Prediction and Correlation-based Response Time Analysis of StackOverflow data

CS 235 Research Project Implementation Correctness Report

JANNAT ARA MEEM, RITUPARNA GUCHHAIT, SAKIB FUAD, and TOMAL MAJUMDER

1 CORRECTNESS REPORT FOR K-MEANS CLUSTERING IMPLEMENTATION

Cluster Samples from sklearn implementation	Cluster Samples from our implementation
java-8, java-ee, tomcat, swing, javafx, jpa, junit, thymeleaf	java8, java-ee, tomcat, swing, javafx, jpa, junit
pandas, numpy, matplotlib, csv	pandas, numpy, matplotlib, anaconda, csv, macos, vscode
amazon-ec2, amazon-s3, aws-lambda, aws-cli	amazon-ec2, amazon-s3, aws-lambda, aws-cli
node.js, typescript, npm, gulp	node.js, npm, gulp

Table 1. Comparison of sample clusters with library implementation (K-Means)

In case of Silhouette score (table 2) , the greater the score is, the better. On the other hand for David-Bouldin score (table 2), lesser value is better. In regards of these scores both, sklearn implementation of K-means performs better. But our implementation's score is also near the scores of sklearn K-means. That's why our implementation generates almost similar clustering results. Some of the similar clusters found by both K-means implementations are shown in table 1.

Index Scores for sklearn Implementation	Index Scores for our implementation
Silhouette Score: 0.49 Davies-Bouldin Score: 0.45	Silhouette Score: 0.38 Davies-Bouldin Score: 0.66

Table 2. Index Score Comparison for both implementation (Kmeans)

2 CORRECTNESS REPORT FOR HIERARCHIAL CLUSTERING IMPLEMENTATION

Here we can see in table 4 that Scilhouette score of sklearn hierarchial clustering is slightly better than our implementation. In case of Davies-Bouldin score, the differences are also very less. Although skelarn's performance is slightly better, our implementation of hierarchial clustering generates almost similar clusters as sklearn's implementation. Comparison of some of the similar clusters are shown in table 3.

Authors' address: Jannat Ara Meem, jmeem001@ucr.edu; Rituparna Guchhait, rguch001@ucr.edu; Sakib Fuad, sfuad001@ucr.edu; Tomal Majumder, tmaju002@ucr.edu.

Cluster Samples from sklearn implementation	Cluster Samples from our implementation
swift, xcode, ios, objective-c, swift3, iphone, swift4	swift, xcode, ios, objective-c, iphone, realm, xcode6
pandas, numpy, matplotlib, csv	pandas, numpy, matplotlib, csv
android, java, android-studio, kotlin, java-ee java8, tomcat, xml, gradle, maven, android-layout	android, android-studio, kotlin, android-layout, arduino, gradle, android-layout, android-fragments
linux, ubuntu, centos, bash, shellscript, raspberry-pi, raspbian, unix, vagrant, virtualbox, debian, yum	linux, ubuntu, centos, swiftui, unix, bash, shellscript, debian, yum

Table 3. Comparison of sample clusters with library implementation (Hierarchial)

Index Scores for sklearn Implementation	Index Scores for our implementation
Silhouette Score: 0.52 Davies-Bouldin Score: 0.56	Silhouette Score: 0.45 Davies-Bouldin Score: 0.62

Table 4. Index Score Comparison for both implementation (Hierarchial)

3 CORRECTNESS REPORT FOR LINEAR REGRESSION IMPLEMENTATION

We have implemented our proposed Linear Regression model from scratch. There are two hyperparameters: learning rate and maximum number of iterations of our model. We have used **K-fold Cross Validation** method to choose the optimal set of parameters for the model. Next, we have run our implemented Linear Regression and sklearn's LinearRegression library with these parameter values to predict the response time and compared their accurracy metrices. The results are presented as follows:

3.1 Hyperparameter Tuning using K-fold Cross Validation

Table 5 shows the average RMSE score of our model with different hyperparameter setup using sklearn's K-fold Cross Validation library. We can see we have got the lowest RMSE error when learning rate is 1.2 and maximum number of iterations is 400 (red marked).

3.2 Comparing Accuracy Metrics with library implementation

Table 6 shows the accuracy metrics for both sklearn's implementation of LR and our own implementation. We can say with respect accuracy metrics, both implementations have similar results in terms of accuracy. Also in case of run-time both implementation took similar times.

