BASIC INFORMATION

Title of Project: KEGG Metabolic Relation Network

Student Name: Aman Gautam

Branch : Artificial Intelligence & Data Science B2-B

Enrolment Number: 00519011922

Email ID: aman.00519011922@ipu.ac.in

Contact Number:8351817505

IMPORTANT LINKS

Git-Hub :

https://github.com/Aman-Gautam1/Kegg-Metabolic-Relation-Network

Website :

https://sites.google.com/view/aman-gautam/home

Google Drive Link:

https://drive.google.com/drive/folders/1H9UsRRvOgtOEBIujrcIYqb4BQ6ZpN3rN?usp=drive_link

KEGG Metabolic Reaction Network

ABSTRACT

The objective of this project is to develop a regression model for predicting the the neighborhood connection based on various network features. The data used for this analysis is sourced from the KEGG database and is represented in the form of a reaction network.

The project begins by importing the necessary libraries and loading the data into a pandas DataFrame. The 'Neighborhood Connectivity' column is identified as the target variable, and the remaining columns are selected as independent variables.

To evaluate the performance of the regression model, the dataset is split into training and testing sets. The scikit-learn library is utilized to perform this data splitting task. The training set is used to train the regression model, and the testing set is used to assess the model's predictive capabilities.

Although the provided code snippet focuses on data preprocessing and data splitting, it serves as an initial step towards building a regression model for predicting metabolic pathway density. Further steps involving the selection of an appropriate regression algorithm, model training, evaluation, and interpretation of the results are not included in the provided code snippet.

The successful development of a regression model for predicting metabolic pathway density can provide valuable insights into the organization and characteristics of these pathways.

KEYWORDS

1. **Regression modeling**: A statistical approach used to establish a relationship between dependent and independent variables. In this project, regression modeling is employed to predict the density of metabolic pathways based on various network features.

2. **Network analysis**: Involves the study of complex systems represented as networks or graphs. In this project, network analysis is used to model metabolic pathways as either reaction networks or relation networks and extract meaningful insights from the data.

3. **KEGG database**: Short for Kyoto Encyclopedia of Genes and Genomes, KEGG is a comprehensive bioinformatics database that provides information on genes, pathways, diseases, and other biological entities. The project utilizes data from the KEGG database to construct and analyze metabolic pathway networks.

4. **NeighborhoodConnectivity prediction**: The aim of this project is to develop a regression model that can predict the neighborhood connection.it provides information about how well connected the nodes are to their immediate neighbor.

5. **Network modeling**: Involves the representation of metabolic pathways as networks or graphs, where compounds and genes are nodes, and reactions or relations are edges. Network modeling allows for the exploration and analysis of the interconnectedness and relationships within metabolic pathways.

INTRODUCTION

The above project aims to develop a regression model to calculate avg no. of connections that tge direct neighbors of a node have The project utilizes data from the KEGG database, which provides information about genes, compounds, and biological pathways.

The target variable in this project is the **NeighborhoodConnectivity**, which represents the information about how well connected the nodes are to their immediate neighbor .

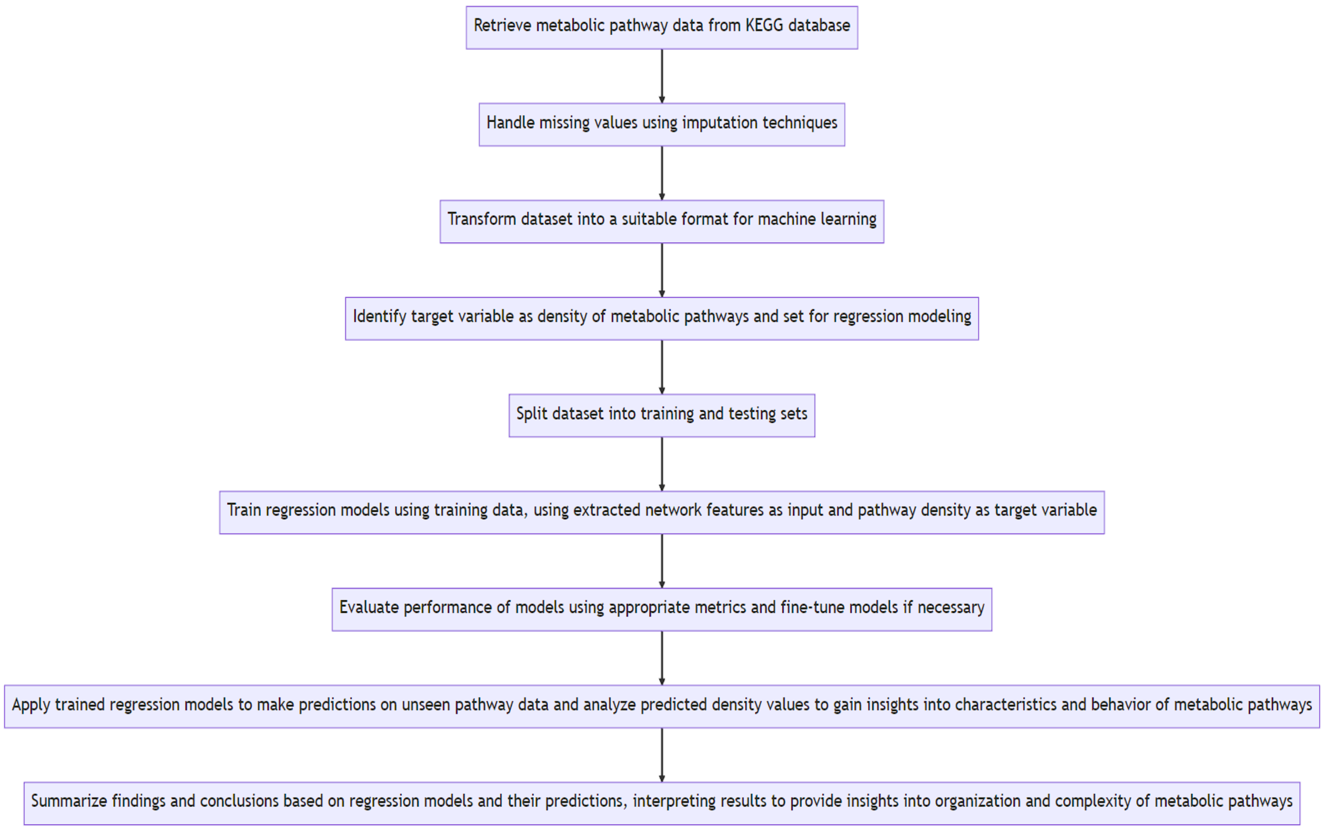
Neighborhood Connectivity can be calculated for each node in the network. It quantifies the average number of connections that the direct neighbors of a node have. This measure helps in understanding the local connectivity patterns within the network and can provide insights into the functional relationships between compounds, enzymes, or genes

