Health Insurance Premium Predictor

Machine Learning-Based Premium Estimation

Presented by Vaibhav Garg

PROBLEM STATEMENT

- Health insurance premiums vary significantly across individuals due to diverse factors like age, BMI, smoking status, and medical history, making cost prediction highly complex.
- Traditional methods struggle to accurately estimate premiums, especially for high-risk or non-linear cases.
- There is a need for a reliable, data-driven solution that can predict premiums fairly and accurately for a wide range of profiles.

PROJECT OBJECTIVES

- Develop a high-accuracy (>97%) predictive model to predict health insurance premium using ML.
- The percentage difference between the predicted and actual value on a minimum of 95% of the errors should be less than 10%.
- Deploy the model in the cloud so that an insurance underwriter can run it from anywhere.
- Create an interactive Streamlit application that an underwriter can use for predictions.

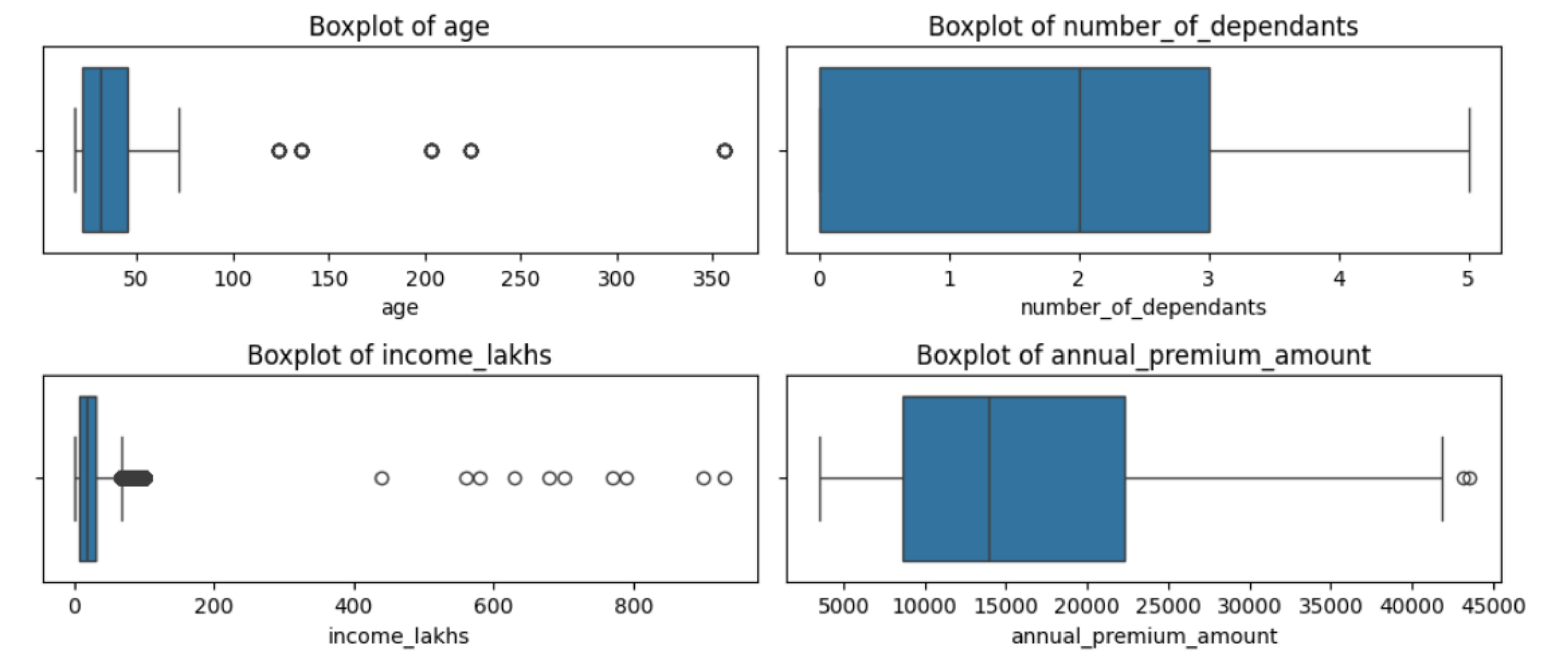
DATASET FEATURES (50000 RECORDS)

Feature Name	Description
age	Age of the individual
gender	Gender: Male / Female
region	Geographic location: Northwest / Southeast / Northeast / Southwest
marital_status	Marital status: Unmarried / Married
number_of_dependants	Count of dependents
bmi_category	BMI category: Underweight / Normal / Overweight / Obesity
smoking_status	Smoking habit: No Smoking / Regular / Occasional
employment_status	Employment type: Salaried / Freelancer / Self-Employed
income_level	Income group: <10L / 10L–25L / 25L–40L / >40L
income_lakhs	Income in lakhs (numerical value)
medical_history	Details of past medical conditions - Diabetes / High blood pressure / No Disease / Diabetes & High blood pressure / Thyroid / Heart disease / High blood pressure & Heart disease / Diabetes & Thyroid / Diabetes & Heart disease
insurance_plan	Type of plan: Bronze / Silver / Gold
annual_premium_amount	**Target variable**: Premium amount to be predicted

DATA PREPROCESSING

- Data Cleaning
  - Dropped NULL Values
  - Dropped duplicate rows.
  - Replaced negative number of dependents with absolute value.

- Numerical Features Analysis
  - Box plots were used to detect and visualize outliers.



- Removed records where age was greater than 100.
- Removed records where income values exceeded the 99th percentile.

- Categorical Feature Analysis
  - Checked unique values in each categorical column.

