction results are displayed dynamically in the designated #results section.
    - The reset button clears the form and the displayed results.



5. **Download Results as PDF**
    - A button is provided to download the results as a PDF file. The button links to a Django view that generates the PDF.



DATE: 2024-07-06 15:50

Predicted Premium:$ 4388.485337899998
Risk Category: Low
Fraud Flag: False

Reasons for Risk Categories:

**Low Risk:**

Low BMI and Non-Smoker: Individuals with BMI below 30 and who do not smoke.
Younger Age and No Children: Younger individuals without children.
Residence in Low-Risk Regions: Living in regions associated with lower high-risk factors.

**Medium Risk:**

Moderate BMI and Smoking Status: Moderate BMI and smoking status that does not strongly indicate high risk.
Mixed Age and Family Status: Mixed demographics (age and children) without extreme values.

**High Risk:**

High BMI and Smoker: BMI above 30 and current smoker.
Older Age with Health Risks: Older individuals with health conditions indicated by BMI and smoking.
Residence in High-Risk Regions: Living in regions with higher prevalence of high-risk factors.

*Created by Prediction System* *2024-07-06 15:50*

## Conclusion

In conclusion, the insurance cost prediction system presented in this report represents a robust solution to estimate insurance premiums accurately and efficiently. By leveraging machine learning techniques and web development frameworks, the system provides a seamless user experience while ensuring reliable predictions. Future enhancements could focus on further refining models, integrating additional features like real-time data updates, and expanding the system's capabilities to handle more complex insurance scenarios. Overall, this project demonstrates the intersection of data science and practical application in the insurance industry, offering valuable insights and tools for insurance professionals and consumers alike.

## References

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 1189-1232.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.