Health Insurance Premium Prediction

By Frederick Kobla Mensah

Abstract

This project explores the development of predictive models to estimate individual health insurance premiums based on demographic and behavioral attributes. With rising healthcare costs, accurate premium prediction models can play a pivotal role in supporting both consumers and insurance providers in decision-making. Initially, a large dataset with over 1.2 million records was considered but was later discarded due to significant missing data that compromised model reliability. A cleaner, more manageable dataset was adopted, comprising 1,338 observations and seven key variables, including age, sex, BMI, number of children, smoking status, and region.

We conducted a comprehensive exploratory data analysis (EDA) to understand feature distributions and detect outliers, followed by data preprocessing, including encoding and scaling. Several machine learning algorithms were evaluated, including Linear Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting, and XGBoost. Hyperparameter tuning was performed using GridSearchCV to enhance model performance. Among all models, XGBoost achieved the best results, with an R^2 of 0.88, MAE of 2427 and RMSE of approximately 4,309.

Feature importance analysis revealed that smoking status, BMI, and age were the most influential predictors of insurance costs. The model findings provide actionable insights for insurance pricing strategies and wellness initiatives aimed at high-risk individuals. Future work will explore the use of RandomizedSearchCV or even much better, Bayesian Optimization which is often smarter than the two tuning techniques mentioned earlier. This project demonstrates the effectiveness of tree-based machine learning models in real-world insurance applications.

Introduction

Health insurance is a critical component of modern healthcare systems, providing individuals and families with financial protection against unexpected medical expenses. Accurately predicting health insurance premiums is essential for insurers to assess risk and set equitable rates, and for consumers to anticipate and manage their healthcare costs. This project aims to develop and evaluate machine learning models that can predict individual health insurance premiums based on personal and demographic characteristics.

The dataset used in this study is sourced from the Machine Learning course website (Spring 2017) hosted by Professor Eric Suess at California State University, East Bay. It contains 1,338 observations and 7 features, including 4 numerical variables: age, bmi (Body Mass Index), children, and expenses (insurance premium), and 3 categorical variables: sex, smoker, and region. These categorical variables were encoded numerically to be compatible with machine learning models. This dataset is widely used for regression tasks in educational and research settings due to its clean structure and relevance to real-world insurance prediction problems.

This study is guided by the following key questions:

- What are the most significant predictors of health insurance premium costs?
- Can we build a predictive model that accurately estimates premiums based on individual characteristics?
- Which machine learning algorithms perform best in this context?

Literature Review

Prior studies have explored the use of statistical and machine learning techniques to model insurance premiums. For example, Ghosh et al. (2019) applied regression models to predict healthcare costs and emphasized the importance of variables like age, BMI, and smoking status. Similarly, Dinh and Nguyen (2021) demonstrated that ensemble models, particularly XGBoost, outperform traditional linear models in premium prediction tasks. Research has also focused on explainability; Lundberg and Lee (2017) introduced SHAP (SHapley Additive exPlanations) to better interpret feature impact in black-box models.

In industry applications, actuaries and data scientists alike are increasingly adopting machine learning to enhance underwriting accuracy and reduce adverse selection. Tutorials from platforms like Towards Data Science and DataCamp have highlighted the predictive value of tree-based models and the importance of preprocessing steps such as feature encoding and outlier detection. By combining insights from the literature with rigorous experimentation, this project aims to build a robust, interpretable model that can be adapted into real-world applications such as premium calculators or automated underwriting tools.