The machine learning technique used in this project is regression modeling. Regression models are trained using the network features extracted from the constructed networks. These features include various characteristics such as average number of neighbors, clustering coefficient, betweenness centrality, and others. The regression models learn the relationships between these features and the target variable .

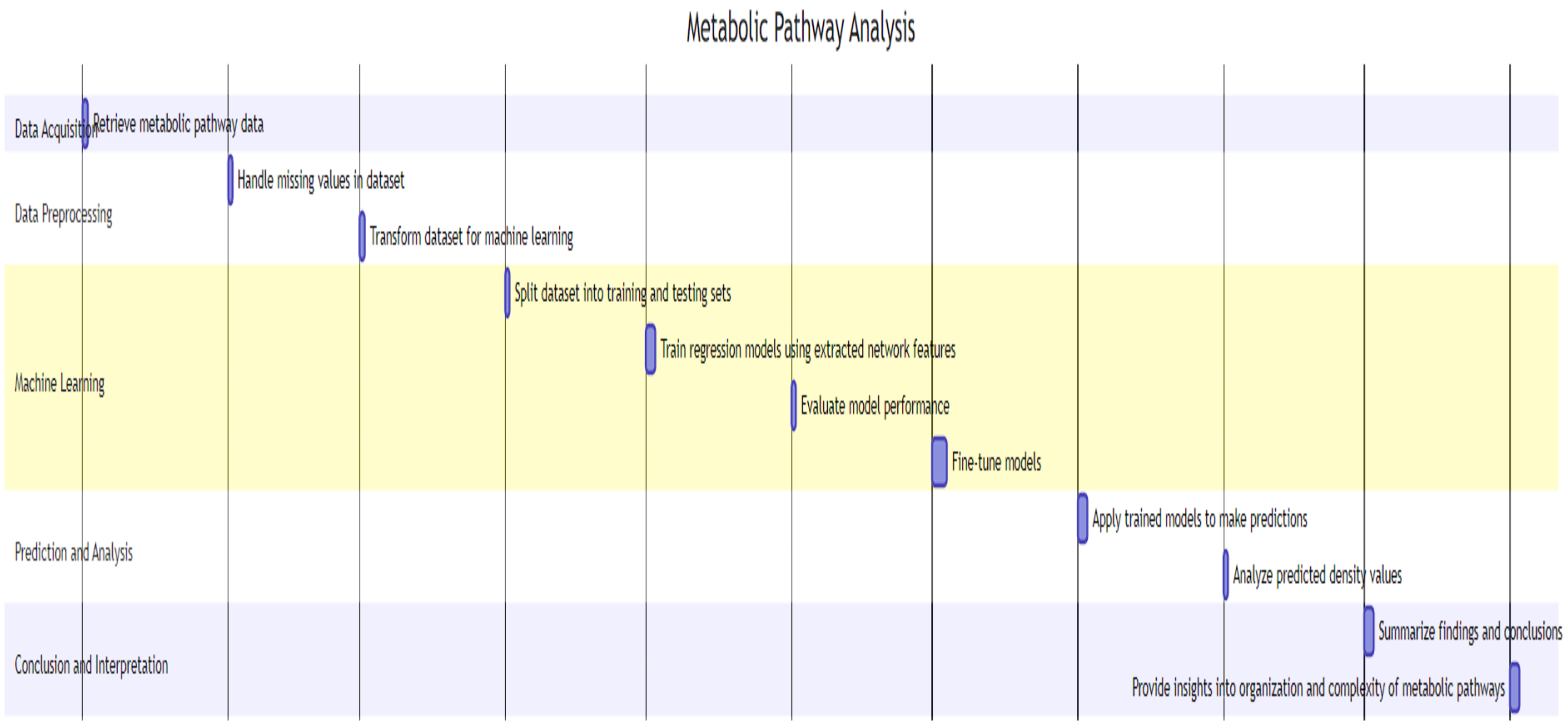
Data preprocessing steps are performed, including handling missing values using imputation techniques. The dataset is split into training and testing sets to evaluate the performance of the regression models. Performance metrics such as mean squared error or R-squared can be used to assess the accuracy of the predictions.

