School of Computer Science Engineering and Technology

Course- BTech
Course Code: CSET 301
Year- 2022-23
Date-

Type- Specialization Core
Course Name- AIML
Semester- 4th
Batch-

Lab Assignment W3L1

Objective: The task is to compare the linear regression and polynomial regression with different degrees based on regression performance metrics.

1. Dataset: Download the dataset from the link https://archive.ics.uci.edu/ml/datasets/Behavior+of+the+urban+traffic+of+the+city+of+Sao+Paulo+in+Brazil

Data Set Information:

The database was created with records of behavior of the urban traffic of the city of Sao Paulo in Brazil from December 14, 2009 to December 18, 2009 (From Monday to Friday). Registered from 7:00 to 20:00 every 30 minutes. The data set Behavior of the urban traffic of the city of Sao Paulo in Brazil was used in academic research at the Universidade Nove de Julho - Postgraduate Program in Informatics and Knowledge Management.

Attribute Information:

1. Hour
2. Immobilized bus
3. Broken Truck
4. Vehicle excess
5. Accident victim
6. Running over
7. Fire Vehicles
8. Occurrence involving freight
9. Incident involving dangerous freight
10. Lack of electricity
11. Fire
12. Point of flooding
13. Manifestations
14. Defect in the network of trolleybuses
15. Tree on the road
16. Semaphore off
17. Intermittent Semaphore
18. Slowness in traffic (%) (Target)

.arff header for Weka:
@relation Behavior

@attribute Hour {7:00, 7:30, 8:00, 8:30, 9:00, 9:30, 10:00, 10:30, 11:00, 11:30, 12:00, 12:30, 13:00, 13:30, 14:00, 14:30, 15:00, 15:30, 16:00, 16:30, 17:00, 17:30, 18:00, 18:30, 19:00, 19:30, 20:00}
@attribute Immobilized_bus INTEGER
@attribute Broken_Truck INTEGER
@attribute Vehicle_excess INTEGER
@attribute Accident_victim INTEGER
@attribute Running_over INTEGER
@attribute Fire_vehicles INTEGER
@attribute Occurrence_involving_freight INTEGER
@attribute Incident_involving_dangerous_freight INTEGER
@attribute Lack_of_electricity INTEGER
@attribute Fire INTEGER
@attribute Point_of_flooding INTEGER
@attribute Manifestations INTEGER
@attribute Defect_in_the_network_of_trolleybuses INTEGER
@attribute Tree_on_the_road INTEGER
@attribute Semaphore_off INTEGER
@attribute Intermittent_Semaphore INTEGER
@attribute Slowness_in_traffic_percent REAL

Task:

2. Encoding: Load the dataset into the code for pre-processing. 1st feature 'Hour' can be discretized into labels such as [morning, noon, afternoon, evening, night], which can be further codes using one-hot encoding, where morning can be represented as [0,0,0,0,1], noon can be [0,0,0,1,0] and so on. This results single feature "Hour" to be represented using five features of binary values. This now makes the dataset to have four extra columns. Choice of discretization is up to you. You can have like [day, night] also.

3. Normalization: Since the features are in different ranges, each column can be normalized into 0 to 1 using different methods such as scaling, standardizing etc. Note: Normalization should not be done for the target feature.

4. Data Splitting: After the range normalization, its time to split the data into training and testing. Dataset contain 135 entries (5 days data, each day 27 entries), so keep the last 27 rows of the original dataset (data of last 1 day) for testing, and rest of them for training.

5. Regression Models: Train different models for regression such as Linear Regression and Polynomial Regression use degree 2. [use sklearn.linear_model.LinearRegression for linear regression model and sklearn.preprocessing.PolynomialFeatures for polynomial regression]

6. Testing: Test the model with the test data and compute the mean squared error (MSE) for test data. Use different train-test split ratio: 70:30, 80:20, 90:10 and see the effect on the performance of the model in testing set.

7. Regression Evaluation Metrics: Evaluate the trained model using regression measures such as mean squared error, mean absolute error, median absolute error, R2 score.

Playing with the Model:

You can try different strategies to see whether testing error comes down or not.
Strategies can be different
1. Encoding of features,
2. Shuffling of training
samples,
3. Degree of polynomials (such as 2, 3 and 4, print parameters of the polynomial models),
Check the model error for the testing data for each setup.

Useful links
1. https://www.analyticsvidhya.com/blog/2021/10/understanding-polynomial-regression-model/
2. https://www.w3schools.com/python/python_ml_polynomial_regression.asp