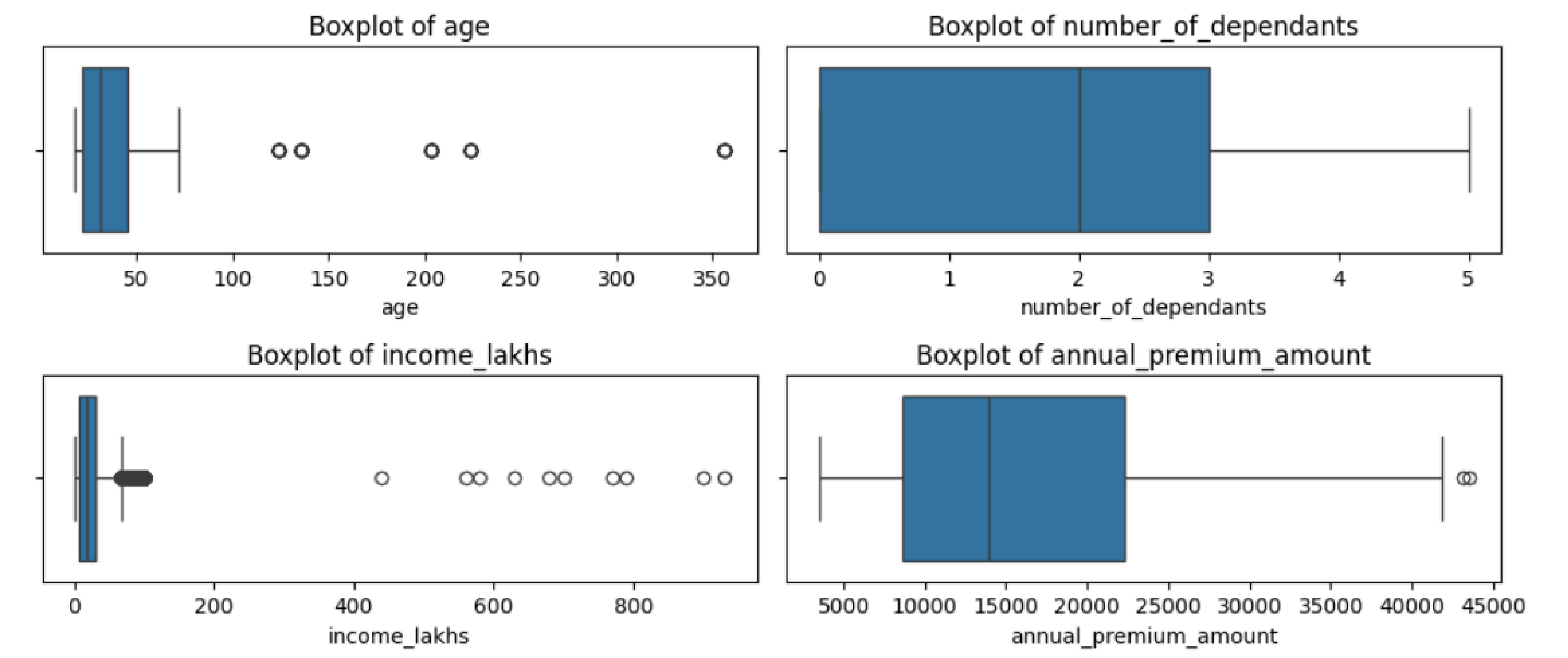
```
gender : ['Male' 'Female']
region : ['Northwest' 'Southeast' 'Northeast' 'Southwest']
marital_status : ['Unmarried' 'Married']
bmi_category : ['Normal' 'Obesity' 'Overweight' 'Underweight']
smoking_status : ['No Smoking' 'Regular' 'Occasional' 'Smoking=0' 'Does Not Smoke'
 'Not Smoking']
employment_status : ['Salaried' 'Self-Employed' 'Freelancer']
income_level : ['<10L' '10L - 25L' '> 40L' '25L - 40L']
medical_history : ['Diabetes' 'High blood pressure' 'No Disease'
 'Diabetes & High blood pressure' 'Thyroid' 'Heart disease'
 'High blood pressure & Heart disease' 'Diabetes & Thyroid'
 'Diabetes & Heart disease']
insurance_plan : ['Bronze' 'Silver' 'Gold']
```

  - Cleaned inconsistent entries in the smoking_status column to ensure uniformity.

DATA PREPROCESSING

- Data Cleaning
  - Dropped NULL Values
  - Dropped duplicate rows.
  - Replaced negative number of dependents with absolute value.

- Numerical Features Analysis
  - Box plots were used to detect and visualize outliers.



FEATURE ENGINEERING

- Created a Normalized Risk Score
  - Combined disease1 and disease2 from medical_history to assign risk scores.
  - Calculated the total risk score and normalized it to form the normalized_risk_score column.

- Encoded Ordinal Features using Label Encoding
  - insurance_plan: 'Bronze' → 1, 'Silver' → 2, 'Gold' → 3
  - income_level: '<10L' → 1, '10L - 25L' → 2, '25L - 40L' → 3, '> 40L' → 4

- Applied One-Hot Encoding to Nominal Features
  - Converted non-ordinal categorical features into binary columns using one-hot encoding.

- Dropped Redundant Columns
  - Removed original columns: medical_history, disease1, disease2, and total_risk_score after deriving new features.

- Scaled Numerical Features
  - Applied Min-Max scaling to bring numerical values to the range [0, 1].

- Checked Multicollinearity using Variance Inflation Factor (VIF)
  - Calculated VIF scores for all features.
  - Dropped features with VIF > 10, such as income_level and recalculated VIF.

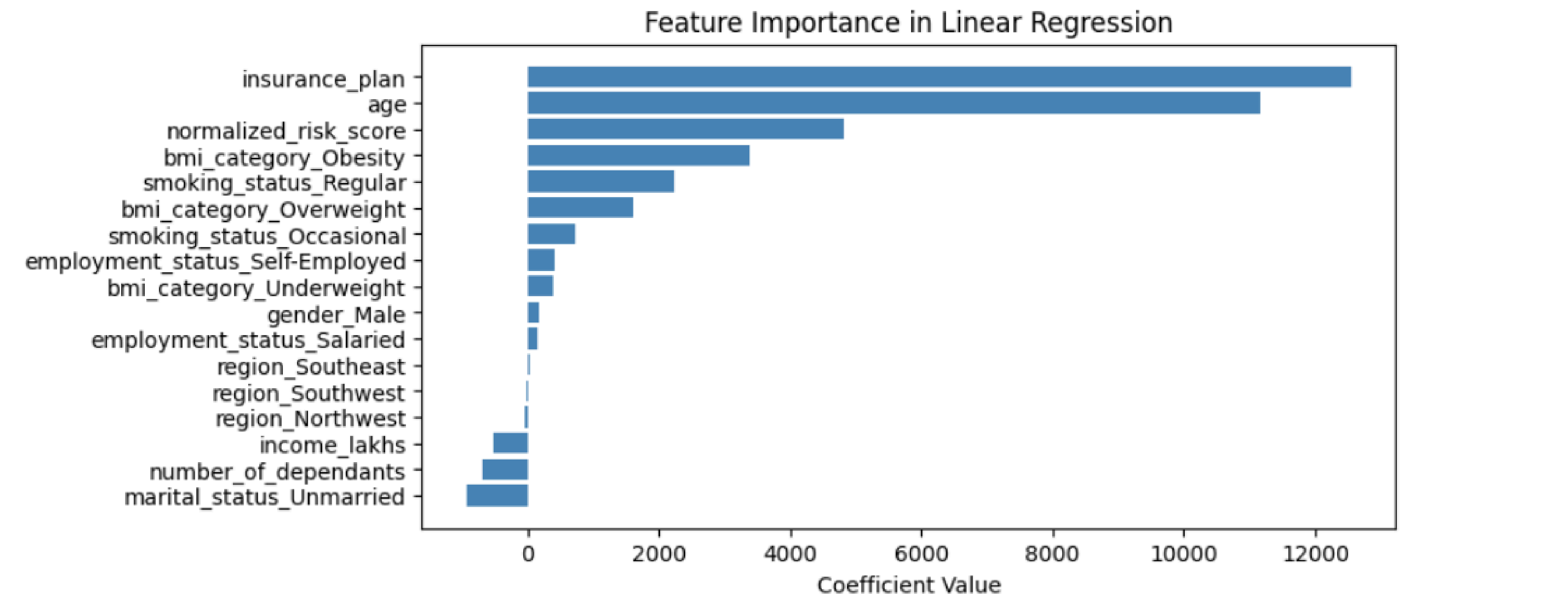
	Column	VIF
0	age	4.567634
1	number_of_dependants	4.534650
2	income_level	12.450675
3	income_lakhs	11.183367
4	insurance_plan	3.584752
5	normalized_risk_score	2.687610
6	gender_Male	2.421496
7	region_Northwest	2.102556
8	region_Southeast	2.922414
9	region_Southwest	2.670666
10	marital_status_Unmarried	3.411185
11	bmi_category_Obesity	1.352806
12	bmi_category_Overweight	1.549922
13	bmi_category_Underweight	1.302886
14	smoking_status_Occasional	1.272745
15	smoking_status_Regular	1.777089
16	employment_status_Salaried	2.382134
17	employment_status_Self-Employed	2.137753

