MASTER OF DATA SCIENCE (SMDS)

Assignment-1

Machine Learning (CSE5ML)

by

SAIKIRAN CHALLA
21356161

(Semester-2 2023)

Aim : The aim of the project is to accurately predict the individual income (greater than 50k or not) based on factors given.

Task -1

Before Pre-processing :

At first, the given data is in the form of .csv(income.csv) file. Here we are importing pandas and NumPy libraries. Pandas is used to load various data formats(CSV, Excel, SQL etc) in to data frames. Data Frame is preferred data structure for working with tabular data. NumPy (Numerical Python) is a fundamental package for numerical computing in python. It is used for handling numerical operations, making them essential for ML algorithms. We have "income.csv" file with us, converting that csv file to data frame and storing it in the variable "ds". The data frame has 26,215 rows and 10 columns of data. The column names of the data should be changes as they contains hyphen ("-"). In pandas, column names with hyphens ("-") are technically allowed and will not result in any syntax errors. So, removing the hyphen in column names by replacing (marital-status by maritalstatus, hours-per-week by hoursperweek) it with the other names as shown in the code. Now checking if there are any missing values in the data and removing the missing columns. Dropping (Deleting) all the duplicate rows. Handling the categorical variables by replacing strings to numbers as asked in the question by **replace( )** function. Applying dummy_coding for the rest of variables by using **get_dummies** function from pandas library. Applying dummy coding to categorical variables essentially transform them into a format that can be effectively used in machine learning models. So we can use this data so that we can train the Machine Learning Model correctly.

After Pre-processing :

The whole data is in numerical values, So the now we can used for training and testing the data. Now assigning first column target and true label(y) to **y**, and rest of the 9 columns to input **X**. Splitting the Training and Testing data with 10% of data for testing and rest of 90% data for training with **sklearn.model_selection.** sklearn.model_selection is a module in the popular Python machine learning library scikit-learn. This module provides a suite of functions and classes to support model selection and evaluation, including techniques for splitting datasets into train and test sets, cross-validation, and hyperparameter tuning. Efficient model selection is critical for building robust and high-performing machine learning models. Now applying the normalization on X (both training and testing set) using sklearn.pre-processing. Normalization, also known as feature scaling, is a crucial pre-processing step in machine learning (ML) models. It involves transforming the features (variables) of the dataset into a standardized range, typically to ensure that they have similar scales or distributions. Normalization offers several benefits in training ML.

Task -2

Logistic Regression :

Logistic regression is a fundamental algorithm in machine learning used for binary classification tasks, where the goal is to predict the probability that a given input belongs to a particular class. Logistic regression predicts the probability that a given input sample belongs to a certain class (usually denoted as class 1). The output is a probability value between 0 and 1.

From the given question the model was trained using the Logistic regression algorithm. Then, after testing the data with test data set the accuracy of the model is 81%.

SVM :

SVM is a strong supervised learning technique that is used for both classification and regression applications. SVM seeks to identify a hyperplane that optimally separates data points belonging to various classes in the context of classification. This hyperplane is defined as the one with the shortest distance between it and the nearest data point of any type. SVM is especially effective in high-dimensional feature spaces.

From the given data of the question, model was trained by Support Vector Machine classifier. The accuracy is 79.5%.

Now, **10-fold Cross Validation** technique was applied with both Logistic Regression and Support Vector machine classifier as given in the question. **K-fold cross-validation** is a crucial technique in machine learning for assessing the performance and generalization ability of a model. It's used to better utilize the available data, especially when the dataset is limited. K-fold cross-validation helps in creating a more reliable and robust machine learning model that is likely to perform well on unseen data. For Logistic Regression, the average accuracy is 80% and for the Support Vector Machine Classifier the average accuracy we got is 79%. Here the accuracy is slightly decreased. As k-fold cross-validation is not intended to improve performance, it is intended to offer a more accurate assessment of performance.

Parameter Fine tuning, also known as hyperparameter tuning, is an important stage in machi ne learning for optimizing model performance. Hyperparameters are settings that are not lea rnt from data but are established before the training process begins. Techniques like grid sea rch and random search are commonly used for parameter tuning. For Logistic regression Mo del, Grid Search technique is used by Importing GridSearchCv Library. Here the hyper parame ters are defined to tune their respective values. The Hyper parameters are stored in the form of dictionaries. The hyper parameters (key-value pairs) we use here are penalty, C and solver with different values. For small 'C' values the model doesn't allow any misclassifications. Here by using GridSearchCv we are determining for which combination of these parameters are ge tting high accuracy. By testing the Model with test data using GridSearchCv parameter Fine t uning the accuracy we got is 81%, and the best parameters are {'C': 1, 'penalty': 'l1', 'solver': 's aga'}.

Similarly, applying the parameter fine tuning to Support Vector Machine Classifier with kernel , C, degree, gamma as parameters and Cross validation as of taken in kfold. The accuracy we got here is 80%.

From the instance, there is an increase in accuracy of the models after applying the paramete r fine tuning. The change in accuracies after applying different types of techniques on two models are as shown below in the table:

Accuracy(%)	Before K-fold CV	After K-fold CV(Avg. accuracy)	After Parameter Finetuning
Logistic Regression	81%	80%	81%
SVC	79.5%	79%	80%

From the above table the accuracy K-fold Cross validation is more than that of after K-fold Cross validation. As, k-fold CV is not to increase the performance but to provide a more accurate measure of performance and it is average of all accuracies so definitely the value will be nearer to the mean of all the accuracy values we got by applying with Cross validation technique. Also there is an increase in accuracy for both the Logistic Regression and SVC model.

Task – 3

K-MEANS:

K-means clustering is a fundamental unsupervised machine learning technique that divides a dataset into a fixed number of groups (K). K-means seeks to reduce the sum of squared distances (Euclidean distance) between data points and the centroid of their respective clusters. It's widely used for clustering and segmentation tasks. K-means is widely used in various applications, including customer segmentation, image compression, document clustering, and more. However, it's important to note that K-means may not always produce the optimal clustering, and the quality of clusters can be influenced by the initial centroids and the choice of K.

In this model, only 2 clusters(k=2) were used as the income has only 2 classes(>50k or not). Here random_state=0,"random_state" parameter serves as a seed to the random number generator used during the initialization of centroids. Setting the random_state ensures that the initialization of centroids is reproducible, yielding consistent results across different runs of the K-means algorithm. So, after training the model, the data samples assigned to Cluster '0' is 11,330 , and to Cluster '1' is 10,207. The prototype for each cluster is as shown in the code. The accuracy on the given data is 72%.

The accuracies after applying three models are as shown below in the table:

Models	Accuracies
Logistic Regression	81%
SVM	79.5%
K-Means	72%

References :

- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow 3e: Concepts, Tools, and Techniques to Build Intelligent Systems.