Assignment on Regression Algorithm – Insurance Charges Prediction

Question :

Problem Statement or Requirement:

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
2. Tell basic info about the dataset (Total number of rows, columns)
3. Mention the pre-processing method if you're doing any (like converting string to number – nominal data)
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.
5. All the research values (r2_score of the models) should be documented. (You can make tabulation or screenshot of the results.)
6. Mention your final model, justify why u have chosen the same.

Solution:

1. **Problem Statement Identification** : Predicting Insurance Charges.

   3 Stages of Problem Identification :

   Stage 1 : ML
   Stage 2 : Supervised Learning
   Stage 3 : Regression

2. **Dataset Basic Info** : 1338 rows × 6 columns

3. **Pre-Processing Method** : One Hot Encoding

4. **Good model with r2_score** : Algorithms Insurance Prediction.zip

5. **Research on r2_score of the models** :

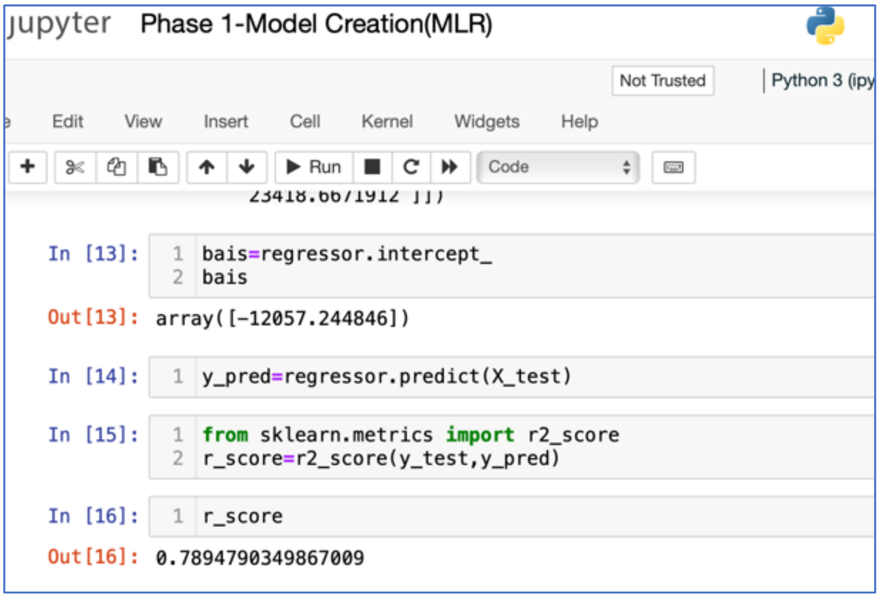
Research on Best Model Via R Score Value

1.MLR

Inference :

Best Model is created with r score Value = **0.7894790349867009**

Proof:



```
23418.6671912 ]])
```

In [13]:
```
bais=regressor.intercept_
bais
```

Out[13]:
```
array([-12057.244846])
```

In [14]:
```
y_pred=regressor.predict(X_test)
```

In [15]:
```
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
```

In [16]:
```
r_score
```

Out[16]:
```
0.7894790349867009
```

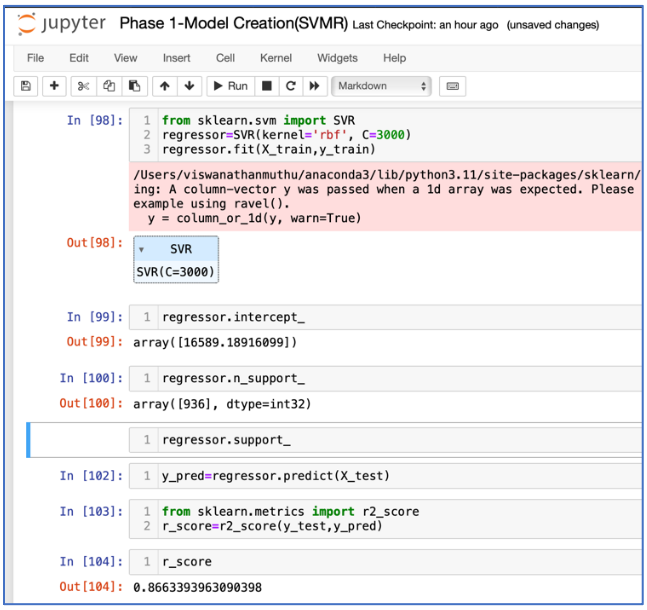
2.SVMR

r score Value w/o HTP= −0.08338238593619329		
kernel	**C Value**	**r score**
linear	1000	0.7649311738597033
	2000	0.7440418308108018
	3000	0.7414236599249162
poly	1000	0.856648767594656
	2000	0.8605579258597715
	3000	0.8598930084494385
rbf	1000	0.8102064874808204
	2000	0.8547766422240716
	3000	0.8663393963090398
sigmoid	1000	0.2874706948697654
	2000	-0.5939509731283503
	3000	-2.1244194786689863