	Column	VIF
0	age	4.545825
1	number_of_dependants	4.526598
2	income_lakhs	2.480563
3	insurance_plan	3.445682
4	normalized_risk_score	2.687326
5	gender_Male	2.409980
6	region_Northwest	2.100789
7	region_Southeast	2.919775
8	region_Southwest	2.668314
9	marital_status_Unmarried	3.393718
10	bmi_category_Obesity	1.352748
11	bmi_category_Overweight	1.549907
12	bmi_category_Underweight	1.302636
13	smoking_status_Occasional	1.272744
14	smoking_status_Regular	1.777024
15	employment_status_Salaried	2.374628
16	employment_status_Self-Employed	2.132810

MODEL TRAINING

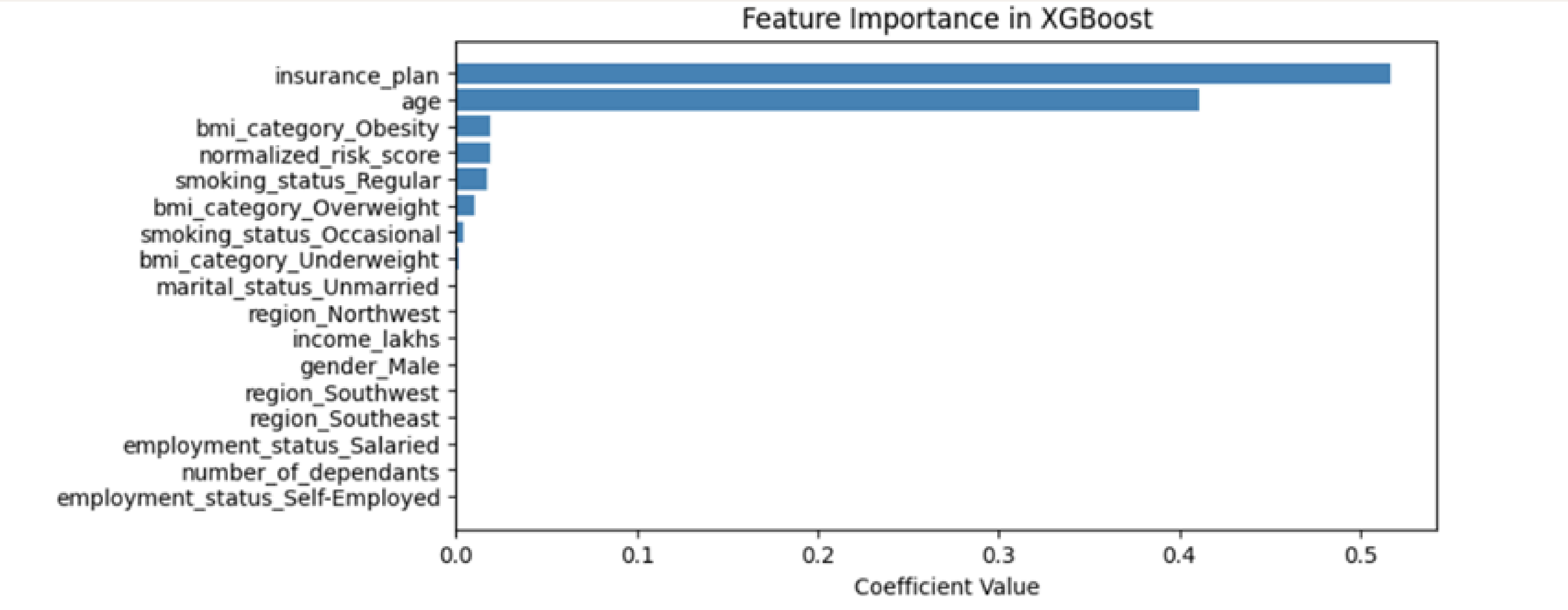
- Train-Test Split
  - Split the dataset into 70% training and 30% testing to evaluate generalization performance.

- Baseline Model – Linear Regression
  - Trained a simple Linear Regression model as a baseline.
  - Evaluated the performance of model using MSE and R^2.
    - Mean Squared Error (MSE) : 5165611.9
    - R^2 Score: 0.92805
  - Plotted feature importance from the trained model.



- Regularized Model – Ridge Regression
  - Trained a Ridge Regression model.
  - Evaluated the performance of model using MSE and R^2.
    - Mean Squared Error (MSE) : 5165652.02
    - R^2 Score: 0.92822

- Advanced Model – XGBoost Regressor
  - Trained an XGBoost model to capture non-linear patterns.
  - Used RandomSearchCV for optimization.
  - Evaluated the performance of model using MSE and R^2.
    - Mean Squared Error (MSE) : 1563064.14
    - R^2 Score: 0.98095
  - Plotted feature importance from the trained model.

