Course: MSc. IN BUSINESS ANALYTICS

Module: Applied Statistics &Machine Learning - B9BA102

Submitted to:

Mr. Kunwar Madan

Submitted By:

Abhinav Singh Voria..10634732

Rohit Verma...............10635420

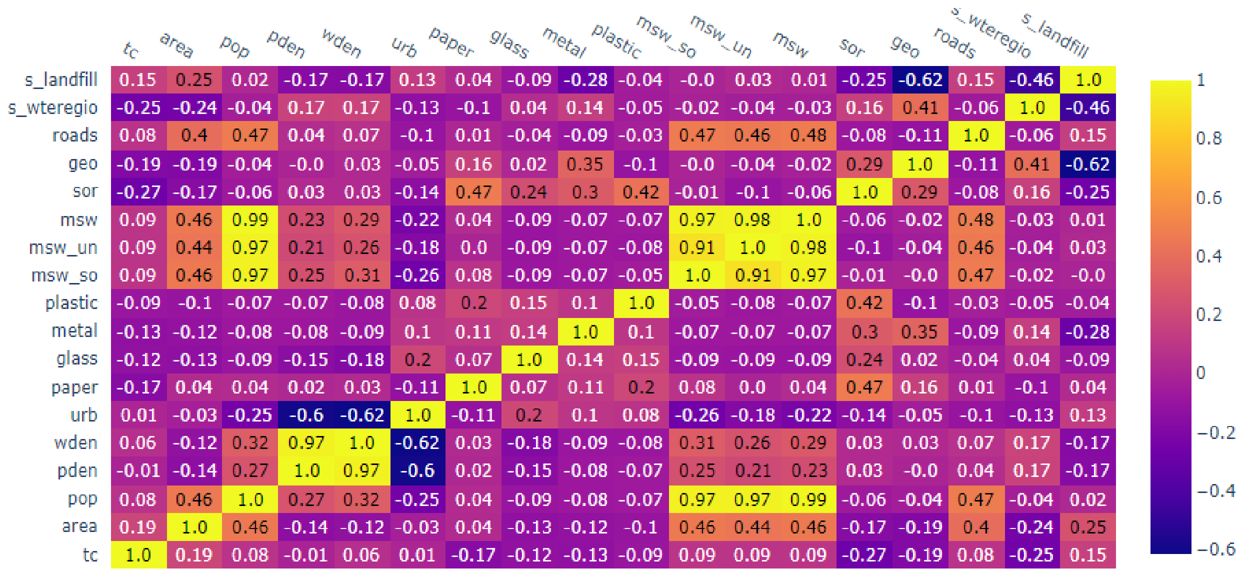
Robin Verma..............10631497

1. TARGET VARIABLE:

tc: Cost of the per-capita cost of waste management.

We have now imported the dataset using costprediction.csv. The dataset is used to display the top 5 rows. The dataset.shape function and head function are used to show the total number of rows and columns, respectively. Any null or missing values from the dataset are shown in info.statistical output is described using dataset.describe.