Inference :

Best model with r score value = **0.8663393963090398** is created for the H.T.P --- **kernel = rbf, c= 3000.**

Proof:


jupyter
Phase 1-Model Creation(SVMR) Last Checkpoint: an hour ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Markdown

```
In [98]:  from sklearn.svm import SVR
          regressor=SVR(kernel='rbf', C=3000)
          regressor.fit(X_train,y_train)

          /Users/viswanathanmuthu/anaconda3/lib/python3.11/site-packages/sklearn/
          ing: A column-vector y was passed when a 1d array was expected. Please
          example using ravel().
            y = column_or_1d(y, warn=True)

Out[98]:  SVR
          SVR(C=3000)

In [99]:  regressor.intercept_
Out[99]:  array([16589.18916099])

In [100]: regressor.n_support_
Out[100]: array([936], dtype=int32)

          regressor.support_

In [102]: y_pred=regressor.predict(X_test)

In [103]: from sklearn.metrics import r2_score
          r_score=r2_score(y_test,y_pred)

In [104]: r_score
Out[104]: 0.8663393963090398
```

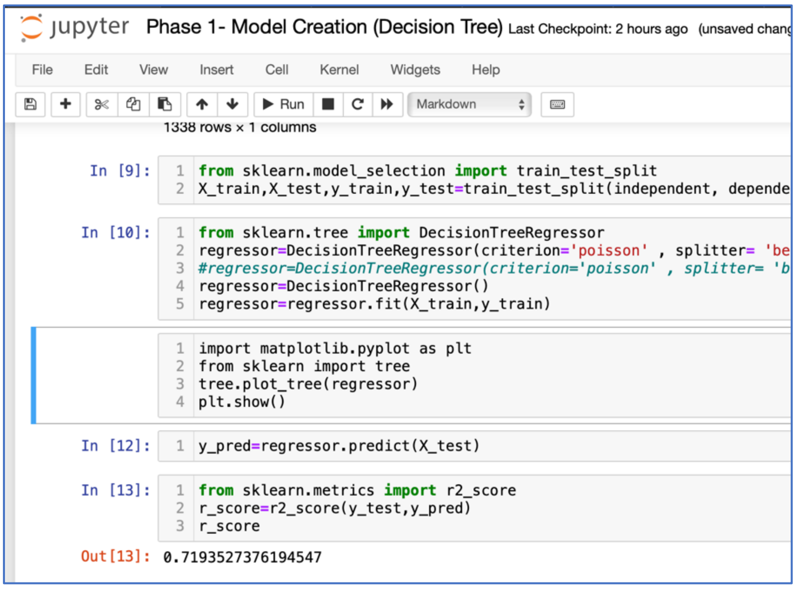
3.Decision Tree

r score Value w/o HTP= 0.6966581868843034				
criterion	**splitter**	**r score (w/o max features)**	**max_ features**	**r score (with max_ features)**
squared_error	best	0.686215300008399	sqrt	0.6834198870363453
			log2	0.7025375319783884
	random	0.7066187403980948	sqrt	0.6953161555966039
			log2	0.6893398873680485
absolute_error	best	0.6983683936823608	sqrt	0.6861931160079187
			log2	0.6911740381386748
	random	0.6834749673466649	sqrt	0.6890368691000321
			log2	0.6908542771938115
friedman_mse	best	0.6865081878698951	sqrt	0.7046724095212571
			log2	0.6814302611125168
	random	0.7003007945640454	sqrt	0.68036570083318
			log2	0.681959664205278
poisson	best	0.7137637844731028	sqrt	0.6911692507256455
			log2	0.6899048836268635
	random	0.6894199406241438	sqrt	0.6900364781583089
			log2	0.6971256952095196

Inference :

Best model with r score Value = **0.7137637844731028** is created for the HTP --- **criterion = poisson & splitter= best** without max features

Proof:



jupyter
Phase 1- Model Creation (Decision Tree) Last Checkpoint: 2 hours ago (unsaved chang
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Markdown

1338 rows × 1 columns

```
In [9]:  from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test=train_test_split(independent, depende

In [10]: from sklearn.tree import DecisionTreeRegressor
         regressor=DecisionTreeRegressor(criterion='poisson' , splitter= 'be
         #regressor=DecisionTreeRegressor(criterion='poisson' , splitter= 'b
         regressor=DecisionTreeRegressor()
         regressor=regressor.fit(X_train,y_train)

         import matplotlib.pyplot as plt
         from sklearn import tree
         tree.plot_tree(regressor)
         plt.show()

In [12]: y_pred=regressor.predict(X_test)

In [13]: from sklearn.metrics import r2_score
         r_score=r2_score(y_test,y_pred)
         r_score

Out[13]: 0.7193527376194547
```

