Marathwada Mitramandal's

COLLEGE OF ENGINEERING

Karvenagar, PUNE – 411 052

Accredited with "A++" Grade by NAAC // Accredited by NBA (Mechanical Engg. & Electrical Engg.)
Recipient of "Best College Award 2019" by SPPU // Recognized under section 2(f) and 12B of UGC Act 1956

DEPARTMENT OF ARTIFICIAL INTELLIGENCE& DATA SCIENCE

LABORATORY MANUAL

BE (AI & DS)

(SEMESTER – VII)

COMPUTER LABORATORY – I
[MACHINE LEARNING]

2020 Course

Prepared By
Mrs. G.G.Bilaye

Preface

Machine learning (ML) represents a transformative approach to artificial intelligence (AI) that empowers computers to learn from data and improve over time without explicit programming. In essence, ML algorithms enable systems to recognize patterns, make decisions, and even predict outcomes based on experiences or historical data. This capability has revolutionized industries ranging from healthcare and finance to entertainment and transportation. Understanding machine learning is not just about mastering algorithms; it's about binding the potential to extract meaningful insights, automate complex processes, and drive decision-making with unique accuracy and efficiency.

This manual is designed to be companion on a journey through the principles, techniques, and applications of machine learning (ML). It also aims to provide you with a hands-on approach to mastering the fundamentals and advancing your skills in ML. The aim is to bridge the gap between theory and practice in machine learning. It offers a structured learning path that combines theoretical insights with practical implementation through guided exercises. You will have gained proficiency in building and evaluating ML models, understanding their underlying concepts, and applying them to real-world problems.

Marathwada Mitramandal's

COLLEGE OF ENGINEERING

S.No.18, Plot No.5/3, Karvenagar, Pune-52
Accredited with "A++" Grade by NAAC | Recipient of "Best College Award 2019" by SPPU
Accredited by NBA (Mechanical Engg. & Electrical Engg.)

Vision of the Institute

- To aspire for the welfare of society through excellence in science and technology.

Mission of the Institute

- Mould young talent for higher endeavors.
- Meet the challenges of globalization.
- Commit for social progress with values and ethics.
- Orient faculty and students for research and development.
- Emphasize excellence in all disciplines.

Department of Artificial Intelligence & Data Science

Vision of the Department

Excellence in the field of Artificial Intelligence and Data science

Mission of the Department

- To encourage the students for learning various AI & DS based tools & techniques
- To groom students technologically superior
- To educate students in a way to meet the needs of society

Program Educational Objectives(PEOs)

The students of AI & DS Program after passing out will possess:

- To educate and empower the students with problem-solving, analytical skills using the latest tools in the area of Artificial Intelligence & Data Science for a successful career.
- To inspire self-learning skills amongst students to carry out research, innovation, and pursue higher studies.

- To simplify training programs that associations the gap between academia and industry.
- To inculcate cultural, societal, professional, and ethical responsibilities amid the students.

Program Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

Graduates from our program will

- Design and develop the computational models to solve multi-disciplinary problems in Artificial Intelligence.
- Provide a definite groundwork and augment their capabilities to succeed for employment, higher studies and research in Data Science with principled standards.

Marathwada Mitra Mandal's

COLLEGE OF ENGINEERING

S.No.18, Plot No.5/3, Karvenagar, PUNE -52

Accredited with "A++" Grade by NAAC // Accredited by NBA

Recipient of "Best College Award 2019" by SPPU // Recognized under section 2(f) and 12B of UGC Act 1956

DEPARTMENT OF ARTIFICIAL INTELLIGENCE& DATA SCIENCE

List of Experiments

Academic Year: 2024-25 **Semester:** I

Course Code & Name: Computer Laboratory I (Machine Learning) **Class:** BE

Sr. No.	Expt. No.	Experiment Title
		As per SPPU Syllabus
1	1	**ature Transformation:** Apply LDA Algorithm on Iris Dataset and classify which species a given flower belongs to. taset Link: https://www.kaggle.com/datasets/uciml/iris
2	2	**gression Analysis:** A. Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks: 1. Pre-process the dataset. 2. Identify outliers. 3. Check the correlation. 4. Implement linear regression and ridge, Lasso regression models. 5. Evaluate the models and compare their respective scores like R2, RMSE, etc. tasetlink:https://www.kaggle.com/datasets/yasserh/uber-fares-dataset
3	3	**assification Analysis:** Implementation of Support Vector Machines (SVM) for classifying images of hand- written digits into their respective numerical classes (0 to 9).
4	4	**ustering Analysis**: Implement K-Means clustering on Iris.csv dataset. Determine the number of clusters using the elbow method. taset Link: https://www.kaggle.com/datasets/uciml/iris
5	5	**semble Learning:** Implement Random Forest Classifier model to predict the safety of the car. tasetlink: https://www.kaggle.com/datasets/elikplim/car-evaluation-data-set