3.0 Methodology

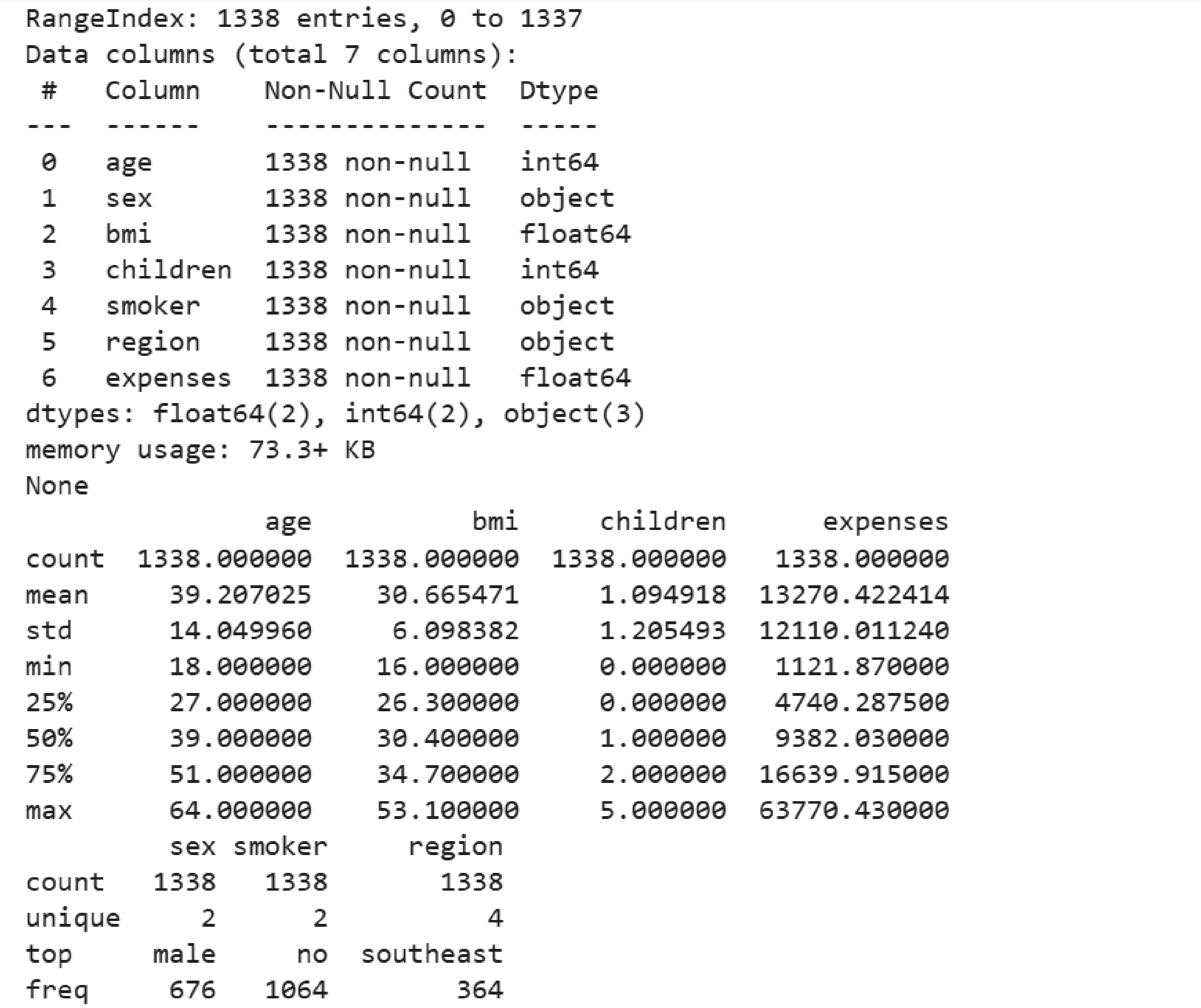
3.1 Exploratory Data Analysis (EDA)

The initial phase of analysis involved a comprehensive exploratory data analysis (EDA) to understand the structure, distribution, and relationships among variables in the dataset. This process began with **summary statistics** to evaluate the central tendency, spread, and potential presence of outliers or skewness across numerical features.

As shown in **Figure 3.1**, the dataset consists of 1,338 observations and seven variables, including both numerical (age, bmi, children, expenses) and categorical (sex, smoker, region) features. The mean insurance expense was approximately **$13,270**, while the median was significantly lower at **$9,382**, and the maximum value exceeded **$63,000**, indicating a strong **right-skew** in the target variable. This initial insight guided later decisions such as **log-transforming the target** for models sensitive to distributional assumptions.

Following the summary statistics, visualizations including **histograms**, **boxplots**, and a **correlation heatmap** provided deeper insights into variable relationships. The **smoker** variable showed a substantial impact on expenses, with smokers incurring significantly higher costs. **BMI** and **age** also displayed moderate positive correlations with the target, while **region**, **sex**, and **children** exhibited weak or negligible associations. These findings highlighted **smoker status** as a dominant predictor and justified the creation of **interaction terms** (e.g., bmi × smoker, age × smoker) in subsequent modeling steps.

.



```
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   expenses  1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
None
               age          bmi     children      expenses
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.665471     1.094918  13270.422414
std      14.049960     6.098382     1.205493  12110.011240
min      18.000000    16.000000     0.000000   1121.870000
25%      27.000000    26.300000     0.000000   4740.287500
50%      39.000000    30.400000     1.000000   9382.030000
75%      51.000000    34.700000     2.000000  16639.915000
max      64.000000    53.100000     5.000000  63770.430000
         sex smoker     region
count   1338   1338       1338
unique     2      2          4
top     male     no  southeast
freq     676   1064        364
```

Figure 3.1*: Summary statistics of numerical and categorical variables. The right-skewness of expenses and wide range in bmi and age are notable.*

To further investigate the skewness observed in the summary statistics, a histogram of the “**expenses**” variable was plotted (Figure 3.2). The distribution is heavily **right skewed**, with most of the data concentrated below $20,000 and a long tail extending toward higher premium values. This pattern supports the decision to apply a **log transformation** on the target variable, particularly for linear and distance-based models such as **Linear Regression** and **KNN**, where normality and homoscedasticity assumptions are relevant.

