Predicting Insurance Premiums Using Machine Learning Regression Techniques.

Dataset:

This dataset includes personal and health details of individuals to predict their medical insurance charges.

	age	sex	bmi	children	smoker	charges
0	19	female	27.900	0	yes	16884.92400
1	18	male	33.770	1	no	1725.55230
2	28	male	33.000	3	no	4449.46200
3	33	male	22.705	0	no	21984.47061
4	32	male	28.880	0	no	3866.85520
...	...	...	...	...	...	...
1333	50	male	30.970	3	no	10600.54830
1334	18	female	31.920	0	no	2205.98080
1335	18	female	36.850	0	no	1629.83350
1336	21	female	25.800	0	no	2007.94500
1337	61	female	29.070	0	yes	29141.36030

1338 rows × 6 columns

Pre-Processing methods:

1. **Handling Categorical Data: (Nominal)**

   Two categorical columns in the dataset were transformed into numeric form for analysis.

Column	Type	Encoding Applied	Details
sex	Nominal	One-Hot Encoding (drop_first = True)	Converted to sex_male (1 = male, 0 = female)
smoker	Nominal	One-Hot Encoding (drop_first = True)	Converted to smoker_yes (1 = smoker, 0 = non-smoker)

	age	bmi	children	charges	sex_male	smoker_yes
0	19	27.900	0	16884.92400	0	1
1	18	33.770	1	1725.55230	1	0
2	28	33.000	3	4449.46200	1	0
3	33	22.705	0	21984.47061	1	0
4	32	28.880	0	3866.85520	1	0
...	...	...	...	...	...	...
1333	50	30.970	3	10600.54830	1	0
1334	18	31.920	0	2205.98080	0	0
1335	18	36.850	0	1629.83350	0	0
1336	21	25.800	0	2007.94500	0	0
1337	61	29.070	0	29141.36030	0	1

1338 rows × 6 columns

Develop a good model with r2_score & Final Model Selection:

1. **Multiple Linear Regression:**

   regressor=LinearRegression()

   R^2 Score: 0.789479

2. **Support Vector Machine(svm):**
   regressor=SVR(kernel="", C=10)

Sl. No.	HYPER PARAMETER	LINEAR (R^2 Score)	RBF (NON-LINEAR) (R^2 Score)	POLY (R^2 Score)	SIGMOID (R^2 Score)
1	Default (C1.0)	-0.1116	-0.0884	-0.0642	-0.0899
2	C10	-0.0016	-0.0819	-0.0931	-0.0907
3	C100	0.5432	-0.1248	-0.0997	-0.1181
4	C500	0.6270	-0.1246	-0.0820	-0.4562

Kernel: Linear
Hyperparameter (C500):
R^2 Score: 0.6270

3. **Decision Tree:**
   regressor = DecisionTreeRegressor(criterion = 'absolute_error', splitter = 'random', max_features= 'log2')

Sl. No.	CRITERION	MAX FEATURES	SPLITTER	R^2 Score
1	squared_error	None	best	0.6855
2	squared_error	log2	best	0.6692
3	squared_error	int (max_features = 2)	best	0.6697
4	squared_error	None	random	0.6827
5	squared_error	log2	random	0.6556
6	friedman_mse	None	best	0.6796
7	friedman_mse	sqrt	best	0.6882
8	friedman_mse	log2	best	0.6970
9	friedman_mse	sqrt	random	0.6170
10	friedman_mse	log2	random	0.6696
11	absolute_error	None	best	0.6875
12	absolute_error	sqrt	best	0.7033
13	absolute_error	log2	best	0.7156

14	**absolute_error**	**None**	**random**	**0.6917**
15	**absolute_error**	**sqrt**	**random**	**0.6347**
16	**absolute_error**	**log2**	**random**	**0.7102**

Criterion: absolute_error
Splitter: random
max_features: log2
R^2 Score: 0.7156

4. **Random Forest:**
   regressor = RandomForestRegressor(criterion = "squared_error", n_estimators = 100, max_features = 2)

Sl. No.	CRITERION	MAX FEATURES	N_ESTIMATORS	R^2 Score
1	squared_error	None	100	0.8560
2	squared_error	sqrt	100	0.8719
3	squared_error	log2	100	0.8715
4	squared_error	int (max_features = 2)	100	0.8678
5	squared_error	float (max_features = 0.5)	100	0.8722
6	squared_error	None	10	0.8417
7	squared_error	sqrt	10	0.8637
8	squared_error	log2	10	0.8557
9	squared_error	int (max_features = 2)	10	0.8479
10	squared_error	float (max_features = 0.5)	10	0.8526
11	friedman_mse	None	100	0.8545
12	friedman_mse	sqrt	100	0.8710
13	friedman_mse	log2	100	0.8691
14	friedman_mse	int (max_features = 2)	100	0.8688
15	friedman_mse	float (max_features = 0.5)	100	0.8690
16	friedman_mse	None	10	0.8526
17	friedman_mse	sqrt	10	0.8557
18	friedman_mse	log2	10	0.8479
19	friedman_mse	int (max_features = 2)	10	0.8569
20	friedman_mse	float (max_features = 0.5)	10	0.8592
21	absolute_error	None	100	0.8565
22	absolute_error	sqrt	100	0.8710
23	absolute_error	log2	100	0.8693
24	absolute_error	int (max_features = 2)	100	0.8681
25	absolute_error	float (max_features = 0.5)	100	0.8705
26	absolute_error	None	10	0.8386
27	absolute_error	sqrt	10	0.8569
28	absolute_error	log2	10	0.8583
29	absolute_error	int (max_features = 2)	10	0.8547
30	absolute_error	float (max_features = 0.5)	10	0.8415

Criterion: squared_error
n_estimators: 100
max_features: sqrt
R^2 Score: 0.8719

Final Model Selection: *Random Forest Regressor (R^2 Score: 0.8719)*

Compared to Multiple Linear Regression, SVM, and Decision Tree models, the Random Forest Regressor achieved the highest R^2 Score, indicating better prediction of insurance charges.

Reason for selecting Random Forest:

The Random Forest Regressor was selected because it achieved the highest R^2 Score (0.8719), demonstrating superior predictive accuracy compared to other models. Its ensemble approach of combining multiple decision trees effectively reduces variance and minimizes overfitting, making it well-suited for datasets with nonlinear relationships. Additionally, Random Forest performs reliably even in the presence of noisy or moderately imbalanced data. Through hyperparameter tuning of factors such as *n_estimators*, *criterion*, and *max_features*, the model was further optimized to generalize effectively on unseen data.