NUMBER: 10515678
COURSE TITLE: MSC IN DATA ANALYTICS
LECTURER NAME: ABHISHEK KAUSHIK
MODULE: MACHINE LEARNING
ASSIGNMENT TITLE: MACHINE LEARNING CA 2
NO OF WORDS: 2456

Koushik Chikkegowda

Data preprocessing Stages

Data preprocessing can be described as techniques used to transform raw data into a human-understandable format. Some of the techniques used in Data Pre-Processing are explained below:

Data Cleaning

Data obtained from the real-world may contain **outliers** and **missing values**. Deciding on an appropriate strategy to deal with missing values and outliers is crucial for machine learning. For example, a weather dataset of Dublin city has these features temperature, wind gust, humidity. Suppose if outliers are showing the temperature above 30 degrees which never happens. Those types of outliers can be eliminated to reduce noise. Similarly, if there are any missing values in any feature appropriate techniques like replacing it with (mean or median or mode) can be used, or missing value can be ignored as well. Missing values can also be filled using suitable algorithms as well.

Feature Engineering

Feature Selection: These are techniques used to select the best features and removal of redundant features in the dataset. Features selection can be done using variance threshold, correlation threshold or genetic algorithms. For example - A diabetic dataset that holds information such as name, salary, BMI, Sugar level and diabetes status. Here features like salary are redundant and can be eliminated by only selecting necessary features.

Feature Extraction: this is the process of creating new smaller features that capture most of the useful information. Some of the feature extraction techniques are PCA (Principal component analysis), LDA (Linear discriminant analysis). If there is a dataset with a high number of features there will be needed to reduce the number of features by capturing most of the important information.

Feature Scaling: These are techniques used to standardize the independent variables. Different variance in independent variables makes the machine learning model less precise. For example, an Employee dataset consists of age, salary, pf fund, years of experience and employee performance (Output Variable). The variance of all the independent variables are different and could be standardized using any one of these techniques like Standardize, Binarize or Normalize.

Decision Tree, Information gain, Entropy

Decision Trees is a non-parametric and supervised learning technique utilized for both classification and regression tasks. The objective is to create a decision model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It's a tree-like structure which consists of nodes and edges. Where the node is where the algorithm chooses a feature and asks a question and edges is where the question is answered. If the decision tree is deep then the rules will be complicated, and the model will be much fitter.

Entropy is the rate of ambiguity in a random variable. It signifies the level of impurity in a subset of data. Entropy decides how a decision tree splits data. The higher the entropy, the more the information content. If the probability that the data is completely pure or impure is equal to one, then entropy would be zero. The value of entropy is high if the probability of impurity or purity is 0.5.

Information gain: In the Decision tree the dataset is divided into subsets. At each level, the entropy value changes. Information gain is this change in entropy. In other words, information gain is the measure in the reduction of entropy. It decides which attribute should be selected as the decision node.

Chinese restaurant algorithm

Chinese restaurant algorithm is a theoretical discreet time process, as the process of seating people in a Chinese restaurant. Suppose there are infinite circular-shaped tables in a Chinese restaurant, where each table can hold an infinite number of people. If a customer enters the shop and sits at a table and another customer that may enter could sit at the same table as the previous customer or could sit in a new table. Any time a new customer enters the shop could sit at a table which is occupied or empty, although the probability of a customer choosing a table with more people is high (the probability of a customer choosing an occupied table is directly proportional to the number of people already sitting in that table). The takeaway from this process is that the probability of the final distribution is not determined by the order in which the customers are seated, this feature of this process simplifies several issues within population genetics, linguistic analysis, and image recognition. There are various applications of using the Chinese restaurant process, this includes modeling text, clustering biological microarray data, biodiversity modeling, and image reconstruction.

Regression Report

Dataset Description: This dataset contains information about the weather in Chapel Hill. Data is obtained from a station located at the Chapel Hill Public Library.

Dataset Link: https://catalog.data.gov/dataset/weather-trends

Dataset columns: Temperature, humidity, wind speed, wind gust, daily rain, monthly rain, yearly rain, UV radiation, and date.
There are 28138 records in this dataset, I have selected **temperature** as my prediction variable. Temperature prediction is used in weather forecasting and is essential to perform various operations.