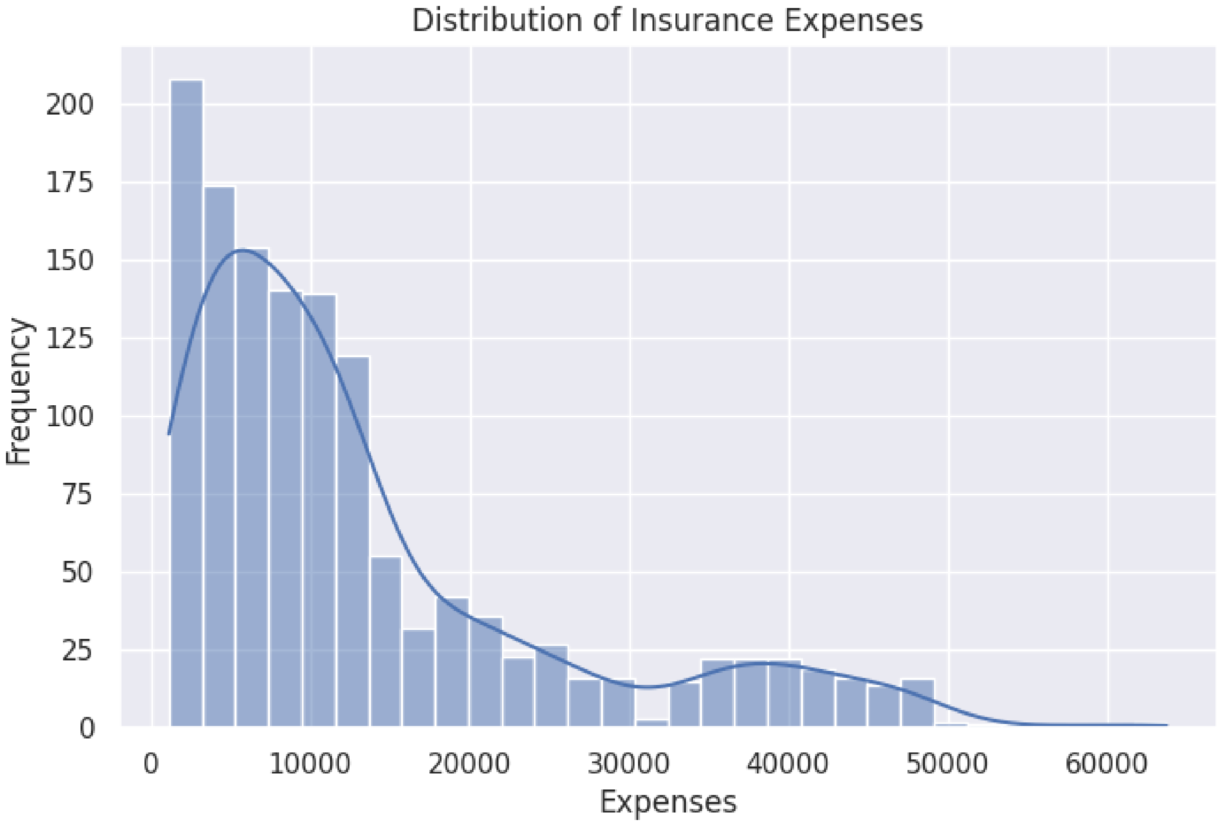


Figure 3.2*: Histogram of insurance expenses showing a highly right-skewed distribution. Most values are concentrated below $15,000, with a long tail toward higher costs.*

Also, a boxplot was created to compare **insurance expenses between smokers and non-smokers** (Figure 3.3). The visualization revealed a **striking difference** in the distribution of expenses across the two groups. Smokers consistently incurred **much higher insurance premiums**, with median expenses and upper quartiles significantly elevated compared to non-smokers. This finding not only reinforces the strong correlation identified earlier but also suggests that **smoker status is one of the most influential predictors** of insurance cost, making it a prime candidate for inclusion in predictive modeling.