The ultimate objective of this project is to develop a regression model that can effectively predict the density of metabolic pathways. By understanding the factors that contribute to pathway density, researchers can gain insights into the organization and behavior of metabolic networks, which has implications in fields such as systems biology, drug discovery, and metabolic engineering.

Proposed Methodology:-

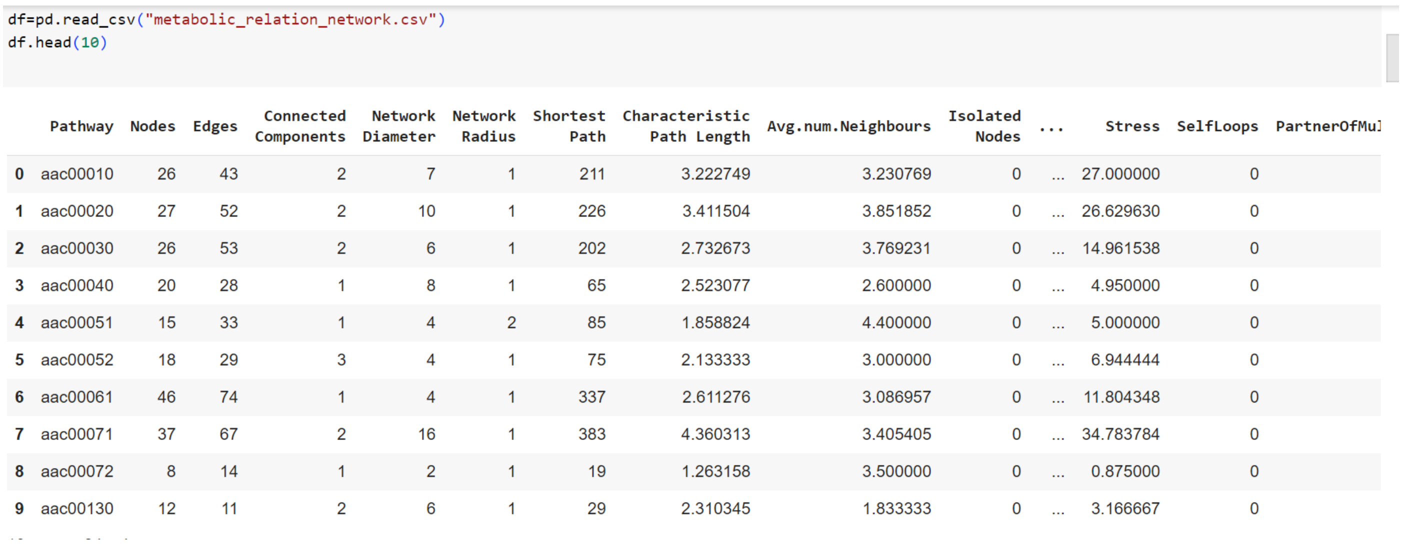


Retrieve metabolic pathway data from KEGG database
Handle missing values using imputation techniques
Transform dataset into a suitable format for machine learning
Identify target variable as density of metabolic pathways and set for regression modeling
Split dataset into training and testing sets
Train regression models using training data, using extracted network features as input and pathway density as target variable
Evaluate performance of models using appropriate metrics and fine-tune models if necessary
Apply trained regression models to make predictions on unseen pathway data and analyze predicted density values to gain insights into characteristics and behavior of metabolic pathways
Summarize findings and conclusions based on regression models and their predictions, interpreting results to provide insights into organization and complexity of metabolic pathways



DataSet:

It is a comprehensive dataset contains 53413 rows and 24 columns that provides information on biological pathways, including metabolic pathways. KEGG Metabolic Pathways is a specific component of the KEGG database that focuses on the metabolic processes occurring within living organisms.

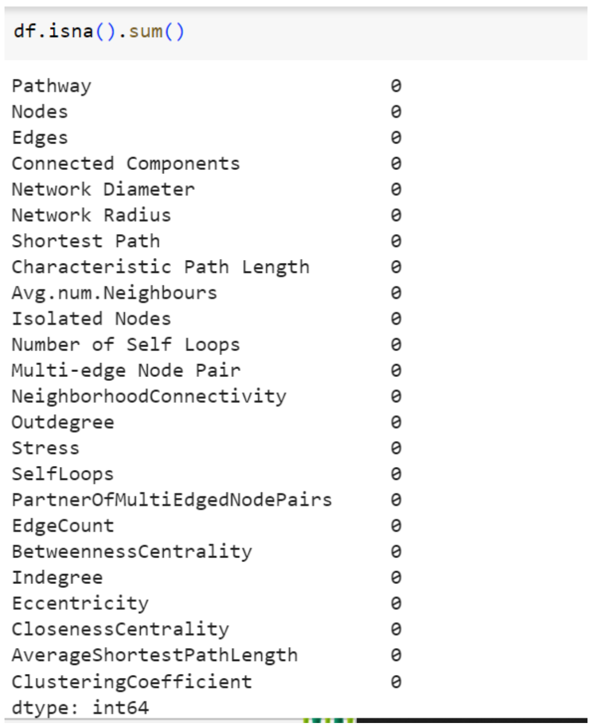


```
df=pd.read_csv("metabolic_relation_network.csv")
df.head(10)
```

```
   Pathway   Nodes  Edges  Connected   Network   Network  Shortest  Characteristic  Avg.num.Neighbours  Isolated  ...  Stress     SelfLoops  PartnerOfMul
                           Components  Diameter  Radius   Path      Path Length                         Nodes
0  aac00010  26     43     2           7         1        211       3.222749        3.230769            0         ...  27.000000  0
1  aac00020  27     52     2           10        1        226       3.411504        3.851852            0         ...  26.629630  0
2  aac00030  26     53     2           6         1        202       2.732673        3.769231            0         ...  14.961538  0
3  aac00040  20     28     1           8         1        65        2.523077        2.600000            0         ...  4.950000   0
4  aac00051  15     33     1           4         2        85        1.858824        4.400000            0         ...  5.000000   0
5  aac00052  18     29     3           4         1        75        2.133333        3.000000            0         ...  6.944444   0
6  aac00061  46     74     1           4         1        337       2.611276        3.086957            0         ...  11.804348  0
7  aac00071  37     67     2           16        1        383       4.360313        3.405405            0         ...  34.783784  0
8  aac00072  8      14     1           2         1        19        1.263158        3.500000            0         ...  0.875000   0
9  aac00130  12     11     2           6         1        29        2.310345        1.833333            0         ...  3.166667   0
```

