Medical Insurance Cost Prediction using Linear Regression

1. Introduction

This project applies a supervised machine learning algorithm — Linear Regression — to predict the cost of medical insurance for individuals based on demographic and personal health-related features.

Linear regression is ideal for this use case since the target variable (insurance charges) is continuous. The project demonstrates how a basic machine learning pipeline can be built for a real-world problem using Python and popular libraries.

---

2. Problem Statement

The objective is to develop a model that accurately predicts medical insurance charges based on features such as age, gender, BMI, number of children, smoking status, and region of residence. This predictive model can assist insurance companies in assessing potential charges more effectively.

---

3. Understanding the Dataset

- **Dataset Name**: insurance.csv
- **Rows**: 1338
- **Features**:
    - `age`: Age of the policyholder
    - `sex`: Gender (male/female)
    - `bmi`: Body Mass Index
    - `children`: Number of children covered under insurance
    - `smoker`: Whether the individual is a smoker (yes/no)
    - `region`: Residential region in the US (northeast, northwest, southeast, southwest)
    - `charges`: Target variable — the medical insurance cost

The dataset is loaded into a Pandas DataFrame and basic inspection (head, shape, info) is performed.

---

4. Data Preprocessing

4.1 Handling Missing Values

No missing values were found in the dataset.

4.2 Encoding Categorical Variables

- `sex` and `smoker`: Label encoded
- `region`: One-Hot Encoded

4.3 Feature Selection

All features are relevant; no redundant fields were removed.

4.4 Feature Scaling

Scaling was not applied as linear regression in scikit-learn can handle unscaled input for this dataset.

---

5. Exploratory Data Analysis (EDA)

- **Univariate Analysis**: Histograms for each feature to understand distribution
- **Bivariate Analysis**: Boxplots for `charges` vs categorical variables (e.g., smoker, region)
- **Correlation Matrix**: Heatmap used to assess feature relationships
- **Outlier Detection**: Boxplots and distribution plots reveal some outliers in `bmi` and `charges`

---

6. Building the Linear Regression Model

6.1 Splitting the Dataset

Used `train_test_split` from scikit-learn with an 80-20 split.

6.2 Model Training

Fitted a `LinearRegression` model from scikit-learn on the training data.

6.3 Model Coefficients

Extracted model coefficients to understand the influence of each independent variable.

---

7. Model Evaluation

Metrics Used:

- **R-squared (R^2)**: Proportion of variance explained by the model
- **Mean Squared Error (MSE)**: Average squared prediction error
- **Mean Absolute Error (MAE)**: Average absolute prediction error
- **Root Mean Squared Error (RMSE)**: Square root of MSE

Code:

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

test_data_prediction = regressor.predict(X_test)

r2_test = r2_score(Y_test, test_data_prediction)
mse = mean_squared_error(Y_test, test_data_prediction)
mae = mean_absolute_error(Y_test, test_data_prediction)
rmse = np.sqrt(mse)

print("R squared value  :", r2_test)
print("Mean Squared Error (MSE) :", mse)
print("Mean Absolute Error (MAE):", mae)
print("Root Mean Squared Error (RMSE):", rmse)
```

---

8. Interpretation of Results

- **R^2** score indicates how well the model explains the variability of charges.
- Smoking and BMI are found to be the most significant predictors.
- Smokers incur significantly higher charges.

---

9. Conclusion

The model performs reasonably well in predicting medical insurance charges. Key insights include:

- Smoking and BMI are strong indicators of higher insurance cost.
- Linear regression provides interpretability and ease of implementation.

Future Improvements:

- Try regularized regression (Ridge, Lasso)
- Add interaction terms or polynomial features
- Apply cross-validation for model robustness

---

10. Appendix (Sample Code Snippets)

```
# Load data
insurance_dataset = pd.read_csv('insurance.csv')

# Preprocessing
insurance_dataset = pd.get_dummies(insurance_dataset, drop_first=True)

# Split data
X = insurance_dataset.drop(columns='charges', axis=1)
Y = insurance_dataset['charges']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=2)

# Train model
regressor = LinearRegression()
regressor.fit(X_train, Y_train)
```