4 HYPERPARAMETER TUNING FOR MLP REGRESSOR

We have used Multilayer Perception Regression model implemented in python's sklearn library in our project as second proposed model to our response time prediction model. The first concern of designing effective regressor is to model the hyperparameters. For that, we have considered some significant parameters: Choice of activation function, maximum

Learning Rate	Average RMSE score for K-folds
0.01	For max_iteration = 200 Average RMSE score = 0.877338 For max_iteration = 300 Average RMSE score = 0.840518 For max_iteration = 400 Average RMSE score = 0.824242 For max_iteration = 500 Average RMSE score = 0.814666
0.5	For max_iteration = 200 Average RMSE score = 0.789034 For max_iteration = 300 Average RMSE score = 0.776489 For max_iteration = 400 Average RMSE score = 0.7698152 For max_iteration = 500 Average RMSE score = 0.7659932
1.0	For max_iteration = 200 Average RMSE score = 0.751711 For max_iteration = 300 Average RMSE score = 0.7504878 For max_iteration = 400 Average RMSE score = 0.749929 For max_iteration = 500 Average RMSE score = 0.749667
1.2	For max_iteration = 200 Average RMSE score = 0.751101 For max_iteration = 300 Average RMSE score = 0.750102 For max_iteration = 400 Average RMSE score = 0.749305 For max_iteration = 500 Average RMSE score = 0.749543
1.5	For max_iteration = 200 Average RMSE score = 0.750485 For max_iteration = 300 Average RMSE score = 0.749773 For max_iteration = 400 Average RMSE score = 0.749543 For max_iteration = 500 Average RMSE score = 0.749467

Table 5. Results of K-fold Cross Validation for Linear Regressor

Accuracy Metrics for sklearn's LR	Accuracy Metrices for scratch LR
Mean Absolute Error: 0.679739 Mean Squared Score: 0.654944 Root Mean Squared Error: 0.8092866	Mean Absolute Error: 0.714045 Mean Squared Score: 0.7591796 Root Mean Squared Error: 0.8713091

Table 6. Accuracy Metrics Comparison of scratch and sklearn's implementation of Linear Regression

number of iterations, and size of the hidden layers. We used K-fold cross validation on our training set, and reported average R2 score for each setting. The higher the R2 score, the better the performance of the model. The average score of k-folds (here k = 10) of each hyperparameter setting is reported in table 7.

Observation: From table 7, we get the higher R2 score when activation is "relu", hidden layer size is (250,) and maximum iteration number is 300 (red marked). We have used this parameter setting to our final report to get the prediction results and accuracy metrics for MLPRegressor.

Additional Fine-Tuning: We have also fine tuned two additional parameters: early_stopping and validation_fraction. We have taken the best parameter setting from table **??** and then first set the early_stopping value to *False*. We also experimented with early_stopping value being *TRUE* and set different validation_fraction values to observe the average R2 score of K-folds. Here K has been assumed to be 10. The result of each hyperparameter setting has been reported at 8. We can see the optimal setting would be when early_stopping is *TRUE* and validation_fraction is 0.15 (red marked)

Activation	Average R2 score for K-folds for different parameters
relu	For hidden size = (100,) max_iteration = 200 Average score = 0.0378042 For hidden size = (100,) max_iteration = 300 Average score = 0.0365092 For hidden size = (100,) max_iteration = 400 Average score = 0.0369469 For hidden size = (150,) max_iteration = 200 Average score = 0.0379482 For hidden size = (150,) max_iteration = 300 Average score = 0.0376009 For hidden size = (150,) max_iteration = 400 Average score = 0.0375752 For hidden size = (200,) max_iteration = 200 Average score = 0.0367349 For hidden size = (200,) max_iteration = 300 Average score = 0.0367349 For hidden size = (200,) max_iteration = 400 Average score = 0.0369095 For hidden size = (250,) max_iteration = 200 Average score = 0.0377488 For hidden size = (250,) max_iteration = 300 Average score = 0.0380658 For hidden size = (250,) max_iteration = 400 Average score = 0.0365088 For hidden size = (300,) max_iteration = 200 Average score = 0.0369947 For hidden size = (300,) max_iteration = 300 Average score = 0.0365046 For hidden size = (300,) max_iteration = 400 Average score = 0.0372936
tanh	For hidden size = (100,) max_iteration = 200 Average score = 0.035264 For hidden size = (100,) max_iteration = 300 Average score = 0.035645 For hidden size = (100,) max_iteration = 400 Average score = 0.035469 For hidden size = (150,) max_iteration = 200 Average score = 0.036110 For hidden size = (150,) max_iteration = 300 Average score = 0.036241 For hidden size = (150,) max_iteration = 400 Average score = 0.035313 For hidden size = (200,) max_iteration = 200 Average score = 0.035371 For hidden size = (200,) max_iteration = 300 Average score = 0.035412 For hidden size = (200,) max_iteration = 400 Average score = 0.034918 For hidden size = (250,) max_iteration = 200 Average score = 0.035009 For hidden size = (250,) max_iteration = 300 Average score = 0.036001 For hidden size = (250,) max_iteration = 400 Average score = 0.035215 For hidden size = (300,) max_iteration = 200 Average score = 0.034996 For hidden size = (300,) max_iteration = 300 Average score = 0.035158 For hidden size = (300,) max_iteration = 400 Average score = 0.034732

Table 7. Results of K-fold Cross Validation for MLPRegressor

Early Stopping	Average R2 score for K-folds
True	For valid_fraction = 0.10 Average R2 score = 0.0350163 For valid_fraction = 0.15 Average R2 score = 0.0385874 For valid_fraction = 0.20 Average R2 score = 0.0385337
False	valid_fraction not needed, Average R2 score = 0.0321421

Table 8. K-fold Cross Validation Report for early stopping (MLPRegressor)