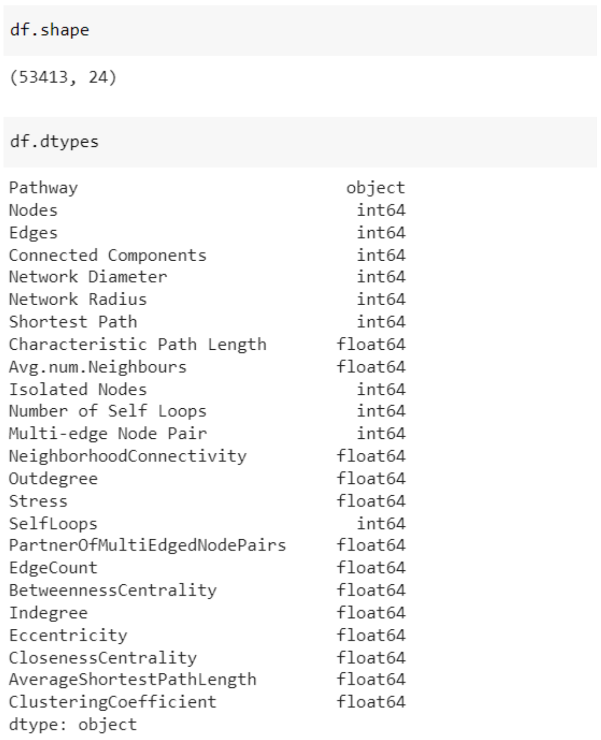
Pre processing:

Checking for null values and remove the alpha numeric (object) values



```
df.isna().sum()
```

```
Pathway                         0
Nodes                           0
Edges                           0
Connected Components            0
Network Diameter                0
Network Radius                  0
Shortest Path                   0
Characteristic Path Length      0
Avg.num.Neighbours              0
Isolated Nodes                  0
Number of Self Loops            0
Multi-edge Node Pair            0
NeighborhoodConnectivity        0
Outdegree                       0
Stress                          0
SelfLoops                       0
PartnerOfMultiEdgedNodePairs    0
EdgeCount                       0
BetweennessCentrality           0
Indegree                        0
Eccentricity                    0
ClosenessCentrality             0
AverageShortestPathLength       0
ClusteringCoefficient           0
dtype: int64
```

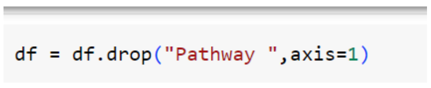


```
df.shape
```

```
(53413, 24)
```

```
df.dtypes
```

```
Pathway                          object
Nodes                             int64
Edges                             int64
Connected Components              int64
Network Diameter                  int64
Network Radius                    int64
Shortest Path                     int64
Characteristic Path Length      float64
Avg.num.Neighbours              float64
Isolated Nodes                    int64
Number of Self Loops              int64
Multi-edge Node Pair              int64
NeighborhoodConnectivity        float64
Outdegree                       float64
Stress                          float64
SelfLoops                         int64
PartnerOfMultiEdgedNodePairs    float64
EdgeCount                       float64
BetweennessCentrality           float64
Indegree                        float64
Eccentricity                    float64
ClosenessCentrality             float64
AverageShortestPathLength       float64
ClusteringCoefficient           float64
dtype: object
```

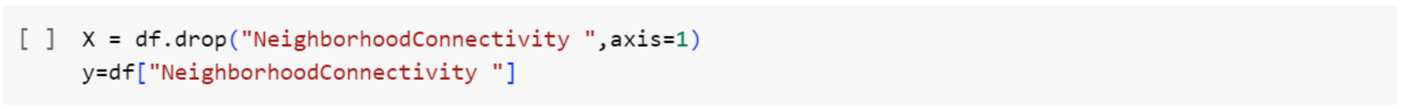


```
df = df.drop("Pathway ",axis=1)
```

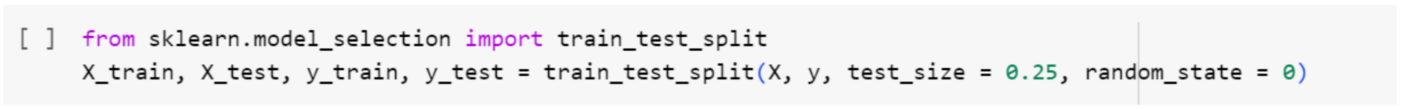
Splitting of the data sets for training and testing:-

- The data is split into training and testing sets using the **train_test_split()** function from sklearn.
- The features in the training and testing sets are standardized using the **StandardScaler()** from sklearn.
- Principal Component Analysis (PCA) is applied to reduce the dimensionality of the data to 15 components.

