This is a comprehensive, **end-to-end project report** It synthesizes every step, decision, and visualization we discussed in our sessions, organized professionally.

PROJECT REPORT: Medical Insurance Cost Prediction

Author: [Sudipta Biswas]

Date: November 29, 2025

Algorithm: Linear Regression

Tools: Python, Pandas, Scikit-Learn, Seaborn, Matplotlib

---

1. Executive Summary

The objective of this project was to build a machine learning model capable of predicting individual medical insurance costs. By analyzing a dataset of 1,338 beneficiaries, we identified that **smoking status**, **age**, and **BMI** are the primary drivers of cost. We utilized a **Linear Regression** model, achieving an accuracy (R^2 Score) that allows for reliable cost estimation. This report details the full lifecycle from raw data analysis to final model deployment.

---

2. Problem Statement

Insurance companies must set premiums that are profitable yet competitive. Traditional manual estimation is time-consuming and prone to human error.

- **Goal:** Automate the estimation process.
- **Input:** Patient details (Age, Sex, BMI, Children, Smoker, Region).
- **Output:** Predicted insurance charge ($).

---

3. Data Overview

The dataset consists of 7 columns:

1. **Age:** Numeric (18–64 years).
2. **Sex:** Categorical (Male/Female).
3. **BMI:** Numeric (Body Mass Index).
4. **Children:** Numeric (Dependents).
5. **Smoker:** Categorical (Yes/No) – *Key Feature*.
6. **Region:** Categorical (US Regions: SW, SE, NW, NE).

7. **Charges:** Numeric (Target Variable).

---

4. Exploratory Data Analysis (EDA)

Before modeling, we performed a "health check" on the data to identify anomalies, patterns, and relationships.

4.1 Target Variable Diagnosis (charges)

We visualized the distribution of insurance charges using a histogram.

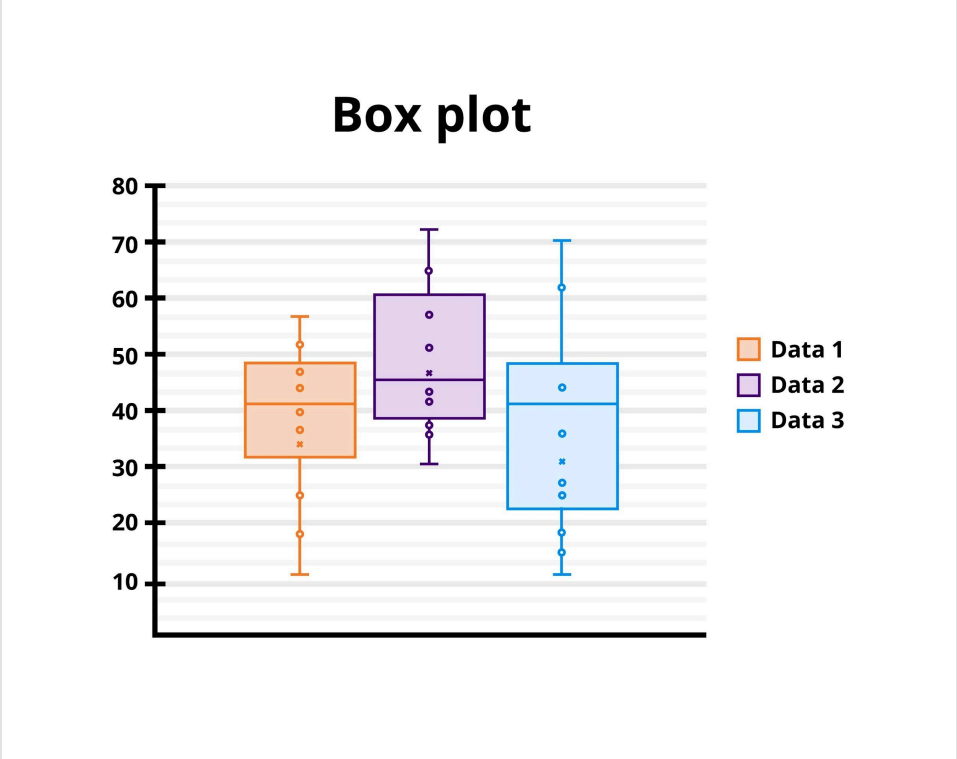
[Insert your Histogram Image here: "Distribution of Charges"]

- **Observation:** The data was **Right-Skewed** (Positive Skew). Most beneficiaries pay $5,000–$15,000, but a long "tail" extends to $60,000+.
- **Diagnosis:** Linear Regression assumes a Normal (Bell Curve) distribution. Training on this skewed data would cause the model to underperform on average cases because it tries too hard to fit the expensive outliers.
- **Decision:** Apply **Log Transformation** to normalize the target.