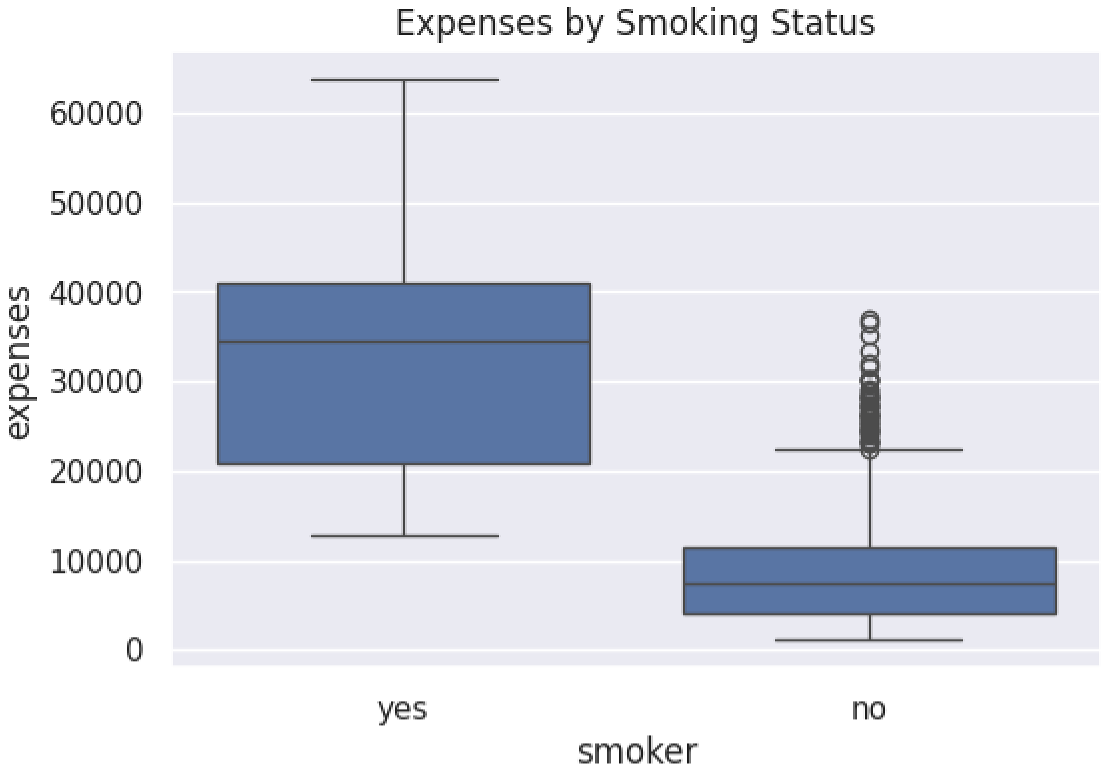


Figure 3.3*: Boxplot comparing insurance expenses between smokers and non-smokers, highlighting a significant cost difference.*

Additionally, to quantify the strength of relationships between numerical features, a **correlation heatmap** was generated (Figure 3.4). The heatmap confirmed a **moderate positive correlation** between both **age** and **BMI** with expenses, aligning with patterns observed in earlier scatterplots. In contrast, features such as children showed a **weaker correlation**, while categorical variables such as sex and region were not included, as they had been one-hot encoded for modeling. These findings helped prioritize variables for further preprocessing and informed **feature selection** in model training.

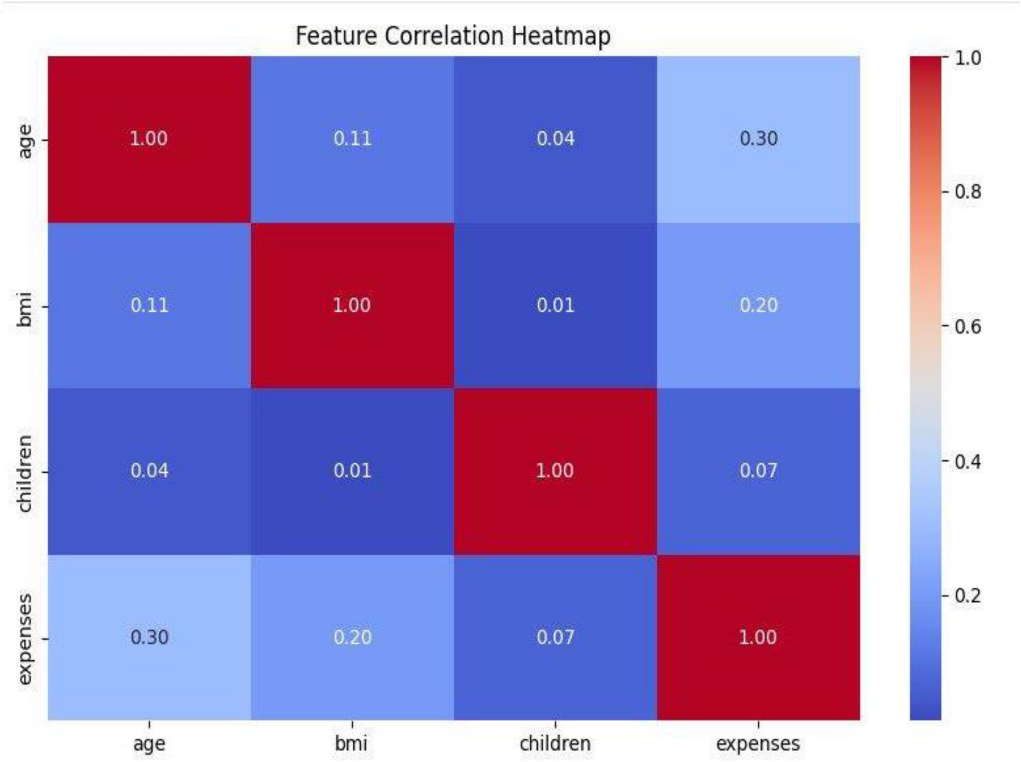


Figure 3.4*: Correlation heatmap showing that 'BMI' and 'age' have moderate positive correlations with health insurance expenses, while other features such as 'children' show weak or negligible associations*

Lastly, a scatterplot was created to visualize the relationship between **BMI and insurance expenses**, further **segmented by smoker status** (Figure 3.5). The plot clearly shows that **smokers incur significantly higher expenses** than non-smokers across nearly all BMI ranges. This visual reinforces the earlier findings and reveals a potential **interaction effect** between bmi and smoker. These insights guided the creation of engineered features such as bmi × smoker and age × smoker, which were later included in model training to improve predictive power.