Data Import: Since the dataset has 28138 rows and only 9 rows it would impractical to read the entire rows because it will cause underfitting. To avoid that I'm importing a sample of 1500 rows to perform modelling.

Data Pre-processing:

Checking Null values: There were no Null Values within any feature.

Checking for Outliers: In the box plot, I detected outliers within wind speed, wind gust, daily rain, and UV radiation. To tackle this, I have used a Z-Score technique to reduce noise. I have kept the z threshold as 3. So whichever data element which has a z-score more than 3 will be eliminated. 95 rows were removed after applying this method.

Feature Engineering: I have performed both manual and automatic feature selection approaches here.

Manual Feature selection analysis

Correlation Analysis: I have plotted a correlation matrix here, from this I found that humidity had a high negative correlation with temperature and wind speed, wind gust, yearly rain, and UV radiation had a positive correlation with temperature. Both daily rain and monthly rain correlated near to zero.

Checking for skewness: By observing the skewness I found that humidity, yearly rain, and monthly rain had neutral skewness whereas wind speed, wind gust, and UV emission had positive skew. daily rain had extreme positive skew which would not be ideal as input for machine learning.

Density Plot: Looking at the density plot we can see humidity, wind speed, and UV radiation somewhat follow a gaussian distribution. Gaussian distribution is ideal for the regression algorithm.

Automatic Feature Selection: For automatic feature selection I have used SelectKBest from Scikit Learn pre-processing. This function provides a score for each feature based on the importance level towards the output variable. I have used the link function f_regression to analyse. After analysis, I found humidity, wind speed, wind gust, yearly rain and UV emission as having the maximum importance.

Bias and Variance Trade-off: As my sample dataset has 1405 rows after removing outliers and I'm selecting 5 important features, to main bias and variance balance. As each feature should have at least 200 rows or a maximum of 1000 rows to maintain balance.

Data Transformation: For data transformation, I'm using Min Max Scaler which is best suited for regression algorithms like Linear Regression.

Cross-Validation: For cross-validation, I will be using cross-validation function from Scikit Learn-model selection library. Using this function, I can evaluate multiple models to check which model is best suited for my dataset. I have checked linear regression, random forest regression, Lasso regression and BayesianRidge regression for analysis. I found that the cross-validation score for Linear regression, Random forest Regression, Lasso regression, and BayesianRidge were 0.69, 0.80, 0.67 and 0.69 respectively. So, by looking at the results Linear regression would be a better choice.

Applying Model: I'm applying both Linear Regression and Random forest regression for my dataset. I'm using train_test_split function from the Scikit Learn model selection library to split the data into training (70 percent) and testing (30 percent).

Evaluating Model: For evaluation, I have used Mean squared error, Mean Absolute Error, variance Score. For Linear regression Mean squared error, Mean Absolute Error and variance Score were 0.28, 0.42 and 0.71. Whereas for Random forest regression Mean squared error, Mean Absolute Error and variance Score were0.30, 0.43 and 0.70. By analysing this result, I can conclude that Linear regression had a low error and accuracy.

Classification Report

Dataset Description: This is a LIHEAP Demographics Dataset where LIHEAP stands for a low-income home energy assistance program. A complete record of all LIHEAP clients based upon specified Start/End date and Status (Approved, Denied, Pending, All Statuses) including but not limited to total benefit, first/last name, age, race, disability, address, the client who processed, application number, application date/time submitted, income, etc.

Dataset link: https://catalog.data.gov/dataset/liheap-demographics

Here I will be predicting the application status (Approved, Denied and Void) using an appropriate machine learning algorithm. By creating the ideal model, it can be used to automate the entire process. Which in turn reduce expenses and increases the speed of the process.
The dataset has 90839 rows and 30 features in the dataset.

Data Import: Since the dataset contains a high number of rows and only 30 columns. It would not be ideal to import all the rows because it would cause underfitting. So, I would be importing a sample set of 3000 rows to perform the analysis.

Data Pre-processing:

Checking for null values: During the analysis, I found that both LIHEAPFUELTYPE and VENDOR NAME had 44 and 154 null values respectively. So, to handle this I'm replacing these Null values with the mode of the feature.