SPLITTING THE DATASET

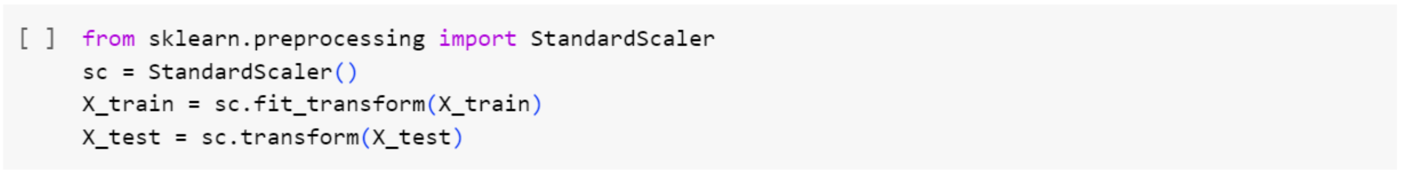


```
[ ]  X = df.drop("NeighborhoodConnectivity ",axis=1)
     y=df["NeighborhoodConnectivity "]
```



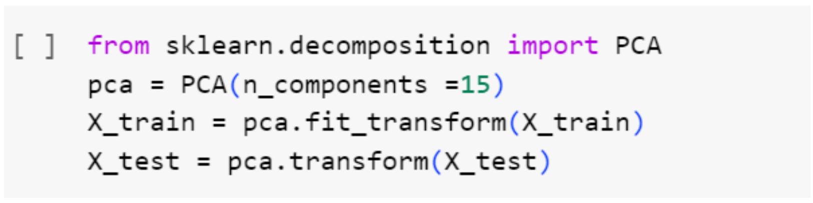
```
[ ]  from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

FEATURE SCALING



```
[ ]  from sklearn.preprocessing import StandardScaler
     sc = StandardScaler()
     X_train = sc.fit_transform(X_train)
     X_test = sc.transform(X_test)
```

APPLYING PCA

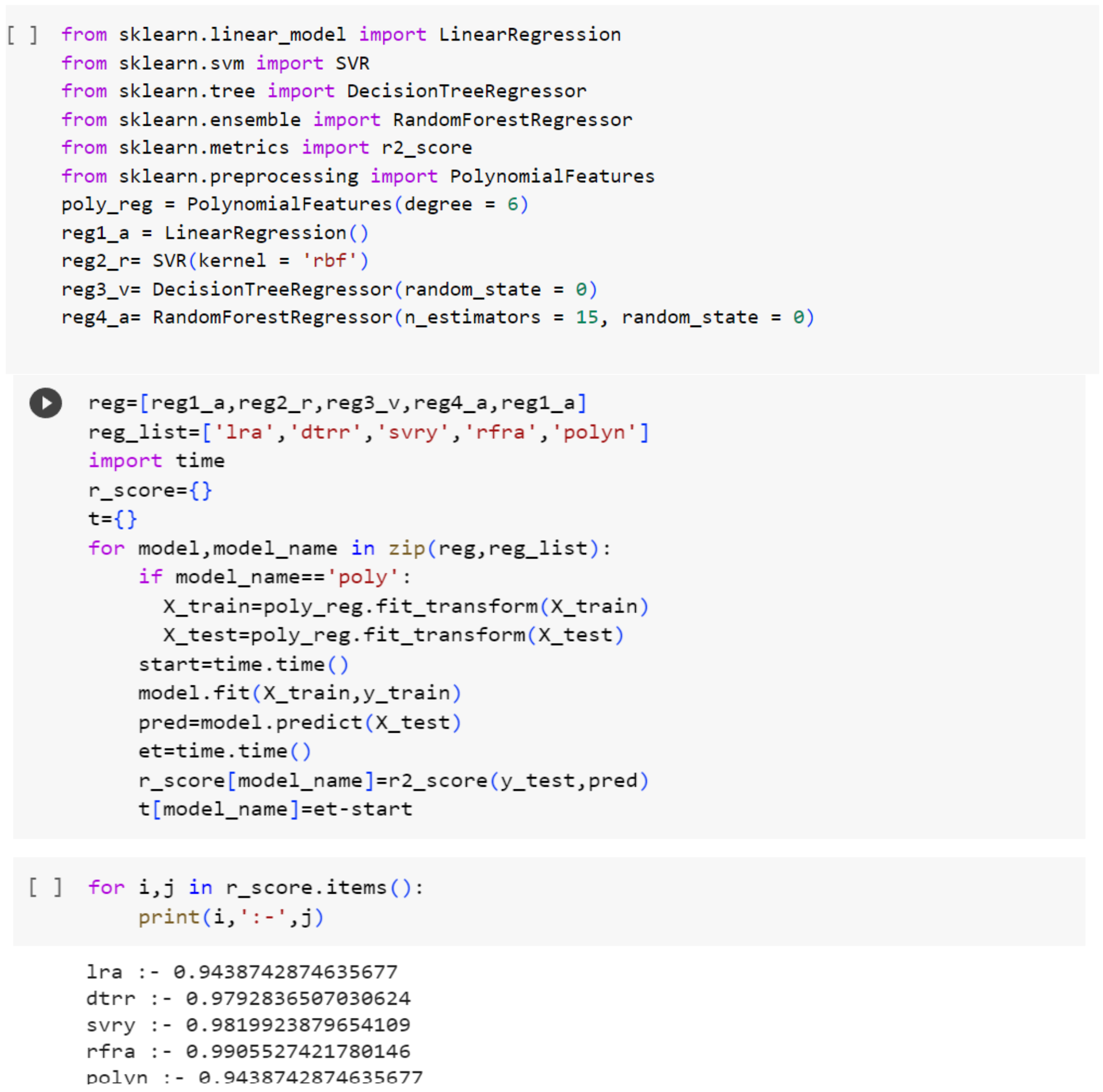