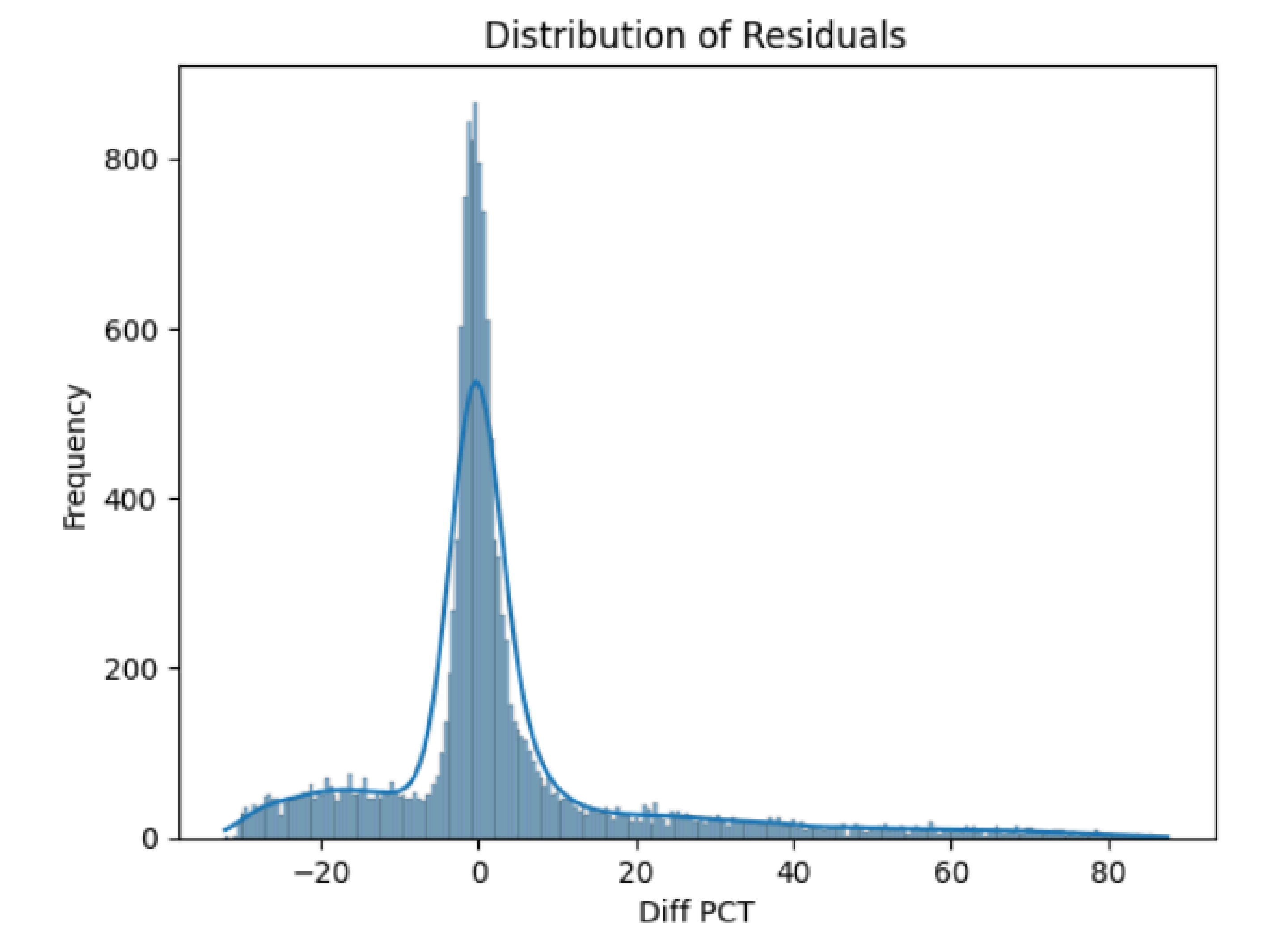


ERROR ANALYSIS

- Calculated residuals and their percentage error using the formula: (y_pred - y_test) / y_test × 100.

- Plotted histogram of residual percentage errors.

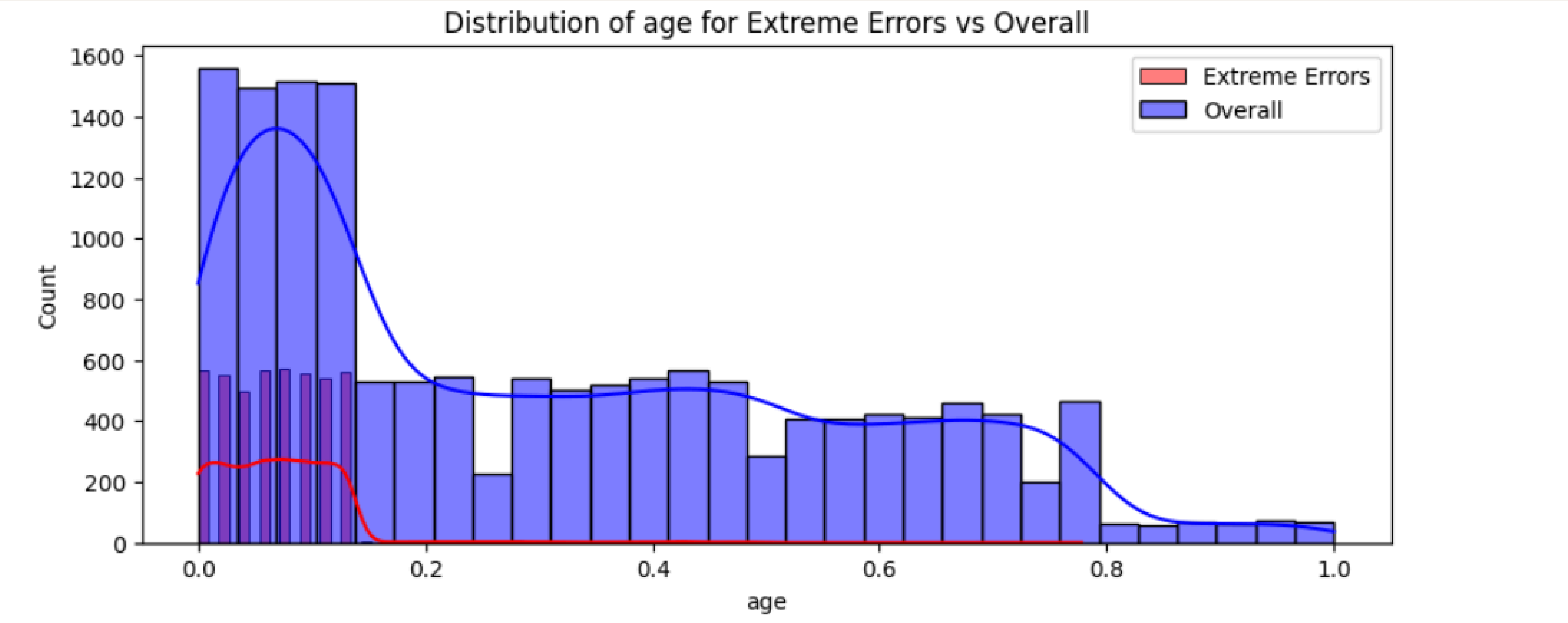
.