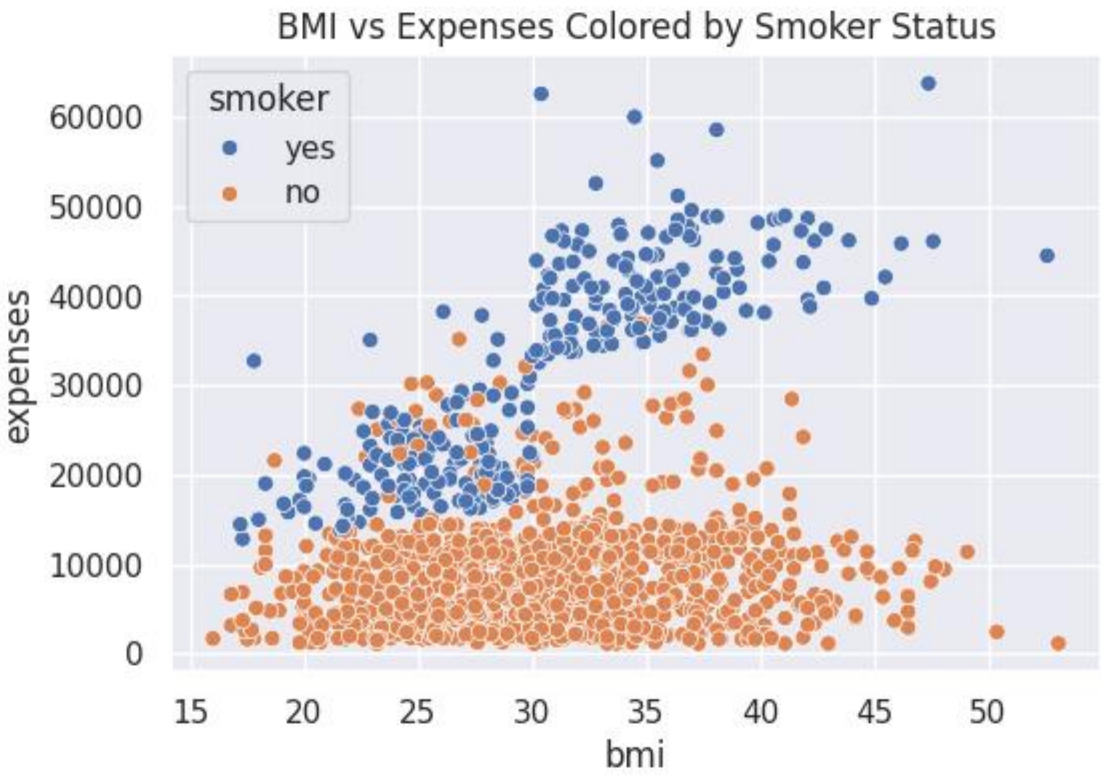


Figure 3.5*: Scatterplot of BMI vs. Expenses colored by smoker status. The plot shows that smokers consistently incur higher costs, indicating an interaction effect*

3.2 Data Preprocessing

Prior to model development, a structured data preprocessing pipeline was implemented to prepare the dataset for effective machine learning. The steps addressed both the nature of the variables and the assumptions required by different types of models.

1. Encoding of Categorical Variables

Categorical features—specifically sex, region, and smoker—were transformed using **one-hot encoding** to convert them into a numerical format compatible with machine learning algorithms. For instance, the smoker variable was converted into a binary feature named smoker_yes, enabling models to interpret this binary classification as a predictive signal. Redundant columns were dropped to avoid multicollinearity and dimensional redundancy.

2. Log Transformation of Target Variable

Exploratory analysis revealed that the target variable expenses was **highly right-skewed**, with extreme values affecting the mean and variance. To address this, a **log transformation** (log1p) was applied to the target, primarily for algorithms such as **Linear Regression** and **K-Nearest Neighbors**, which assume normally distributed residuals and benefit from reduced heteroscedasticity.

3. Train-Test Split Strategy: Dual Partitioning

To accommodate different modeling assumptions, a **dual train-test splitting approach** was adopted:

- For **tree-based models** (Random Forest, Gradient Boosting, XGBoost), the dataset was split using the **raw target values**, as these models are robust to skewed distributions and do not assume normality.
- For models trained on the **log-transformed target**, a separate train-test split was created to ensure appropriate evaluation conditions for algorithms that rely on linear assumptions.

This strategy allowed each class of model to operate under conditions best suited to its architectural strengths.