```
[ ]  from sklearn.decomposition import PCA
     pca = PCA(n_components =15)
     X_train = pca.fit_transform(X_train)
     X_test = pca.transform(X_test)
```

Regression:-

- Four regression models, namely Linear Regression, Support Vector Regression (SVR), Decision Tree Regressor, and Random Forest Regressor, are initialized using their respective classes from sklearn.
- A dictionary 'r_score' is created to store the R-squared scores of the models on the test set.
- A dictionary 't' is created to store the time taken by each model for training and prediction.
- For each model, the code fits the model to the training data, makes predictions on the test data, and calculates the R-squared score.
- The R-squared scores and execution times are printed for each model.

REGRESSION



```
[ ]  from sklearn.linear_model import LinearRegression
     from sklearn.svm import SVR
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.metrics import r2_score
     from sklearn.preprocessing import PolynomialFeatures
     poly_reg = PolynomialFeatures(degree = 6)
     reg1_a = LinearRegression()
     reg2_r= SVR(kernel = 'rbf')
     reg3_v= DecisionTreeRegressor(random_state = 0)
     reg4_a= RandomForestRegressor(n_estimators = 15, random_state = 0)
```

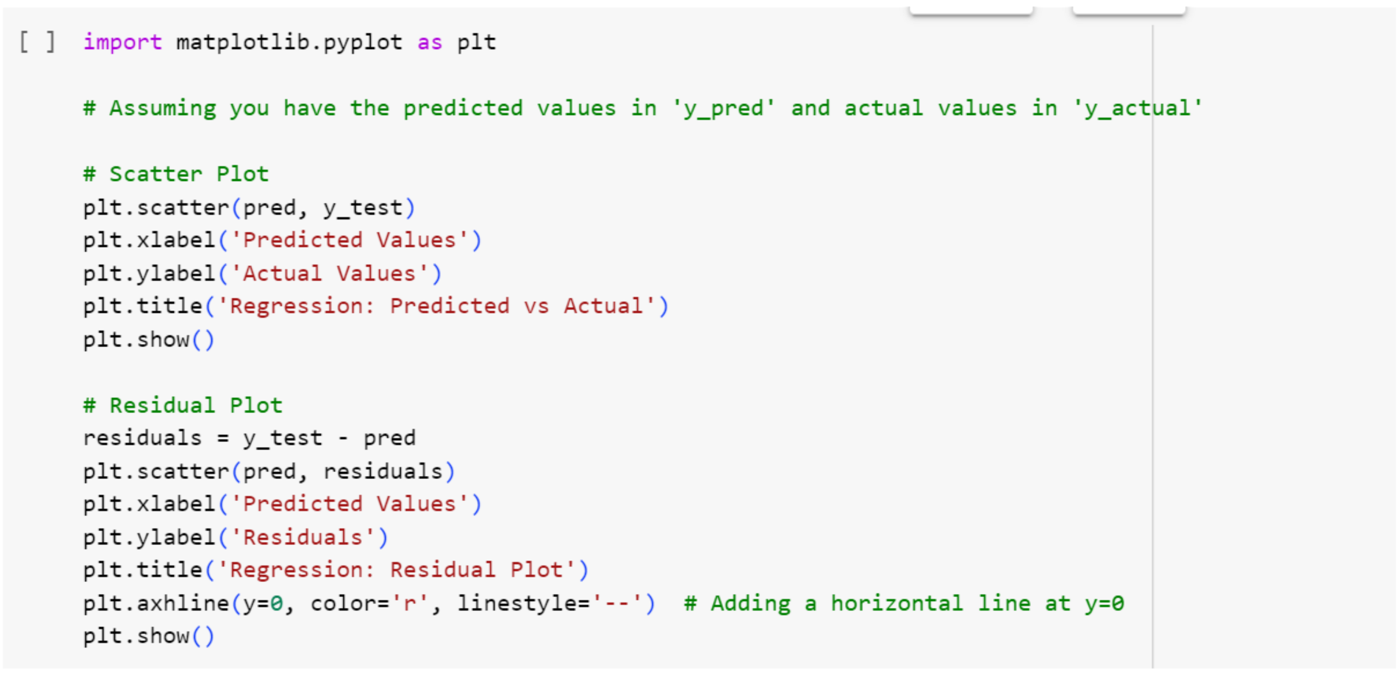
```
reg=[reg1_a,reg2_r,reg3_v,reg4_a,reg1_a]
reg_list=['lra','dtrr','svry','rfra','polyn']
import time
r_score={}
t={}
for model,model_name in zip(reg,reg_list):
    if model_name=='poly':
      X_train=poly_reg.fit_transform(X_train)
      X_test=poly_reg.fit_transform(X_test)
    start=time.time()
    model.fit(X_train,y_train)
    pred=model.predict(X_test)
    et=time.time()
    r_score[model_name]=r2_score(y_test,pred)
    t[model_name]=et-start
```

