REGRESSION ALGORITHM

Assignment

Abstract

To predict the insurance charge based on the insurer data

10th August 2025

Ilango P

pailango@gmail.com

Objective

To develop a predictive model that estimates insurance charges based on various input parameters.

Dataset Overview

The client has provided a dataset containing 1,338 records with the following features:

- **Age**
- **Sex**
- **BMI**
- **Children**
- **Smoker**
- **Charges** (Target variable)

The goal is to predict the **Charges** using the other fields as input features. The dataset size and feature set are considered sufficient for model development.

Model Development Approach

1. Domain Selection

- The dataset consists of numerical and categorical data, making it suitable for **Machine Learning**.

2. Learning Type

- Since both input features and the target variable are available and clearly defined, this falls under **Supervised Learning**.

3. Learning Task

- The target variable is a continuous numerical value, indicating a **Regression** problem.

Modelling Phases

Data Preparation

- **Data Source**: The dataset is provided in a file named Test.csv.
- **Feature Types**:
    - **Numerical**: age, bmi, children – no preprocessing required.

- **Categorical**: sex, smoker – need to be converted to numerical format.
- Both are binary and ordinal (e.g., male/female, yes/no).
- Encoding options: **Label Encoding** or **One-Hot Encoding** (both yield similar results).

Train-Test Split

- The dataset will be split into training and testing sets in a **70:30 ratio**.

Model Training and Evaluation

The following regression algorithms were evaluated:

Support Vector Machine

Sl.No.	Hyper parameter	default	linear	rbf	poly	sigmoid
1	default	-0.08338				
2	1		-0.111661287	-0.08843	-0.06429	-0.07543
3	10		-0.001617632	-0.08197	-0.09312	0.039307
4	100		0.54328182	-0.1248	-0.09976	0.52761
5	1000		0.634036931	-0.11749	-0.05551	0.287471
6	5000		**0.764893815**	-0.0731	0.146224	-7.53004
7	10000		0.744482485	-0.01728	0.352902	-34.1515

Decision Tree

Sl.No.	criterion	splitter	max_features	R2 value
1	default	default	default	0.710562
2	squared_error	best	None	0.701941
3	squared_error	best	log2	0.715498
4	squared_error	best	sqrt	0.7223
5	squared_error	random	None	0.741621
6	squared_error	random	log2	0.66082
7	squared_error	random	sqrt	0.662775
8	absolute_error	best	None	0.659213
9	absolute_error	best	sqrt	0.687251
10	absolute_error	random	None	**0.746062**
11	absolute_error	random	sqrt	0.674751
12	friedman_mse	best	None	0.678774
13	friedman_mse	best	sqrt	0.693982
14	friedman_mse	random	None	0.719241
15	friedman_mse	random	sqrt	0.711558

16	poisson	best	None	0.727782
17	poisson	best	sqrt	0.589042
18	poisson	random	None	0.690436
19	poisson	random	sqrt	0.650584

Random Forest

Sl.No.	n_estimators	random_state	Criterion	R2 value
1	default	default	default	0.82515
2	1	0	None	0.604032
3	10	0	None	0.797463
4	50	0	None	0.821617
5	100	0	None	0.82276
6	1	10	None	0.680783
7	10	10	None	0.811367
8	50	10	None	**0.828431**
9	100	10	None	0.828152
10	1	0	squared_error	0.604032
11	10	0	squared_error	0.797463
12	50	0	squared_error	0.821617
13	100	0	squared_error	0.82276
14	1	10	squared_error	0.680783
15	10	10	squared_error	0.811367
16	50	10	squared_error	0.828431
17	100	10	squared_error	0.828152
18	1	0	friedman_mse	0.60461
19	10	0	friedman_mse	0.796371
20	50	0	friedman_mse	0.822835
21	100	0	friedman_mse	0.823662
18	1	0	absolute_error	0.633383
19	10	0	absolute_error	0.807406
20	50	0	absolute_error	0.825886
21	100	0	absolute_error	0.827003

1. Multiple Linear Regression

- **R^2 Score**: 0.78948

2. Support Vector Regression (SVR)

- Various kernels and hyperparameters were tested.

- **Best R^2 Score**: 0.76489 (with C=5000, kernel=linear)

3. Decision Tree Regressor

- Multiple configurations tested.
- **Best R^2 Score**: 0.74606 (criterion=absolute_error, splitter=random)

4. Random Forest Regressor

- Extensive hyperparameter tuning performed.
- **Best R^2 Score**: **0.82843**
    - Parameters: n_estimators=50, random_state=10, criterion=None

---

Model Selection

The **Random Forest Regressor** achieved the highest performance and is selected as the final model.
It will be saved as: final_model_randomforest.sav

Deployment Steps

1. **Load Model**
    - Use pickle to load the saved model file.
2. **Input Collection**
    - Collect user inputs: age, bmi, children, sex, smoker.
    - Convert sex and smoker from text to numerical format using conditional statements.
3. **Prediction**
    - Use the model's predict() function to estimate insurance charges.
4. **Action**
    - Use the predicted value to determine the insurance premium.