4. Feature Scaling

Since scaling is essential for algorithms that compute distances or are sensitive to feature magnitude, **StandardScaler** was applied to standardize numerical predictors (age, bmi, children) for **Linear Regression** and **KNN**. Scaling ensures all features contribute equally to the learning process and improves model convergence. This step was intentionally **excluded** for tree-based models, which are inherently **scale-invariant**.

5. Interaction Terms for Linear Models

To improve the flexibility of linear models in capturing complex patterns, **interaction features** were engineered manually. New variables such as age × smoker, bmi × smoker, and children × smoker were constructed based on insights from EDA, particularly where combined effects were visually evident. These terms were added exclusively to the log-transformed dataset used for linear modeling, and empirical evaluation showed a **notable improvement in model performance**.

3.3 Model Training and Evaluation

The modeling phase involved a progressive and experimental approach to evaluate how different algorithms perform under varying preprocessing conditions. The goal was not only to maximize predictive performance but also to reinforce understanding of model behavior under different levels of data transformation and feature engineering.

1. Linear Regression and K-Nearest Neighbors (KNN)

The modeling process began with two baseline models—**Multiple Linear Regression** and **K-Nearest Neighbors**—to establish a benchmark. Both models were trained using:

- **Log-transformed target variable** (log(expenses))
- **Standardized numerical features**
- **Manually engineered interaction terms** (e.g., bmi × smoker, age × smoker)

This configuration aimed to satisfy assumptions of linearity and normality required by these algorithms.

Findings:

- **Linear Regression** achieved an R^2 score of **0.83**, MAE ≈ **2,396**, and RMSE ≈ **5,063**.
- **KNN** performed slightly worse with R^2 = **0.81**, MAE ≈ **3,069**, and RMSE ≈ **5,369**.

  These results show that while log transformation and interaction terms helped, both models were limited in capturing the dataset's complex, non-linear patterns.

2. Tree-Based Models with Log-Transformed Target (No Interaction Terms)

Four tree-based regressors—**Decision Tree**, **Random Forest**, **Gradient Boosting**, and **XGBoost**—were then trained using:

- **Log-transformed target variable**
- **Original features (no interaction terms)**
- **No feature scaling** (tree models are scale-invariant)

This setup leveraged tree models' ability to inherently detect non-linearities and feature interactions.

Findings:

- **Random Forest** achieved the highest performance with $R^2 \approx$ **0.87**, MAE ≈ **2,168**, RMSE ≈ **4,249**.
- **Gradient Boosting** and **XGBoost** followed closely, while **Decision Tree** underperformed ($R^2 \approx$ **0.66**).
- The results confirmed that **tree-based models effectively learn interactions** without manual feature engineering.

3. Tree-Based Models with Raw Target and Predictors

To test models under the most natural conditions, tree-based models were also trained with:

- **Raw (untransformed) expenses target**

- **Original predictors**
- **No log transformation**
- **No scaling**
- **No manually engineered interactions**

Findings:

- Surprisingly, this configuration produced **even better results** than the log-transformed setups.
- **Gradient Boosting** achieved R² ≈ **0.88**, MAE ≈ **2,469**, and RMSE ≈ **4,334**.
- **Random Forest** performed similarly well; **XGBoost** was slightly behind.

This suggests that for tree-based models, **retaining raw values preserved signal variance**, and the models' internal structure handled complexity and skewness effectively.

4. Tree-Based Models with Manually Created Interaction Terms

As an additional experiment, tree-based models were trained on the dataset **with explicit interaction terms** manually added. However, **this did not yield better performance** than allowing the models to learn interactions on their own.

This supports the idea that **manual interaction terms may introduce redundancy or noise** in models already capable of learning complex relationships natively.

5. Hyperparameter Tuning and Final Evaluation

To refine performance, **GridSearchCV** was applied to the top-performing models—Random Forest, Gradient Boosting, and XGBoost—using a carefully selected hyperparameter grid. Tuning parameters included:

- n_estimators, max_depth, min_samples_split, min_samples_leaf
- learning_rate, subsample, max_features, colsample_bytree

After tuning, **XGBoost surpassed all other models**, showing the most significant gains across all metrics.

Final Tuned Performance (Raw Target & Predictors):

- **XGBoost**: R^2 = **0.8804**, MAE = **2,427.62**, RMSE = **4,309.89**
- **Gradient Boosting**: R^2 = **0.8802**, MAE = **2,485.37**, RMSE = **4,313.15**
- **Random Forest**: R^2 = **0.8669**, MAE = **2,844.40**, RMSE = **4,546.40**

6. Feature Importance Analysis

To understand what drove model decisions, feature important plots were extracted from the tuned ensemble models. All three models consistently ranked:

- **smoker_yes** as the most important predictor,
- Followed by **BMI** and **age**,
- With **children**, **region**, and **sex** contributing minimally.

These findings aligned with initial EDA insights and reinforced confidence in the model's interpretation of underlying patterns.

7. Final Model Selection

Based on performance, interpretability, and robustness, the **final selected model** was:

Tuned XGBoost trained on raw predictors and raw target variable, without engineered interaction terms.