```
[ ]  for i,j in r_score.items():
         print(i,':-',j)
```

```
lra :- 0.9438742874635677
dtrr :- 0.9792836507030624
svry :- 0.9819923879654109
rfra :- 0.9905527421780146
polyn :- 0.9438742874635677
```

Data Visualization:-

- The code visualizes the predicted values versus the actual values in a scatter plot using matplotlib.
- The code also plots a residual plot, which shows the difference between the actual and predicted values

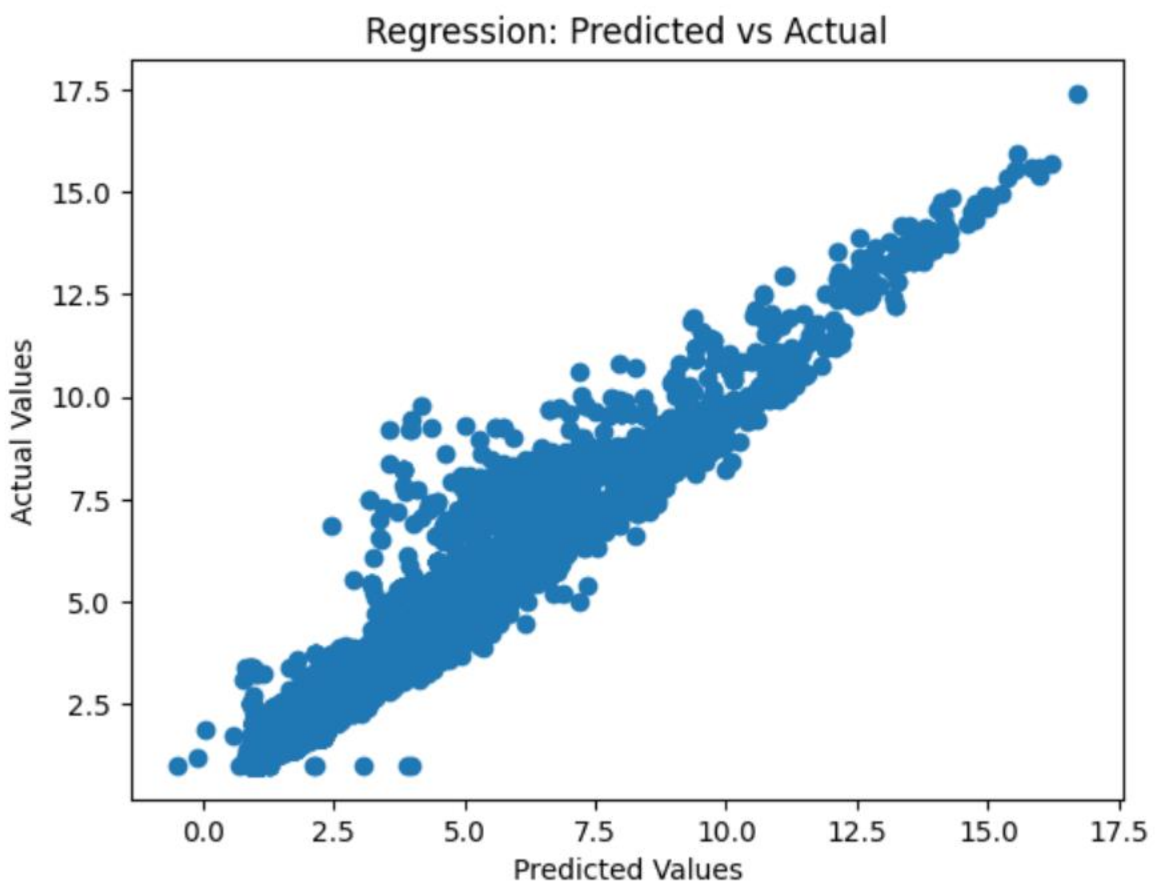


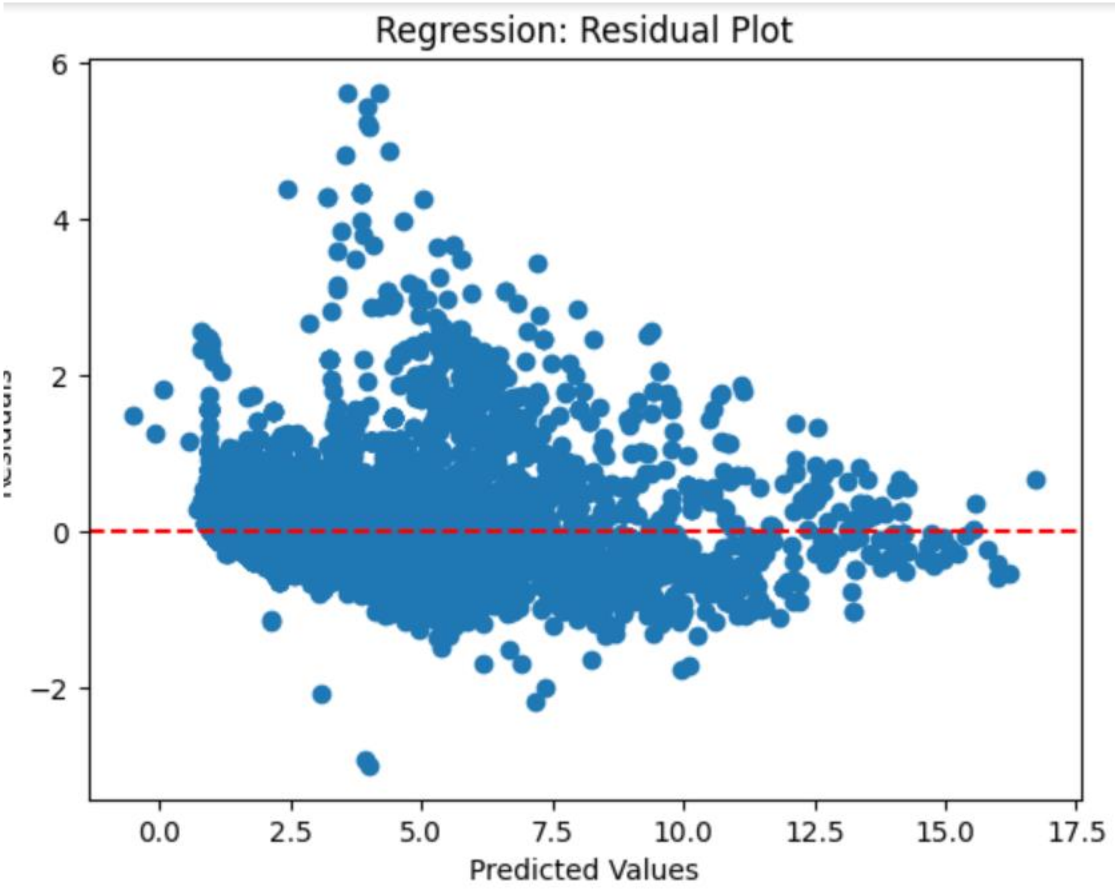
```
[ ]  import matplotlib.pyplot as plt

