MEDICAL INSURANCE COST PREDICTION - FULL PROJECT DOCUMENTATION
---

1. INTRODUCTION

The Medical Insurance Cost Prediction project uses machine learning regression techniques to predict
the healthcare insurance cost of an individual. The prediction depends on features such as age, BMI,
number of children, region, and smoking status. The aim is to understand which factors significantly
influence medical charges.

2. DATA COLLECTION AND UNDERSTANDING

The dataset includes the following features:
  - Age: Age of the individual
  - Sex: Gender (male/female)
  - BMI: Body Mass Index (a measure of body fat)
  - Children: Number of dependents
  - Smoker: Indicates if the person is a smoker (yes/no)
  - Region: Residential region (northeast, northwest, southeast, southwest)
  - Charges: The medical insurance cost (target variable)

Data loading example:
```
    import pandas as pd
    data = pd.read_csv('insurance.csv')
    data.head()
```

3. DATA PREPROCESSING

The preprocessing step includes handling categorical data, encoding, and preparing features for model input.

  - Label Encoding or One-Hot Encoding for categorical features
  - Scaling features like BMI or Charges if necessary
  - Checking for missing values and cleaning data

Example code:
```
    from sklearn.preprocessing import LabelEncoder
    encoder = LabelEncoder()
    data['sex'] = encoder.fit_transform(data['sex'])
    data['smoker'] = encoder.fit_transform(data['smoker'])
    data['region'] = encoder.fit_transform(data['region'])
```

4. EXPLORATORY DATA ANALYSIS (EDA)

```
EDA is performed to understand data patterns and relationships using
seaborn and matplotlib visualizations.

Common plots used:
  - Distribution of charges using sns.displot()
  - Correlation heatmap to identify feature relationships
  - Pairplots for multivariate visualization

Example code:
    import seaborn as sns
    sns.displot(data['charges'], kde=True)
    sns.heatmap(data.corr(), annot=True)

5. DATA SPLITTING

The data is divided into training and test sets to evaluate model
performance.

    from sklearn.model_selection import train_test_split
    X = data.drop(columns='charges', axis=1)
    Y = data['charges']
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=2)

6. MODEL TRAINING

A Linear Regression model is trained to predict insurance costs.

    from sklearn.linear_model import LinearRegression
    model = LinearRegression()
    model.fit(X_train, Y_train)

7. MODEL EVALUATION

The model performance is evaluated using R² score and Mean Absolute
Error (MAE).

    from sklearn.metrics import r2_score, mean_absolute_error
    y_pred = model.predict(X_test)
    print('R² Score:', r2_score(Y_test, y_pred))
    print('MAE:', mean_absolute_error(Y_test, y_pred))

8. BUILDING A PREDICTIVE SYSTEM

After training, the model can predict costs for new input data.

Example code:
    input_data = (46, 1, 33.44, 1, 0, 2)
    input_df = pd.DataFrame([input_data], columns=X.columns)
```

```
    prediction = model.predict(input_df)
    print(prediction)

Predicted output:
    array([1520.59])

9. CONCLUSION

The model successfully predicts medical insurance costs based on
multiple personal and health factors.
The R² value indicates a strong model fit, confirming the effectiveness
of regression in this context.

10. FUTURE IMPROVEMENTS

  - Apply polynomial regression for non-linear relationships
  - Implement feature scaling and regularization techniques
  - Use ensemble models such as Random Forest or XGBoost
  - Integrate the model with a Streamlit or Flask web interface

------------------------------------------------------

AUTHOR

This documentation was prepared by Satyam Gajjar.

------------------------------------------------------
```