- Set a threshold of ±10% to identify extreme prediction errors.

- Found that 30% of customers were overcharged or undercharged by more than 10%

- Plotted KDE distributions of selected features to compare customers with extreme errors (|residual %| > 10%) against the overall population.

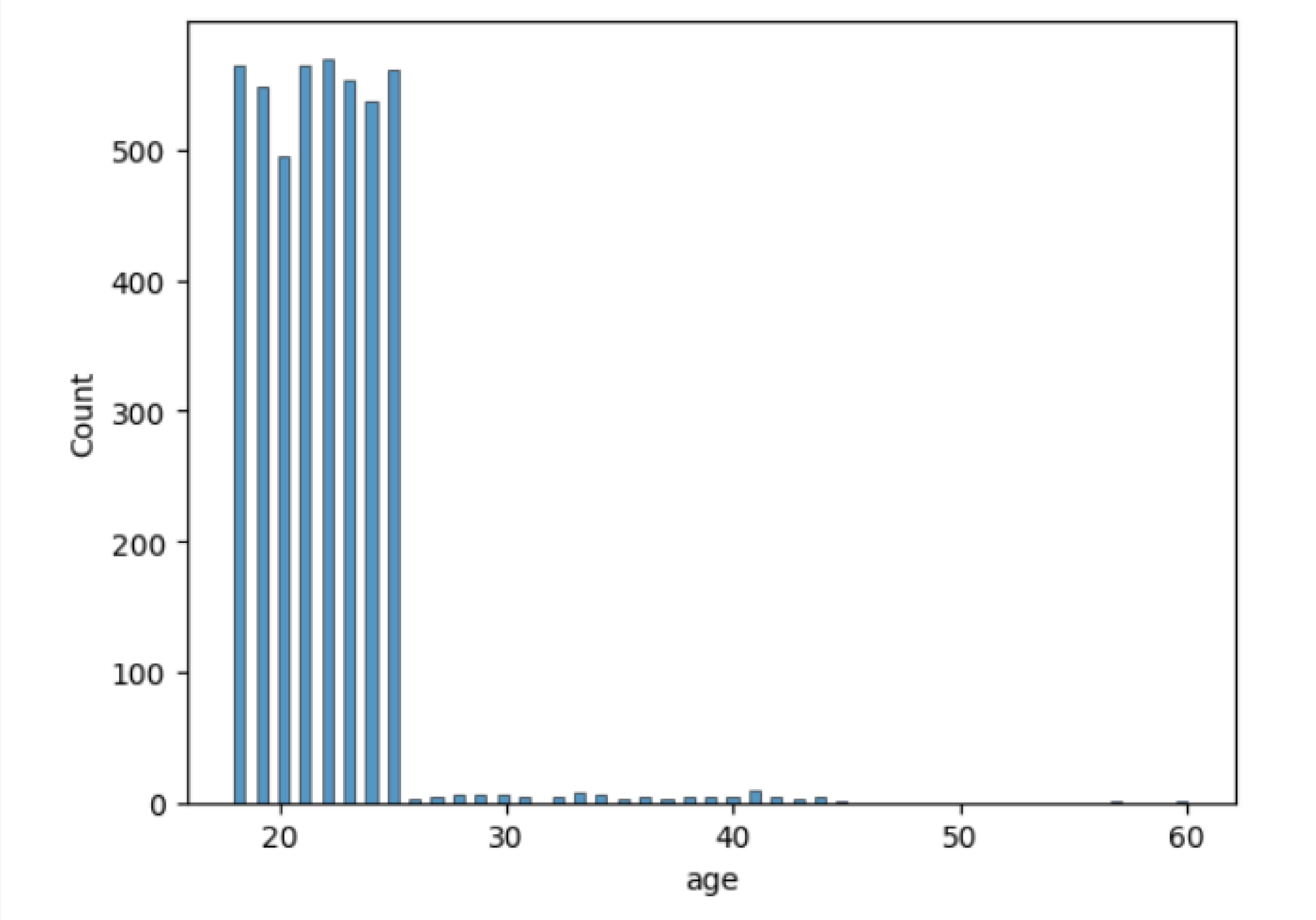


- Found a noticeable pattern:
  - Majority of the extreme errors are concentrated in the younger age group.

- Indicates that age may be a key driver of high prediction deviations.

- Reverse scaled the age feature to bring it back to its original range for interpretability.