2. PLOTTING CORRELATION HEAT MAP:



	tc	area	pop	pden	wden	urb	paper	glass	metal	plastic	msw_so	msw_un	msw	sor	geo	roads	s_wteregio	s_landfill
s_landfill	0.15	0.25	0.02	-0.17	-0.17	0.13	0.04	-0.09	-0.28	-0.04	-0.0	0.03	0.01	-0.25	-0.62	0.15	-0.46	1.0
s_wteregio	-0.25	-0.24	-0.04	0.17	0.17	-0.13	-0.1	0.04	0.14	-0.05	-0.02	-0.04	-0.03	0.16	0.41	-0.06	1.0	-0.46
roads	0.08	0.4	0.47	0.04	0.07	-0.1	0.01	-0.04	-0.09	-0.03	0.47	0.46	0.48	-0.08	-0.11	1.0	-0.06	0.15
geo	-0.19	-0.19	-0.04	-0.0	0.03	-0.05	0.16	0.02	0.35	-0.1	-0.0	-0.04	-0.02	0.29	1.0	-0.11	0.41	-0.62
sor	-0.27	-0.17	-0.06	0.03	0.03	-0.14	0.47	0.24	0.3	0.42	-0.01	-0.1	-0.06	1.0	0.29	-0.08	0.16	-0.25
msw	0.09	0.46	0.99	0.23	0.29	-0.22	0.04	-0.09	-0.07	-0.07	0.97	0.98	1.0	-0.06	-0.02	0.48	-0.03	0.01
msw_un	0.09	0.44	0.97	0.21	0.26	-0.18	0.0	-0.09	-0.07	-0.08	0.91	1.0	0.98	-0.1	-0.04	0.46	-0.04	0.03
msw_so	0.09	0.46	0.97	0.25	0.31	-0.26	0.08	-0.09	-0.07	-0.05	1.0	0.91	0.97	-0.01	-0.0	0.47	-0.02	-0.0
plastic	-0.09	-0.1	-0.07	-0.07	-0.08	0.08	0.2	0.15	0.1	1.0	-0.05	-0.08	-0.07	0.42	-0.1	-0.03	-0.05	-0.04
metal	-0.13	-0.12	-0.08	-0.08	-0.09	0.1	0.11	0.14	1.0	0.1	-0.07	-0.07	-0.07	0.3	0.35	-0.09	0.14	-0.28
glass	-0.12	-0.13	-0.09	-0.15	-0.18	0.2	0.07	1.0	0.14	0.15	-0.09	-0.09	-0.09	0.24	0.02	-0.04	0.04	-0.09
paper	-0.17	0.04	0.04	0.02	0.03	-0.11	1.0	0.07	0.11	0.2	0.08	0.0	0.04	0.47	0.16	0.01	-0.1	0.04
urb	0.01	-0.03	-0.25	-0.6	-0.62	1.0	-0.11	0.2	0.1	0.08	-0.26	-0.18	-0.22	-0.14	-0.05	-0.1	-0.13	0.13
wden	0.06	-0.12	0.32	0.97	1.0	-0.62	0.03	-0.18	-0.09	-0.08	0.31	0.26	0.29	0.03	0.03	0.07	0.17	-0.17
pden	-0.01	-0.14	0.27	1.0	0.97	-0.6	0.02	-0.15	-0.08	-0.07	0.25	0.21	0.23	0.03	-0.0	0.04	0.17	-0.17
pop	0.08	0.46	1.0	0.27	0.32	-0.25	0.04	-0.09	-0.08	-0.07	0.97	0.97	0.99	-0.06	-0.04	0.47	-0.04	0.02
area	0.19	1.0	0.46	-0.14	-0.12	-0.03	0.04	-0.13	-0.12	-0.1	0.46	0.44	0.46	-0.17	-0.19	0.4	-0.24	0.25
tc	1.0	0.19	0.08	-0.01	0.06	0.01	-0.17	-0.12	-0.13	-0.09	0.09	0.09	0.09	-0.27	-0.19	0.08	-0.25	0.15

3. DIVIDING DATASET INTO FEATURES AND LABELS:

To separate the target column and associated variables from independent variables, use the "drop" strategy. Since, we have msw data, which is a combination of both msw_un and msw_so, we decided to drop msw_un and msw_so.

Since 'tc' serves as both the label and the target variable in this situation, we should eliminate it. The feature list will still include the target variable but not the other variables.

Scaling down characteristics to make them similar in size is what normalization implies. The model functions better consequently, and training stability is maintained.

4. IMPLEMENTING OF LINEAR REGRESSION:

Q1. Impact of L1, L2, and elastic net regularization on linear regression coefficients, performance, and interpretability. (Note - models built with fewer variables are considered more interpretable)

4.1 Linear Regression without Regularization

Tuning the SGD Regressor parameters 'eta0' (learning rate) and 'max_iter', along with the regularization parameter alpha using Grid Search.

By keeping the penalty at zero and removing the L1 ratio and Alpha from the linear regression without regression model, we were able to get a score of 0.14.

4.2 Linear regression with regularization using elastic net

When we used elastic net regression to apply linear regression, preserving the L1 ratio between 0 and 0.25, 0.5 and 0.75, and keeping the elastic net penalty, we discovered that the results were identical to those obtained without regularization. We obtained a positive r2 score of 0.1443 and an adjusted r2 of 0.1395. Additionally, we tried removing alternative attributes to see if we were receiving a higher score.

By removing alternative features, we were able to achieve the highest score of 0.14, but this was insufficient because it did not well fit the model.