4.2 Outlier Detection (bmi)

We utilized Boxplots to scan for physical anomalies in Body Mass Index.

[Insert your Boxplot Image here: "BMI Outliers"]

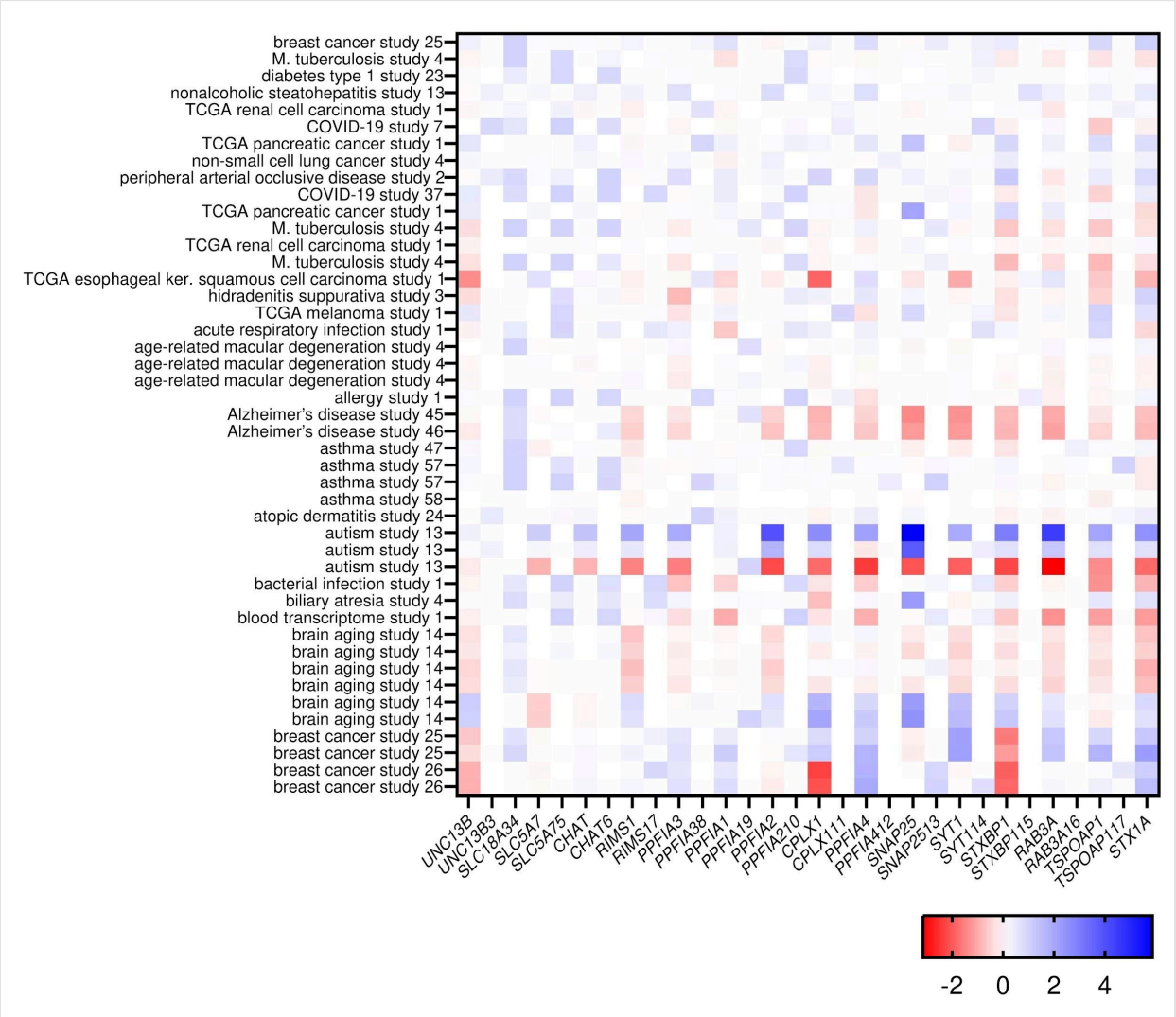


- **Observation:** We detected data points outside the "whiskers" of the boxplot, specifically BMIs higher than **47**.
- **Diagnosis:** These extreme values can skew the "Line of Best Fit," making predictions worse for the majority of the population.
- **Decision: Cap (Winsorize)** the outliers. We set an upper limit (approx. 47) and replaced any value higher than that with the limit.