     # Assuming you have the predicted values in 'y_pred' and actual values in 'y_actual'

     # Scatter Plot
     plt.scatter(pred, y_test)
     plt.xlabel('Predicted Values')
     plt.ylabel('Actual Values')
     plt.title('Regression: Predicted vs Actual')
     plt.show()

     # Residual Plot
     residuals = y_test - pred
     plt.scatter(pred, residuals)
     plt.xlabel('Predicted Values')
     plt.ylabel('Residuals')
     plt.title('Regression: Residual Plot')
     plt.axhline(y=0, color='r', linestyle='--')  # Adding a horizontal line at y=0
     plt.show()
```





Conclusion:-

The project aimed to apply regression analysis to predict the "Neighborhood Connectivity" in a metabolic relation network dataset. The dataset was preprocessed by removing the "Pathway" column and splitting it into features (X) and the target variable (y). The data was then standardized using the StandardScaler and reduced to 15 components using Principal Component Analysis (PCA).

Four regression models, including Linear Regression, Support Vector Regression (SVR), Decision Tree Regressor, and Random Forest Regressor, were trained and evaluated. The performance of each model was measured using the R-squared score on the test set.

The results of the regression analysis indicated the following:

- Linear Regression (lra) achieved an R-squared score of 0.9438742874635677.

- SVR with RBF kernel (svry) achieved an R-squared score of 0.9792836507030624.
- Decision Tree Regressor (dtrr) achieved an R-squared score of 0.9819923879654109.
- Random Forest Regressor (rfra) achieved an R-squared score of 0.9905527421780146.

Based on the R-squared scores, it can be concluded that the **Random Forest Regressor** performed the best among the evaluated models for predicting the "Neighborhood Connectivity" in the metabolic relation network dataset.

The scatter plot visualizations showed a reasonably good alignment between the predicted and actual values, indicating that the models captured the underlying relationships to some extent. The residual plot exhibited a relatively even distribution around zero, indicating that the models had a reasonably good fit.

Refrences :-

1. Kanehisa Laboratories - KEGG: Kyoto Encyclopedia of Genes and Genomes: The KEGG database, which includes KEGG Metabolic Pathways, can be found at the following link: https://www.kegg.jp/
2. Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., & Tanabe, M. (2016). KEGG as a reference resource for gene and protein annotation. Nucleic Acids Research, 44(D1), D457-D462.
3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
4. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
5. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Springer.
6. Müller, A. C., & Guido, S. (2017). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media.
7. Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing
8. Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). Introduction to Linear Regression Analysis. Wiley.
9. Gelman, A., Hill, J., & Vehtari, A. (2020). Regression and Other Stories. Cambridge University Press.
10. Draper, N. R., & Smith, H. (2014). Applied Regression Analysis. Wiley.
11. Faraway, J. J. (2016). Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models. CRC Press
12. scikit-learn Documentation: The official documentation for scikit-learn, a popular Python library for machine learning, provides detailed explanations, tutorials, and examples: https://scikit-learn.org/stable/documentation.html
13. Towards Data Science: A popular online publication with a wide range of articles and tutorials on machine learning and data science:

https://towardsdatascience.com/

14. Medium: Medium hosts numerous articles and blog posts on machine learning and regression. Many data scientists and researchers share their insights and experiences on the platform: https://medium.com/

15. DataCamp: DataCamp offers interactive courses on various topics, including machine learning and regression, with hands-on exercises and projects: https://www.datacamp.com/