Checking for outliers: Since all the input features are categorical there is no need to check for outliers.

Label Encoding: Since machine learning models require input variables to be numerical, I am converting my input categorical variables into numerical values. I am using Label encoder from Scikit Learn pre-processing. I am creating a new column for each categorical feature that holds the converted numerical value. In the later stage, I will be dropping the original column which held the string label.

Class Balance: I plotted a bar chart to visualize the label count in the output variable. I found that the "Approved" count was extremely high compared to "Denied" and "Void". I know the machine learning model generates biased results towards Approved If I don't balance it. To balance the labels, I'm using an up-sampling strategy to balance the output labels. I'm using SMOTETomek function from the imblearn library. **Please install the library since anaconda does not have it inbuilt (command: pip install -U imbalanced-learn)**. Hereafter balancing the output label my dataset has 8313 rows. I have done another visualization to show the balance.

Data Transformation: For data transformation, I'm using Min Max Scaler. I'm transforming only one feature Benefit amount which is a continuous variable and updating the dataset with the transformed value.

Automatic Feature Selection: For automatic feature selection I will be using SelectKBest from Scikit Learn feature selection. The link function I'm using here is mutual_info_classif which is used in classification. I'm selecting the top 12 features since I'm having 8313 rows to maintain a balance between variance and bias.

Bias and variance Balance: I have selected the top 12 features which had relevance towards the target variable. Since the sample dataset has 8313 rows after the class label balancing using SMOTETomek. There is a good balance between bias and variance.

Cross-Validation: For cross-validation, I have used cross_val_score from scikit learn model selection. I have used the scoring link function as accuracy. I have tested the accuracy for KNN, SVM, Gaussian, Decision Tree, and Random forest. When I compared, I found that Decision Tree and Random forest had high accuracy.

Modelling: I have split the dataset into training and testing using train_test_split from scikit learn model selection in the ratio 65:35. I have applied both decision tree and Random Forest since they had a high accuracy in cross-validation.

Model Evaluation: I have used the Confusion matrix and classification_report for analysing the results. I found accuracy, precision and recall for decision tree as 0.96. Whereas random forest had precision, recall, accuracy as 0.86. There is a good balance between recall, precision and f1 score between each label class Approved, Denied and Void.

Since the Decision Tree had the highest accuracy of 96 percent this would be the best suited for this dataset.

Take away from the task

- depth understanding of data with the use of different visualization methods and reports like checking the dimension of the data, skewness, distribution, correlation, etc.
- Identifying the business objective in the data and develop a methodology to satisfy that business goal.
- Better knowledge of Data cleaning methods to handle outliers and missing values like using z score to filter outliers and imputing missing values with mode.
- Better understanding about the use of rescaling techniques, I found that we use rescaling methods mainly to bring all the input variables to a certain range and we use this technique if that data has distance-based data elements where each feature may have different variance.
- In classification, I understood that we need to convert all the categorical inputs to numeric labels because the machine learning won't work on strings.
- I got familiar with feature selection techniques that can be used to select relevant features. Instead of feeding all the features to the model which may give inaccurate results.
- In Regression, I had a better understanding of how we select features by analysing distribution, skewness, and correlation. I understood the input features should be correlated with the out variable but not correlated with each other. And to build an accurate model in regression the input variables which are continuous in nature should be close to Gaussian distribution.
- Better Understanding about the types of errors in machine learning like Underfitting and Overfitting. And the techniques which are used to avoid these errors. This can be avoided by maintaining an ideal balance between the number of rows and columns.
- Became familiar with cross-validation techniques to measure the score between various models and selecting the best algorithm before modelling.
- Leant different sampling techniques like K-fold and train test split. I learned that k-fold uses a shuffle method to obtain different iterations of train and test split set where the accuracy of these are aggregated at the end. Whereas the usual train and test methods split the data into 70 to 30 ratios, or the ratio specified by the user.
- Understood different evaluation techniques for Classification and Regression like Confusion matrix, Auc, Accuracy, Precision, F1 Score, MSE and MAE, R2.