

## Lab Assignment 3

### Machine Learning (UML501)

|     |  |
|-----|--|
| Q 1 | <p>K-Fold Cross Validation for Multiple Linear Regression (Least Square Error Fit)</p> <p>Download the dataset regarding USA House Price Prediction from the following link:<br/> <a href="https://drive.google.com/file/d/1O_NwpJT-8xGfU_-3llU2sgPu0xllOrX/view?usp=sharing">https://drive.google.com/file/d/1O_NwpJT-8xGfU_-3llU2sgPu0xllOrX/view?usp=sharing</a></p> <p>Load the dataset and Implement 5- fold cross validation for multiple linear regression (using least square error fit).</p> <p>Steps:</p> <ol style="list-style-type: none"> <li>Divide the dataset into input features (all columns except price) and output variable (price)</li> <li>Scale the values of input features.</li> <li>Divide input and output features into five folds.</li> <li>Run five iterations, in each iteration consider one-fold as test set and remaining four sets as training set. Find the beta (<math>\beta</math>) matrix, predicted values, and R2_score for each iteration using least square error fit.</li> <li>Use the best value of (<math>\beta</math>) matrix (for which R2_score is maximum), to train the regressor for 70% of data and test the performance for remaining 30% data.</li> </ol>  |
| Q 2 | <p>Concept of Validation set for Multiple Linear Regression (Gradient Descent Optimization)</p> <p>Consider the same dataset of Q1, rather than dividing the dataset into five folds, divide the dataset into training set (56%), validation set (14%), and test set (30%).</p> <p>Consider four different values of learning rate i.e. {0.001,0.01,0.1,1}. Compute the values of regression coefficients for each value of learning rate after 1000 iterations.</p> <p>For each set of regression coefficients, compute R2_score for validation and test set and find the best value of regression coefficients.</p>  |
| Q 3 | <p>Pre-processing and Multiple Linear Regression</p> <p>Download the dataset regarding Car Price Prediction from the following link:<br/> <a href="https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data">https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data</a></p> <ol style="list-style-type: none"> <li>Load the dataset with following column names ["symboling", "normalized_losses", "make", "fuel_type", "aspiration", "num_doors", "body_style", "drive_wheels", "engine_location", "wheel_base", "length", "width", "height", "curb_weight", "engine_type", "num_cylinders", "engine_size", "fuel_system", "bore", "stroke", "compression_ratio", "horsepower", "peak_rpm", "city_mpg", "highway_mpg", "price"] and replace all ? values with NaN</li> <li>Replace all NaN values with central tendency imputation. Drop the rows with NaN values in price column</li> <li>There are 10 columns in the dataset with non-numeric values. Convert these values to numeric values using following scheme: <ol style="list-style-type: none"> <li>For "num_doors" and "num_cylinders": convert words (number names) to figures for e.g., two to 2</li> <li>For "body_style", "drive_wheels": use dummy encoding scheme</li> <li>For "make", "aspiration", "engine_location", fuel_type: use label encoding scheme</li> <li>For fuel_system: replace values containing string pfi to 1 else all values to 0.</li> <li>For engine_type: replace values containing string ohc to 1 else all values to 0.</li> </ol> </li> <li>Divide the dataset into input features (all columns except price) and output variable (price). Scale all input features.</li> <li>Train a linear regressor on 70% of data (using inbuilt linear regression function of Python) and test its performance on remaining 30% of data.</li> <li>Reduce the dimensionality of the feature set using inbuilt PCA decomposition and then again train a linear regressor on 70% of reduced data (using inbuilt linear regression function of Python). Does it lead to any performance improvement on test set?</li> </ol> |