

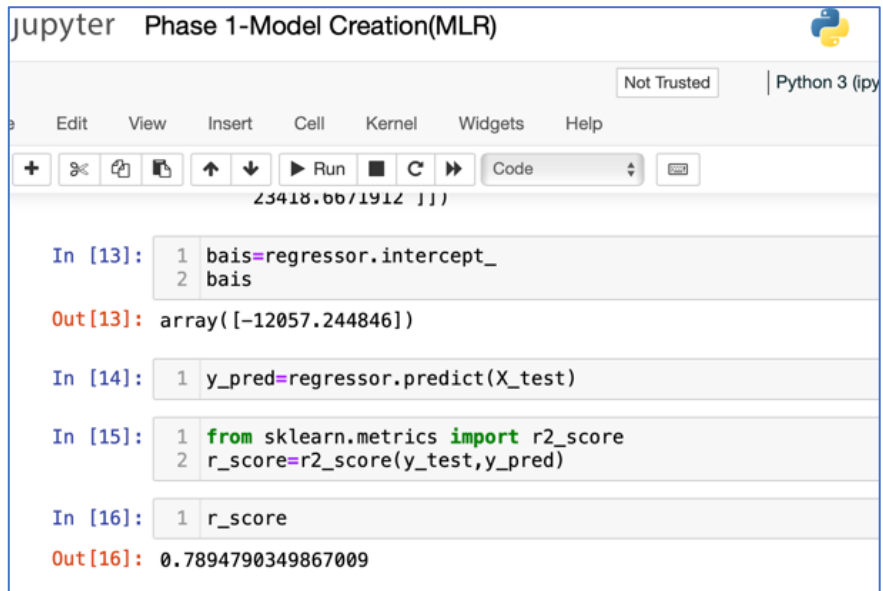
# Research on Best Model Via R Score Value

## 1.MLR

### Inference :

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

### Proof:



The image shows a Jupyter Notebook interface titled "Phase 1-Model Creation(MLR)". The notebook is running on Python 3 (ipykernel) and is marked as "Not Trusted". The interface includes a menu bar with options: Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar is a toolbar with icons for adding new cells, undo, redo, and running code. The notebook contains four code cells:

```
In [13]: 1 bais=regressor.intercept_
          2 bais
Out[13]: array([-12057.244846])

In [14]: 1 y_pred=regressor.predict(X_test)

In [15]: 1 from sklearn.metrics import r2_score
          2 r_score=r2_score(y_test,y_pred)

In [16]: 1 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
In [98]: 1 from sklearn.svm import SVR
          2 regressor=SVR(kernel='rbf', C=3000)
          3 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]: 1 regressor.intercept_
Out[99]: array([16589.18916099])

In [100]: 1 regressor.n_support_
Out[100]: array([936], dtype=int32)

1 regressor.support_

In [102]: 1 y_pred=regressor.predict(X_test)

In [103]: 1 from sklearn.metrics import r2_score
          2 r_score=r2_score(y_test,y_pred)

In [104]: 1 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 changes)
File Edit View Insert Cell Kernel Widgets Help
[Icons] Run [Icons] Markdown [Icons]
1338 rows x 1 columns

In [9]: 1 from sklearn.model_selection import train_test_split
        2 X_train,X_test,y_train,y_test=train_test_split(independent, dependent)

In [10]: 1 from sklearn.tree import DecisionTreeRegressor
        2 regressor=DecisionTreeRegressor(criterion='poisson', splitter='best')
        3 #regressor=DecisionTreeRegressor(criterion='poisson', splitter='best')
        4 regressor=DecisionTreeRegressor()
        5 regressor=regressor.fit(X_train,y_train)

        1 import matplotlib.pyplot as plt
        2 from sklearn import tree
        3 tree.plot_tree(regressor)
        4 plt.show()

In [12]: 1 y_pred=regressor.predict(X_test)

In [13]: 1 from sklearn.metrics import r2_score
        2 r_score=r2_score(y_test,y_pred)
        3 r_score

Out[13]: 0.7193527376194547

```

## 4.Random Forest

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

### Inference :

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

### Proof:

```

jupyter Phase 1-Model Creation(RF) Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) O

In [11]: 1 from sklearn.ensemble import RandomForestRegressor
          2 r = RandomForestRegressor(n_estimators = 50, criterion='absolute_error',max_features='log2',random_state = 0)
          3 r.fit(X_train, y_train)

/Users/viswanathanmuthu/anaconda3/lib/python3.11/site-packages/sklearn/base.py:1151: DataConversionWarning: A column
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]: 1 y_pred=r.predict(X_test)

In [13]: 1 from sklearn.metrics import r2_score
          2 r_score=r2_score(y_test,y_pred)

In [14]: 1 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
<b>RF</b>	<b>criterion = absolute error , n estimators = 50, max features = sqrt &amp; log2</b>	<b>0.8725426987486276</b>

### **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:**

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