4.3 Linear Regression using L1(Lasso Regression)

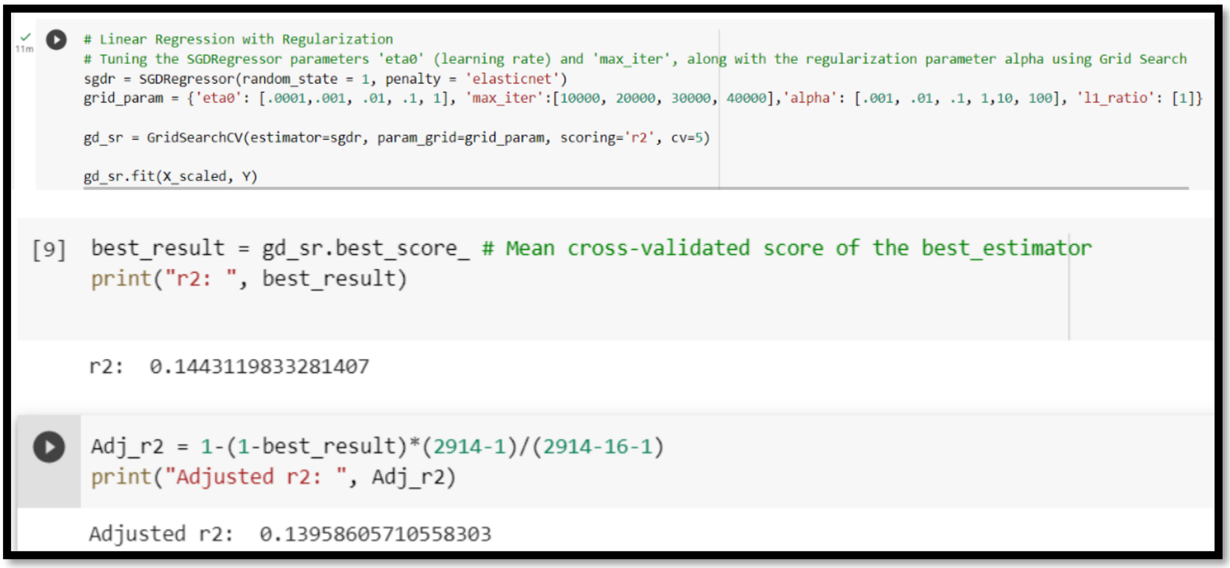
We attempted to preserve L1 ratio 1, which will act as L1 regression entirely and penalize as elastic net, as we went on to the succeeding regression. Both the positive r2 score and the adjusted r2 score we received, 0.1443 and 0.1443, are very low numbers that clearly underfit the model.

4.4 Linear regression using L2(Ridge Regression)

When we use L2 regression and leave the L1 ratio and penalty elastic net both at 0, the model completely switches to L2 regression. We acquired r2 values of 0.1443 and adjusted r2 as 0.1395. These results showed that the score is steadily declining. Because it won't fit well, this score likewise doesn't seem to be a good one.

4.5 Conclusion (Regularization)

Regularization does not enhance the quality of a linear model and yields results that are identical to those of a horizontal hyperplane and insufficient for making accurate predictions.



```
# Linear Regression with Regularization
# Tuning the SGDRegressor parameters 'eta0' (learning rate) and 'max_iter', along with the regularization parameter alpha using Grid Search
sgdr = SGDRegressor(random_state = 1, penalty = 'elasticnet')
grid_param = {'eta0': [.0001,.001, .01, .1, 1], 'max_iter':[10000, 20000, 30000, 40000],'alpha': [.001, .01, .1, 1,10, 100], 'l1_ratio': [1]}

gd_sr = GridSearchCV(estimator=sgdr, param_grid=grid_param, scoring='r2', cv=5)

gd_sr.fit(X_scaled, Y)
```

```
[9] best_result = gd_sr.best_score_ # Mean cross-validated score of the best_estimator
    print("r2: ", best_result)
```

```
r2:  0.1443119833281407
```

```
Adj_r2 = 1-(1-best_result)*(2914-1)/(2914-16-1)
print("Adjusted r2: ", Adj_r2)
```

```
Adjusted r2:  0.13958605710558303
```