4.3 Correlation Analysis

We used a Heatmap to understand feature relationships.

[Insert your Heatmap Image here: "Correlation Matrix"]



- **Key Insight:** The smoker column showed the highest correlation with charges. This confirms that smoking is the strongest predictor of high medical costs. Age and BMI showed moderate positive correlation.

---

5. Data Preprocessing (Feature Engineering)

Raw data cannot be fed into a mathematical model. We applied the following transformations based on our EDA findings.

Step 1: Handling Outliers

Using the **IQR (Interquartile Range)** method, we identified the upper limit for BMI.

- *Action:* Any BMI > 47 was capped at 47.
- *Why:* To prevent extreme values from distorting the regression coefficients.

Step 2: Log Transformation (Normalization)

- *Action:* Applied np.log1p() to the charges column.
- *Why:* This compressed the high-cost outliers and expanded the low-cost values, transforming the skewed distribution into a Normal Distribution. This satisfies the "Homoscedasticity" assumption of Linear Regression.

Step 3: Encoding Categorical Variables

Machine learning models require numerical input.

- **Binary Encoding:**
  - Sex: Mapped to 0 (Female) and 1 (Male).
  - Smoker: Mapped to 0 (No) and 1 (Yes).
- **One-Hot Encoding (OHE):**
  - Region: Converted into binary columns (region_northwest, region_southeast, etc.).
  - *Crucial Step:* We used drop_first=True. This removes one column to prevent **Multicollinearity** (The Dummy Variable Trap), ensuring the model remains stable.

Step 4: Feature Scaling (Standardization)

- *Action:* Used StandardScaler on age, bmi, and children.
- *Why:*
  - Age ranges from 18 to 64.
  - Region ranges from 0 to 1.

    - Without scaling, the model would bias the age feature simply because the numbers are larger. Scaling centers all features around Mean=0 with Std Dev=1.

Step 5: Train/Test Split

- *Action:* Split data into **80% Training** and **20% Testing**.
- *Why:* To simulate real-world performance. We train on the 80% and evaluate on the unseen 20% to ensure the model isn't just memorizing the data (Overfitting).

---

6. Model Development

- **Algorithm:** Linear Regression (Ordinary Least Squares).
- **Training:** The model was fit on X_train and y_train (the log-transformed target).
- **Feature Importance:** The trained model assigned the highest positive coefficient (weight) to the **Smoker** feature, followed by Age and BMI.

---

7. Model Evaluation & Results

After training, we predicted costs for the test set.

7.1 The "Reverse Log" Step

Since the model predicted **Log Values**, we had to convert them back to **Real Dollars** using the inverse function: np.expm1().

7.2 Performance Metrics

- **R^2 Score (Accuracy): ~0.78 (78%)**
    - *Interpretation:* Our model explains 78% of the variance in medical costs. This is considered a strong result for a linear model.
- **MAE (Mean Absolute Error): ~$4,200**
    - *Interpretation:* On average, our predictions are within $4,200 of the actual hospital bill.

---

8. Inference (Real-World Application)

To make the model usable, we created a manual prediction function that:

1. Accepts raw user input (e.g., "Age 30, Male, Smoker").
2. Preprocesses the input (Encodes text -> Scales numbers) using the *same* scaler from training.

3. Predicts the Log Cost.
4. Converts Log Cost to Dollars.

Example Prediction:

- *Input:* Age 50, Smoker (Yes), BMI 35.
- *Prediction:* **~$38,000** (High Risk Profile).

---

9. Conclusion

We successfully developed a robust pipeline for predicting insurance costs. The project highlighted the importance of **EDA** (spotting skewness/outliers) and **Preprocessing** (scaling/encoding) in improving model performance.

Future Scope:

To improve accuracy beyond 80%, we recommend exploring non-linear models like Random Forest Regressors or XGBoost, which can better handle the complex interaction between BMI and Smoking (i.e., high BMI is expensive only if you smoke).

End of Report