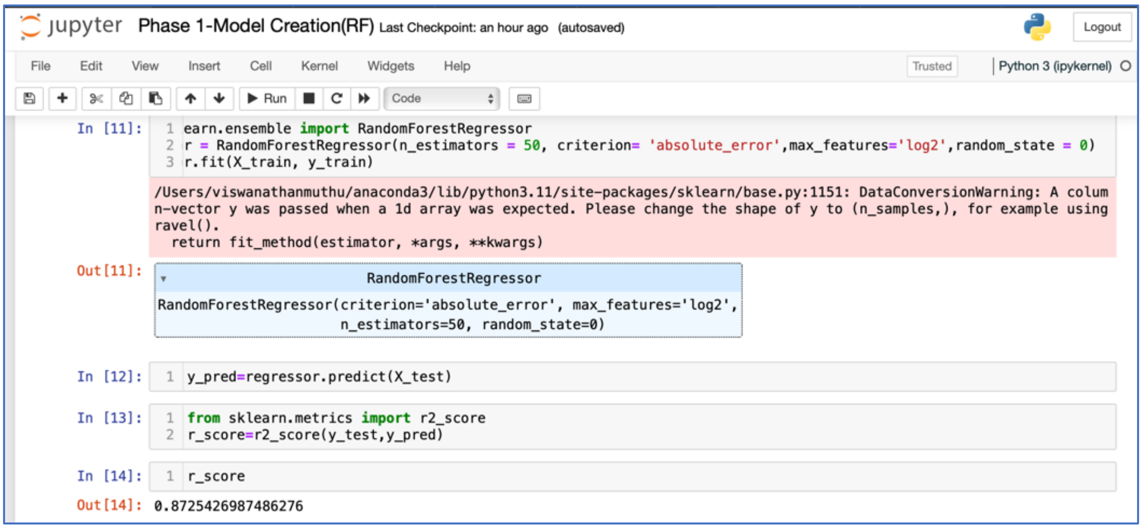
4.Random Forest

criterion	n estimators	r score (w/o max features)	max_ features	r score (with max features)
squared_error	50	0.8537074492312178	sqrt log2	0.8699196004695238
	100	0.8495860472309916	sqrt log2	0.8712882947395911
absolute_error	50	0.8533104199010396	sqrt log2	0.8725426987486276
	100	0.8522171666048011	sqrt log2	0.8714014632724219
friedman_mse	50	0.8498058213339406	sqrt log2	0.8698363819890867
	100	0.8540807721486975	sqrt log2	0.871314345410434
poisson	50	0.8491113222296434	sqrt log2	0.8635474039861692
	100	0.8526481325996583	sqrt log2	0.8681653187265531

Inference :

Best model with r score Value = **0.8725426987486276** is created for the HTP--- **criterion = absolute error & n estimators = 50** with max features = sqrt & log2

Proof:



```
In [11]: earn.ensemble import RandomForestRegressor
         r = RandomForestRegressor(n_estimators = 50, criterion= 'absolute_error',max_features='log2',random_state = 0)
         r.fit(X_train, y_train)

/Users/viswanathanmuthu/anaconda3/lib/python3.11/site-packages/sklearn/base.py:1151: DataConversionWarning: A colum
n-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using
ravel().
  return fit_method(estimator, *args, **kwargs)

Out[11]:                 RandomForestRegressor
         RandomForestRegressor(criterion='absolute_error', max_features='log2',
                               n_estimators=50, random_state=0)

In [12]: y_pred=regressor.predict(X_test)

In [13]: from sklearn.metrics import r2_score
         r_score=r2_score(y_test,y_pred)

In [14]: r_score

Out[14]: 0.8725426987486276
```

Summary

Algorithm	HTP	Best r score
MLR	-	0.7894790349867009

SVM	kernel = rbf,
	c =3000
		0.8663393963090398

DT	criterion = poisson & splitter = best	0.7137637844731028
RF	criterion = absolute error , n estimators = 50, max features = sqrt & log2	0.8725426987486276

Result Analysis:

For the given dataset **RF algorithm for HTP criterion = absolute & n estimators = 50** suits the best with a maximum **r score value = 0.8725426987486276** when compared to the models created by other algorithms.

Appendix:

Abbreviations	Expansion
MLR	Multiple Linear Regression
SVM	Support Vector Machine
DT	Decision Tree
RF	Random Forest
HTP	Hyper Tuning Parameters

6. Final Model :

For the given dataset **RF algorithm for HTP criterion = absolute & n estimators = 50** suits the best with a maximum **r score value = 0.8725426987486276** when compared to the models created by other algorithms.