5. SUPPORT VECTOR REGRESSION:

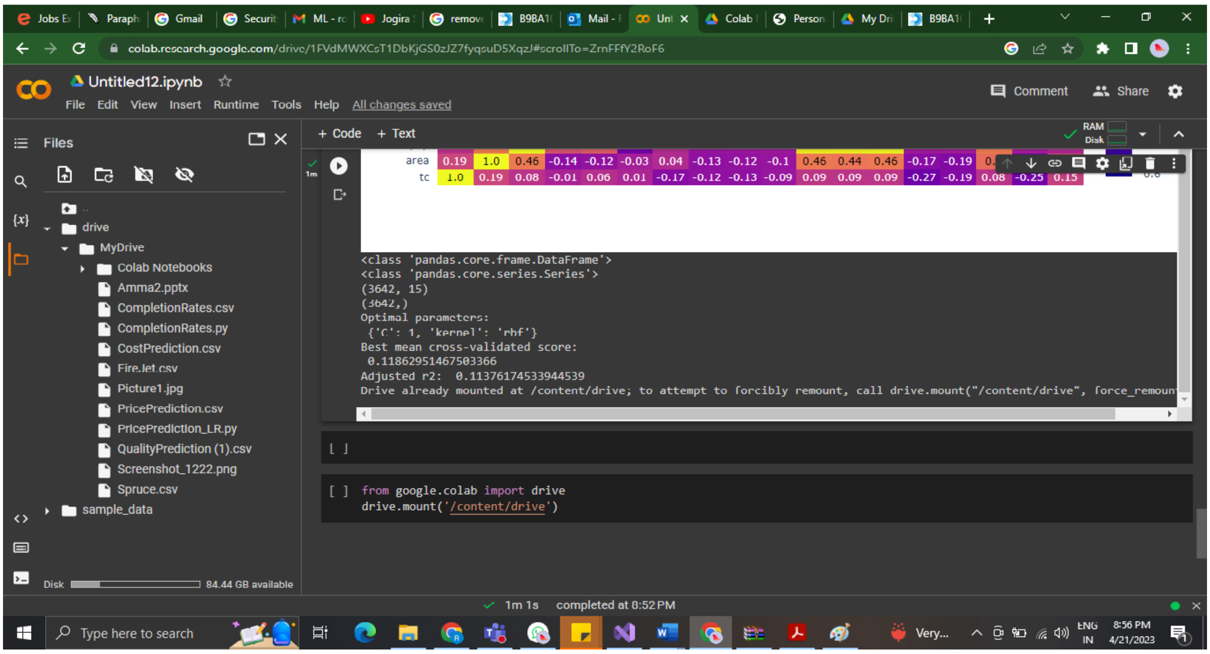
Q2. Impact of L2 regularization on support vector regression performance and interpretability

The model's performance remains equal as evidenced by its negative R2 score of -0.005562 and alpha(C=1000) and epsilon values of 100, even after we changed it using Regularization in SVR.

The results of our support vector regression with a high value of c were r2 -3.9012 and an adjusted score of -0.0055, which are not excellent and will not adequately fit the data. In order to see whether we can raise our score, we will experiment with low values of c and epsilon.

We experimented with low C and epsilon values and obtained a positive r2 score of 0.1187 and an adjusted r2 score of 0.1139, but this is still not a good enough score to forecast the model. Therefore, we will do the experiment without the epsilon value to see if we are obtaining a better value.

Without epsilon, the value steadily climbed to 0.2703, while using low values of C (0.001,0.01,0.1,1) resulted in a lower value of 0.1186 and an adjusted r2 of 0. 11376.



area	0.19	1.0	0.46	-0.14	-0.12	-0.03	0.04	-0.13	-0.12	-0.1	0.46	0.44	0.46	-0.17	-0.19	0.	
tc	1.0	0.19	0.08	-0.01	0.06	0.01	-0.17	-0.12	-0.13	-0.09	0.09	0.09	0.09	-0.27	-0.19	0.08	-0.25	0.15

```
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.series.Series'>
(3642, 15)
(3642,)
Optimal parameters:
 {'C': 1, 'kernel': 'rbf'}
Best mean cross-validated score:
 0.11862951467503366
Adjusted r2:  0.11376174533944539
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remoun
```

```
[ ]

[ ] from google.colab import drive
    drive.mount('/content/drive')
```

Q3. If you were to implement random forest regression, then its comparative performance and interpretability with respect to regularized linear regression and regularized support vector regression models.