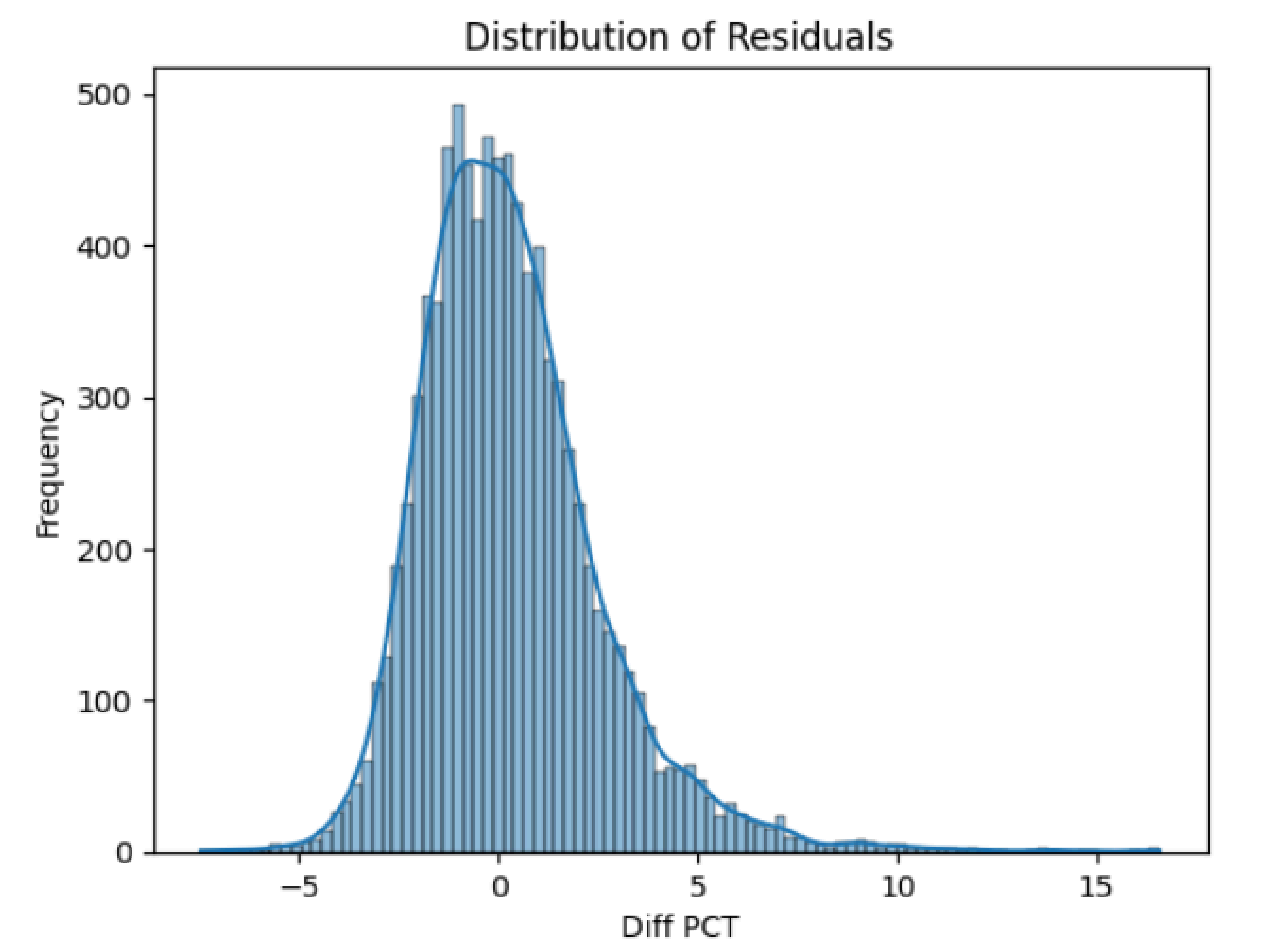
- Plotted a histogram of age values for customers with |residual %| > 10%..



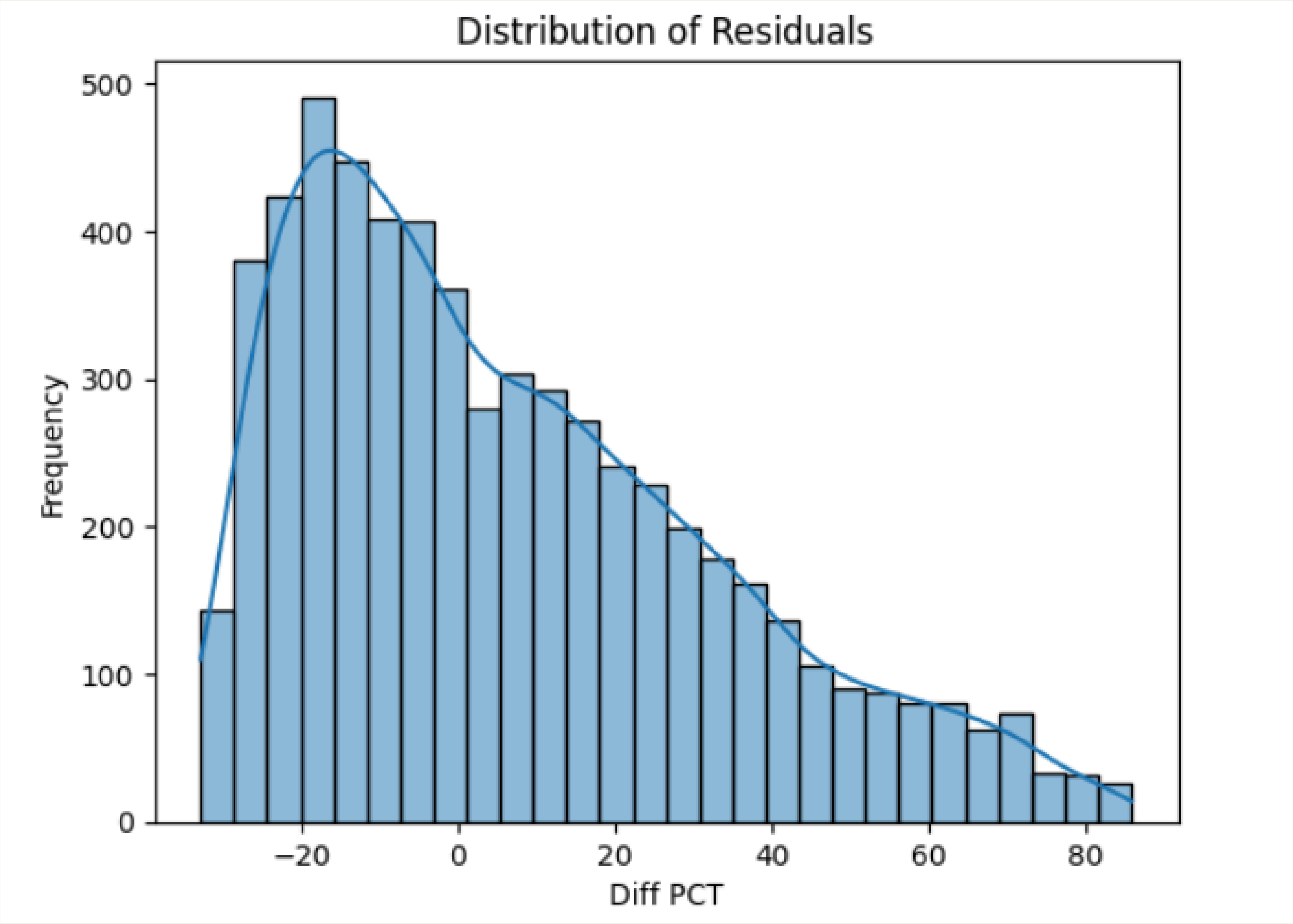
- Observed that a large portion of extreme errors occurred among customers in the younger age group.

MODEL SEGMENTATION

- Segment 1: Age > 25
  - Only 0.3% of customers in this group had extreme errors.
  - The model performs well for this segment.
  - No further investigation required.

