

# Hope Artificial Intelligence

## Regression Algorithm – Assignment (Model identification using the “R<sup>2</sup> Score”)

### Requirement:

Customer wants to predict the insurance charges based on the several parameters and Client has provided the “insurance\_pre” dataset for processing.

### Dataset Overview:

**Rows:** 1,338

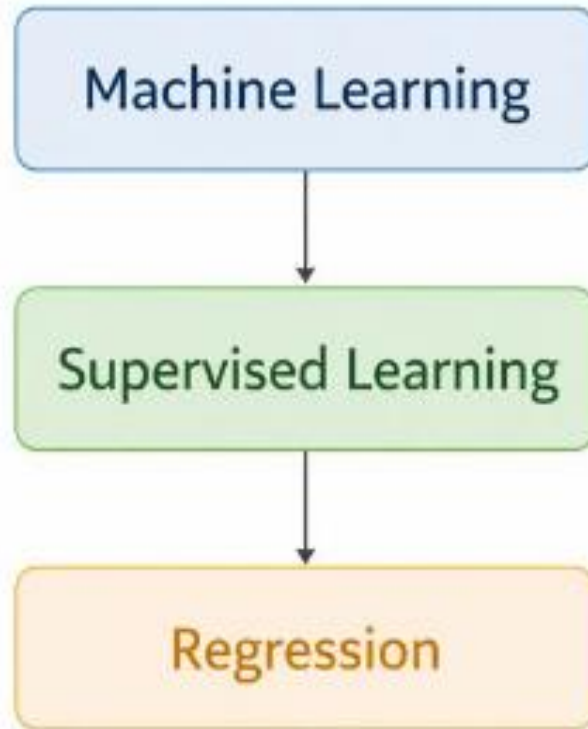
**Columns:** 6

**Missing values:** None

**Size:** Small-to-medium

**I/p and O/p:** The objective is to predict insurance charges. Therefore, the “**Charges**” column is considered as the output (dependent) variable, while all remaining columns are treated as input (independent) variables.

## 3 Stages of Problem Identification



## Process:

- The dataset contains multiple independent variables; therefore, Simple Linear Regression was not used. Instead, the  $R^2$  score was evaluated using Multiple Linear Regression, Support Vector Regression, Decision Tree Regression, and Random Forest Regression.
- Used “`dataset=pd.get_dummies(dataset,drop_first=True)`” coding to convert categorical variables into numerical form.
- Support Vector Regression (SVR) produced negative prediction values, making it unsuitable for this dataset without additional preprocessing or transformation.
- Multiple Linear Regression achieved an  $R^2$  score of approximately 78%.
- Decision Tree Regression yielded a moderately improved performance, achieving an  $R^2$  score above 80%.
- Random Forest Regression delivered the best performance, achieving an  $R^2$  score of around 89%, approaching the 90% benchmark.

Multiple Linear model R\_score value is 0.789

## Support vector Machine

kernel	degree	gamma	r_score
linear	3	auto	-0.010
poly	3	scale	-0.075
rbf	3	scale	-0.083
sigmoid	3	auto	-0.075

## Decision Tree Regression

criterion	splitter	max_depth	r_score
squared_error	best	5	0.847
squared_error	random	5	0.864
squared_error	best	4	0.882
friedman_mse	best	5	0.831
friedman_mse	best	4	0.882
friedman_mse	random	5	0.836
absolute_error	best	5	0.815
absolute_error	random	5	0.855
poisson	best	5	0.849
poisson	random	5	0.873

## Random Forest

n_estimators	criterion	max_depth	r_score
50	squared_error	5	0.847
10	squared_error	5	0.886
5	absolute_error	5	0.884
5	friedman_mse	5	0.885
n_estimators	max_depth	min_samples_leaf	r_score
200	8	2	0.881
300	5	2	0.891
250	5	6	0.892

```
[460]: RandomForestRegressor ⓘ ⓘ  
  
[461]: y_pred=regressor.predict(x_test)  
  
[462]: from sklearn.metrics import r2_score  
       r_score=r2_score(y_test, y_pred)  
  
[463]: r_score  
  
[463]: 0.8923642099635969
```

## Final Model Selection Based on $R^2$ Score

Based on the model evaluation, Random Forest Regression achieved an  $R^2$  score of approximately 89%, which is higher than the other models tested. Additionally, most parameter combinations for the Random Forest model consistently produced  $R^2$  scores around the same range.

Therefore, based on its stable performance and higher accuracy, **Random Forest Regression** is selected as the final model for this dataset.

*Thank  
you!*