This model balanced predictive accuracy with computational efficiency and will be deployed using serialization (joblib) for future integration into applications or dashboards.

4. App Development and Deployment

To ensure real-world usability of the trained machine learning model, a web application was developed using **Streamlit**, a Python-based open-source framework tailored for building and deploying data science tools.

This application allows users to interact with the model in a seamless and intuitive way, making real-time premium predictions based on a small set of demographic and lifestyle inputs.

4.1 Why Streamlit?

Streamlit was chosen due to its simplicity, fast deployment capabilities, and strong integration with Python-based machine learning workflows. Compared to traditional web frameworks like Flask or Django, Streamlit provides:

- Fewer development overheads (no need for separate front-end or backend layers),
- Clean default UI components (e.g., sliders, buttons, dropdowns),
- Built-in support for hosting (via **Streamlit Community Cloud**),
- Faster deployment cycles for academic and demo purposes.

4.2 Key Functional Features

The final version of the app includes the following functionalities:

- **Live Premium Prediction**: Users input their age, sex, region, number of children, BMI, and smoker status to get a real-time insurance premium estimate.
- **BMI Calculator**: For users who do not know their BMI, the app provides a simple module to calculate BMI based on height and weight. This value is automatically fed into the predictor.
- **Download Prediction Report**: After each prediction, users can download a **PDF report** summarizing the input data and predicted premium for personal reference.

- **Clean, Responsive Interface**: The interface is intuitive and responsive, supporting both light/dark modes and organized layout for mobile and desktop views.

4.3 Technical Implementation

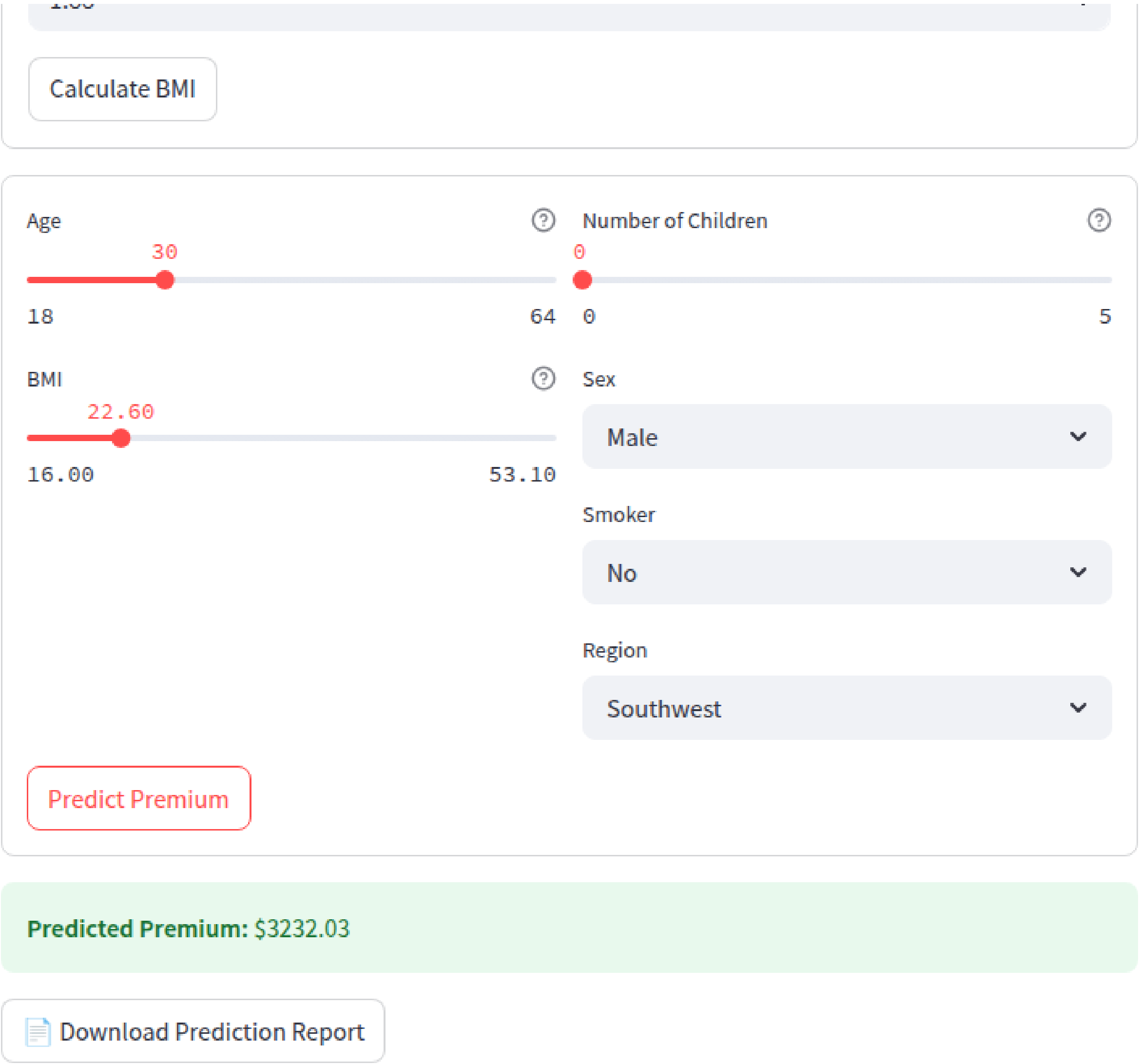
- The trained **XGBoost model**, selected for its superior performance ($R^2 = 0.8804$, MAE = $2,427.62, RMSE = $4,309.89), was saved using joblib.
- Upon user submission, the app converts all inputs to match the model’s expected format. **One-hot encoding** is applied dynamically to categorical variables such as sex, region, and smoker.
- No scaling is performed, as the final model was trained on raw values.