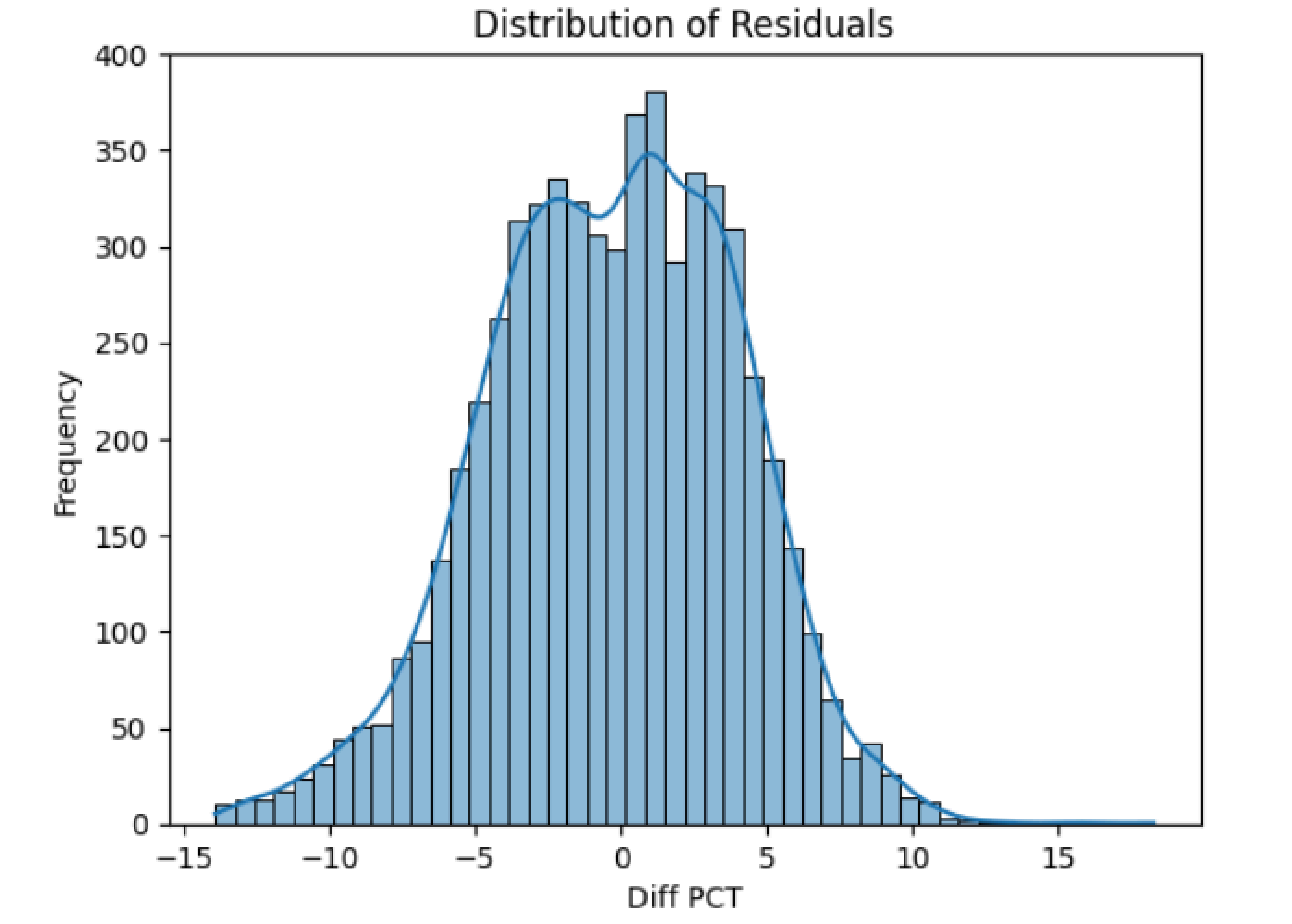


- Segment 2: Age < 25
  - Around 73% of customers experienced extreme errors.
  - Compared feature distributions but found no meaningful patterns.
  - Concluded that the model may be lacking important predictive features for this group.

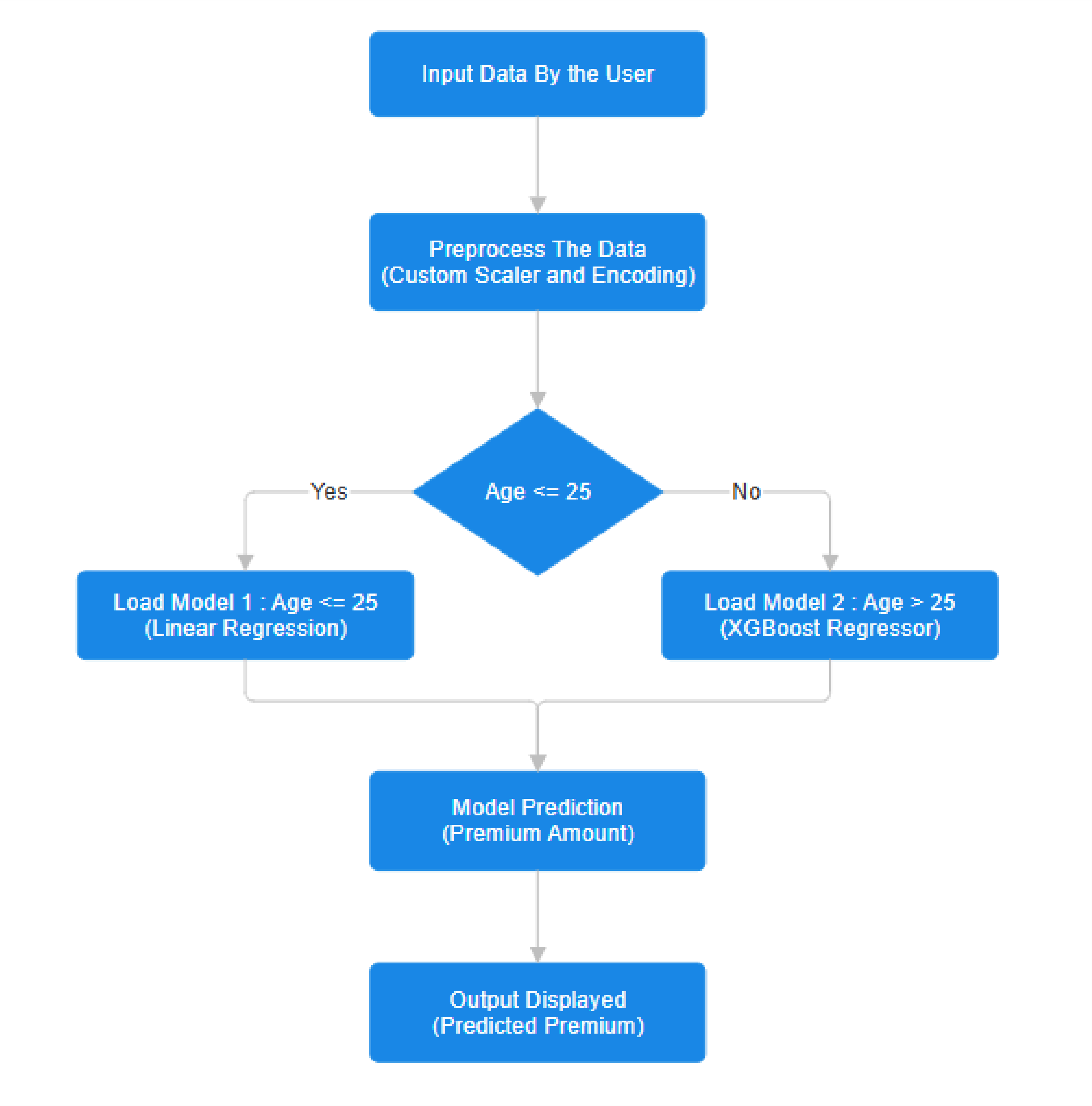


MODEL RETRAINING: ADDED GENETIC RISK FEATURE:

