Artificial Intelligence

Regression Algorithm

Problem Statement:

A client's requirement is; he wants to predict the insurance charges based on the several parameters. The Client has provided the dataset of the same.
As a data scientist, you must develop a model which will predict the insurance charges.

1.) Identify your problem statement:

- **Stage 1: Machine Learning**
  Since all the data points are numerical, the problem falls under the domain of **Machine Learning**.
- **Stage 2: Supervised Learning**
  We have clearly provided both input and output data for the model to learn from, which categorizes this as **Supervised Learning**.
- **Stage 3: Regression**
  As the data consists of continuous numerical values (e.g., 36,788.98; 106,029.90; etc.), this is specifically a **Regression** problem.

2.) Tell basic info about the dataset (Total number of rows, columns)

The dataset you've provided consists of **1338 rows** and **6 columns**.

- **AGE** – Age of the person (numerical)
- **GENDER** – Gender (categorical: male/female)
- **BMI** – Body Mass Index (numerical)
- **CHILDREN** – Number of children/dependents (numerical)
- **SMOKER** – Whether the person is a smoker (categorical: yes/no)
- **CHARGES** – Medical insurance charges (numerical – this is the **target/output** variable for regression)

3.) Mention the pre-processing method if you're doing any (like converting string to number – nominal data)

GENDER (male/female) and **SMOKER** (yes/no) are nominal.
One HOT Encoding-> {male = 1, female = 0; yes = 1, no = 0}

```
dataset = pd.get_dummies(dataset, columns=['GENDER', 'SMOKER'], dtype=int)
#1338 rows × 8 columns
```

4.) Develop a good model with r2_score. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model.

S.NO	Algorithm	R2 Score	Parameter	Result-Good/Bad	Percentage
1	Linear Algorithm	0.7894790349867009	LinearRegression(fit_intercept=True, copy_X=True, n_jobs=None, positive=False)	Bad	69%
2	SVR	-0.089941217025675 7	kernel='sigmoid'	Bad	40%
3	SVR	-0.071956721897779 69	kernel='rbf', C=100, gamma=0.1, epsilon=0.1	Bad	45%
4	SVR	0.5432818196697926	kernel='linear', C=100	Bad	55%
5	Decision Tree	0.6957078199551677	DecisionTreeRegressor(random_state=0)	Bad	67%
6	Random Forest	0.8542219814642709	RandomForestRegressor(n_estimators=100, random_state=42)	Not Bad	80%
7	Random Forest	0.8685064586450466	RandomForestRegressor(n_estimators=200, min_samples_split=5, random_state=42)	Not Bad	85%

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators=200,  min_samples_split=5,  random_state=42 )
regressor.fit(X_train, y_train)
```

```
C:\Users\User\anaconda3\Lib\site-packages\sklearn\base.py:1473: DataConversionWarning: A colu
en a 1d array was expected. Please change the shape of y to (n_samples,), for example using r
  return fit_method(estimator, *args, **kwargs)
```

[163]:
```
RandomForestRegressor(min_samples_split=5, n_estimators=200, random_state=42)
```

[164]:
```
y_pred=regressor.predict(X_test)
```

[165]:
```
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
```

[166]:
```
r_score
```

[166]:
```
0.8685064586450466
```

5.) Mention your final model, justify why u have chosen the same.

Tried But only get 85% Accuracy.

171]:
```
# Create DataFrame from user input
user_input_df = pd.DataFrame([{
    'AGE': age,
    'BMI': bmi,
    'CHILDREN': children,
    'GENDER_male': gender_male,
    'SMOKER_yes': smoker_yes
}])

# Display the input
print(user_input_df)
```

```
   AGE   BMI  CHILDREN  GENDER_male  SMOKER_yes
0   33  44.8         2            0           0
```

172]:
```
result = loaded_model.predict([[age, bmi, children, gender_male, smoker_yes]])
print("Predicted Insurance Charges: $", result[0])
```

```
Predicted Insurance Charges: $ 7124.220612264528
```