4.4 Deployment Strategy

- The application is hosted on **Streamlit Cloud**.
- The app’s codebase and model file are stored in a connected GitHub repository.
- A requirements.txt file ensures reproducibility, and the application is automatically rebuilt and redeployed when changes are pushed to GitHub.

4.5 Screenshots and UI Overview

The interface includes input sliders and dropdowns for age, BMI, children, sex, region, and smoker status. Users can compute BMI (if needed) and download a personalized PDF prediction report.



Age: 30 (18 – 64)

Number of Children: 0 (0 – 5)

BMI: 22.60 (16.00 – 53.10)

Sex: Male

Smoker: No

Region: Southwest

Predicted Premium: $3232.03

Figure 5.1 *– Prediction Interface with BMI and Report Download*

5. Conclusion and Discussion

This project successfully combined predictive analytics with real-world deployment by building a health insurance premium estimation tool from end to end. From data preparation and exploratory analysis to model training, tuning, and deployment, each step was designed to reflect both academic rigor and practical application.

5.1 Key Insights

- **EDA revealed** smoker status, BMI, and age as the most impactful predictors of insurance costs. These insights were later confirmed through feature importance analysis of ensemble models.
- **Tree-based models**, particularly Random Forest, Gradient Boosting, and XGBoost, significantly outperformed linear models like Multiple Linear Regression and KNN — even without scaling or manual feature engineering.
- **XGBoost** emerged as the best-performing model and was selected for deployment based on its high R^2, low error values, and consistent reliability after tuning.

5.2 Practical Deployment with Streamlit

- To make the model accessible and usable for non-technical users, the final model was embedded in a **Streamlit web application**. This enabled:
- Interactive prediction based on user-provided information,
- Optional BMI calculation for users without that data,
- PDF report download for offline reference and documentation.
- By bridging the gap between model and end-user, this app demonstrates the value of translating machine learning outcomes into practical decision-making tools.

5.3 Limitations and Future Work

- While the model performs well within its data context, it is limited by:
- The absence of socioeconomic or medical history variables,
- A dataset restricted to U.S.-based demographics.

5.4 Future directions include:

- Expanding features to include more health indicators,
- Improving interpretability via SHAP value explanations,
- Deploying a mobile-friendly or API-driven version for broader access.

5.5 Future Enhancements

Plans for future versions of the app include:

- **User login/account system** to track prediction history
- **Health analytics dashboard** for visualizing input trends and comparisons
- **RESTful API version** of the model for mobile or enterprise integration
- **Enhanced input validation** for error-proofing and user assistance

References

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Dinh, L. T., & Nguyen, T. T. (2021). Predicting Health Insurance Premiums Using Machine Learning Techniques. Journal of Healthcare Engineering, 2021, 1–10. https://doi.org/10.1155/2021/6631412

Ghosh, S., Mondal, S., & Das, D. (2019). Predicting Healthcare Costs Using Machine Learning. International Journal of Scientific & Technology Research, 8(8), 1456–1460. https://www.ijstr.org/final-print/aug2019/Predicting-Healthcare-Costs-Using-Machine-Learning.pdf

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 30, 4765–4774. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. http://www.jmlr.org/papers/v12/pedregosa11a.html

Python Software Foundation. (2024). Python Language Reference, version 3.10. https://www.python.org/

Render. (n.d.). Deploy your apps with Render. https://render.com/

Streamlit, Inc. (2020). Streamlit: Turn data scripts into shareable web apps. https://streamlit.io/

Suess, E. (2017). Insurance Dataset. Machine Learning Course, California State University, East Bay. http://www.sci.csueastbay.edu/~esuess/stat6620/#week-6

Towards Data Science. (n.d.). Tutorials on Regression, Tree-Based Models, and Feature Engineering. https://towardsdatascience.com/

Appendix – Access to Full Code

The complete source code used in this project for data cleaning, exploratory data analysis (EDA), model training, hyperparameter tuning, and Model selection.

This notebook is publicly accessible in **read-only mode** to support transparency, reproducibility, and future reference.

Access it here:

Health Insurance Premium Prediction – Google Colab Notebook