- Introduced a new feature: Genetic Risk.
- Since 73% of extreme errors were observed in the younger age group, models were retrained on this segment after introducing the genetic risk feature.
- Retrained all models with this additional feature.
- Evaluation (R^2 Score):
  - Linear Regression: 0.988
  - Ridge Regression: 0.988
  - XGBoost: 0.987
- Final Model Selected:
  - Linear Regression, due to strong performance and better explainability.
- Post-Improvement Result:
  - Extreme errors reduced to 2%.



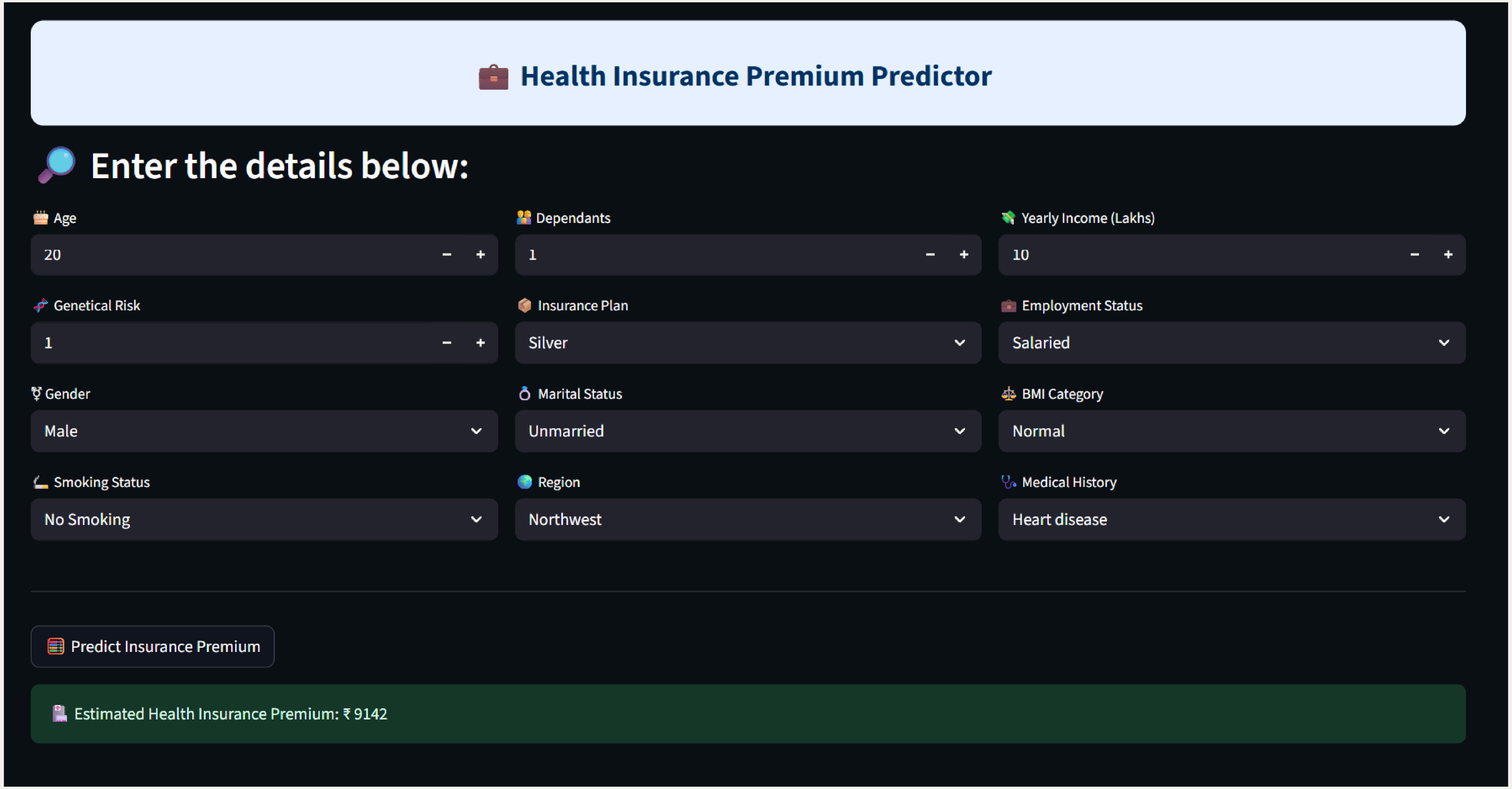
MODEL FLOW OVERVIEW



STREAMLIT APP INTEGRATION

- Developed an interactive web application using Streamlit.
- Integrated the trained model to allow real-time premium prediction based on user input.
- Handled preprocessing steps within the app to ensure consistent predictions.
- Implemented age-based model selection logic inside the app for accurate segmentation.
- Deployed the app on Streamlit Cloud for public access.
- The app enables users to enter features such as age, income, medical history, and get instant premium predictions.

USER INTERACTION PREVIEW



PROJECT SUMMARY

- Built a machine learning model to predict health insurance premiums based on user data.
- Cleaned and preprocessed the dataset, handled outliers, and engineered features like risk scores.
- Trained and compared multiple models (Linear, Ridge, XGBoost) using a 70:30 train-test split.
- Initially observed 30% of predictions had extreme errors (±10% or more).
- Performed age-based segmentation to investigate error sources:
  - For Age > 25, extreme errors were only 0.3% → used XGBoost.
  - For Age < 25, errors were initially 73%, reduced to 2% after adding genetic risk → used Linear Regression for better explainability.
- Deployed the final solution as an interactive Streamlit web application with segment-specific model selection.
- Live App: https://vaibhav-project-premium-prediction.streamlit.app
- GitHub Repository: https://github.com/vaibhavgarg2004/Health-Insurance-Premium-Predictor

THANK YOU