RANDOM FOREST REGRESSION

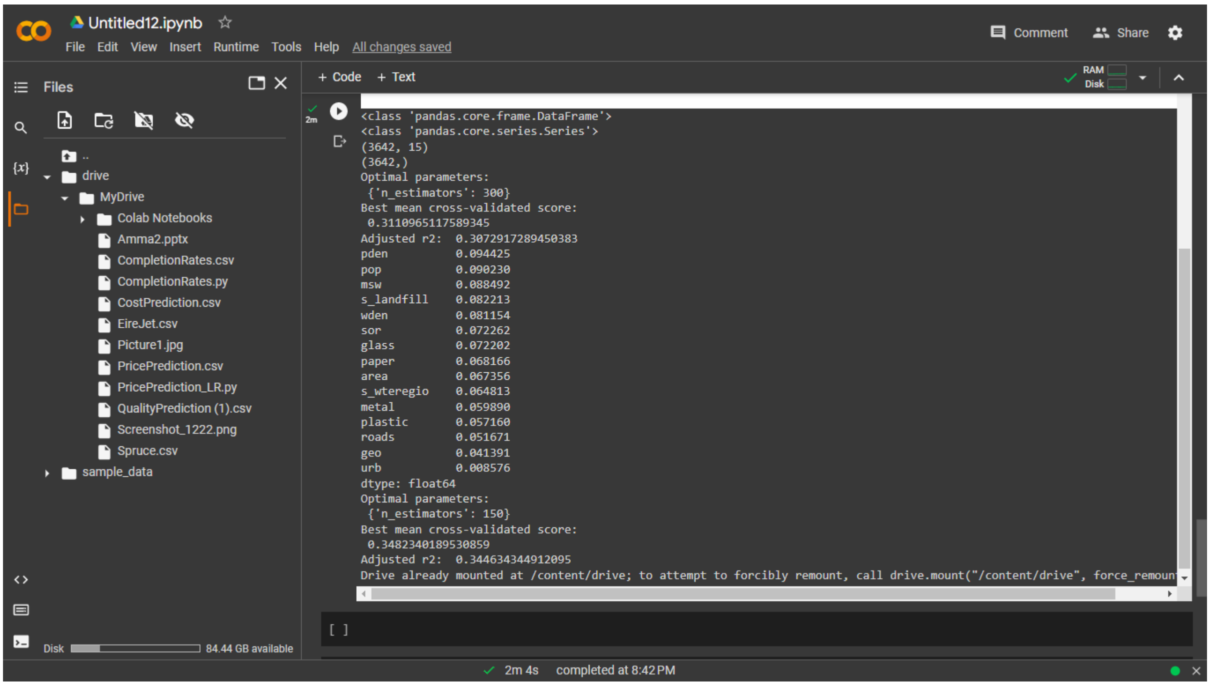
We are changing the Hypermeters (n_estimators) as more decision trees produce more accurate results, which will improve the model's ability to draw the appropriate conclusion.

The Grid Search function is implemented using 5 folds and n_estimators. Cross-validating numerous models with varied numbers of decision trees will yield the best score and the appropriate number of decision trees.

The value steadily increased by comparing Random Forest Regression without significant features and with significant features, but 0.35 is still not a good score to forecast the model, therefore we are trying to adjust the n_estimator value once again to see if we are receiving a higher score.

The value is 0.3511 when compared with n_estimator with 200, which is 0.3482, thus when we tried to modify the n_estimator with 500 we are obtaining the same number but can observe a tiny drop.

To see if we were obtaining better values, we tried removing some of the other properties. After removing msw_so, msw_un, we got a better result from this that is 0.3482, but it is still not a good score to forecast the model.



```
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.series.Series'>
(3642, 15)
(3642,)
Optimal parameters:
 {'n_estimators': 300}
Best mean cross-validated score:
 0.3110965117589345
Adjusted r2:  0.3072917289450383
pden          0.094425
pop           0.090230
msw           0.088492
s_landfill    0.082213
wden          0.081154
sor           0.072262
glass         0.072202
paper         0.068166
area          0.067356
s_wteregio    0.064813
metal         0.059890
plastic       0.057160
roads         0.051671
geo           0.041391
urb           0.008576
dtype: float64
Optimal parameters:
 {'n_estimators': 150}
Best mean cross-validated score:
 0.3482340189530859
Adjusted r2:  0.344634344912095
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remoun
```

```
[ ]
```

2m 4s completed at 8:42 PM

CONCLUSION:

The comparison of the three models—linear regression, random forest, and support vector regression—shows that no model performs well. Only random forest regression performs well when compared to linear regression (LR) and support vector regression. Although we got a better value in Random Forest regression, it is not enough to predict the model, so we conclude that the dataset is underfitting. The best value we obtained from Linear Regression is 0.1443, the best value we obtained from Random Forest regression is 0.3511, and the best value we obtained from Support Vector Regression is 0.2703.

Personal Input:

Robin Varma, as a part of a team, utilized various methods, such as elastic net, lasso, and ridge, to conduct linear regression.

Abhinav Singh Voria, as part of the same team, carried out support vector regression, adjusting the value of the L2 regularization.

Rohit Verma, who was also part of the team, performed both feature-based and feature-free Random Forest regression.

After all of these tasks were completed, the team consolidated their work and presented their findings in a group report.