6	6	**Reinforcement Learning:** Build a Tic-Tac-Toe game using reinforcement learning in Python by using following tasks a. Setting up the environment b. Defining the Tic-Tac-Toe game c. Building the reinforcement learning model d. Training the model e. Testing the model

Course Coordinator: Mrs. G.G.Bilaye

Course Code: 417525

Course Name: Computer Laboratory I (ML) (2020 Course)

ASSIGNMENT NO.1

Problem Statement:

Apply LDA Algorithm on Iris Dataset and classify which species a given flower belongs to.

Dataset Link: https://www.kaggle.com/datasets/uciml/iris

Tools / Environment:

Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder.

64-bit Open source Linux or its derivative

Related Theory:

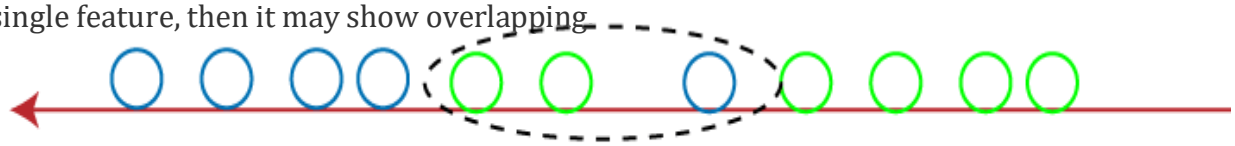
Linear Discriminant Analysis (LDA):

Although the logistic regression algorithm is limited to only two-class, linear Discriminant analysis is applicable for more than two classes of classification problems.

Linear Discriminant analysis is one of the most popular dimensionality reduction techniques used for supervised classification problems in machine learning.

It is also considered a pre-processing step for modeling differences in ML and applications of pattern classification.

Whenever there is a requirement to separate two or more classes having multiple features efficiently, the Linear Discriminant Analysis model is considered the most common technique to solve such classification problems. For e.g., if we have two classes with multiple features and need to separate them efficiently. When we classify them using a single feature, then it may show overlapping.

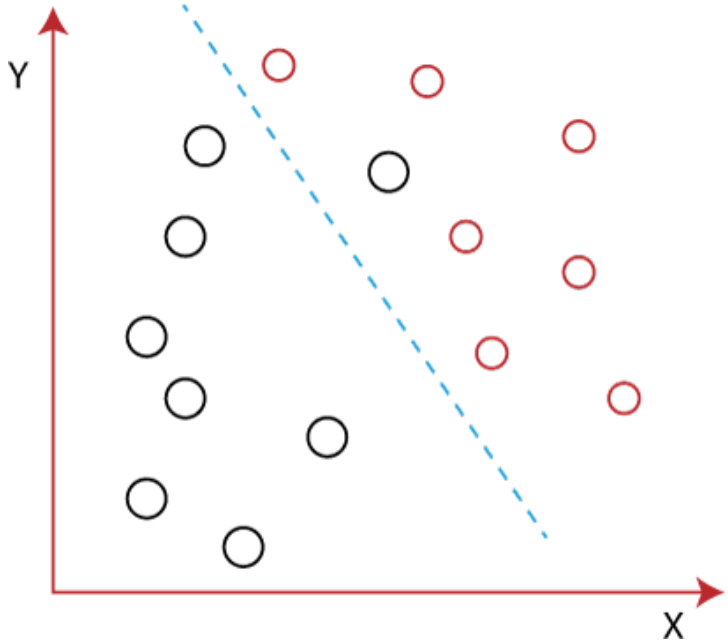


Overlapping

To overcome the overlapping issue in the classification process, we must increase the number of features regularly.

Example:

Let's assume we have to classify two different classes having two sets of data points in a 2-dimensional plane as shown below image:



However, it is impossible to draw a straight line in a 2-d plane that can separate these data points efficiently but using linear Discriminant analysis; we can dimensionally reduce the 2-D plane into the 1-D plane. Using this technique, we can also maximize the separability between multiple classes.

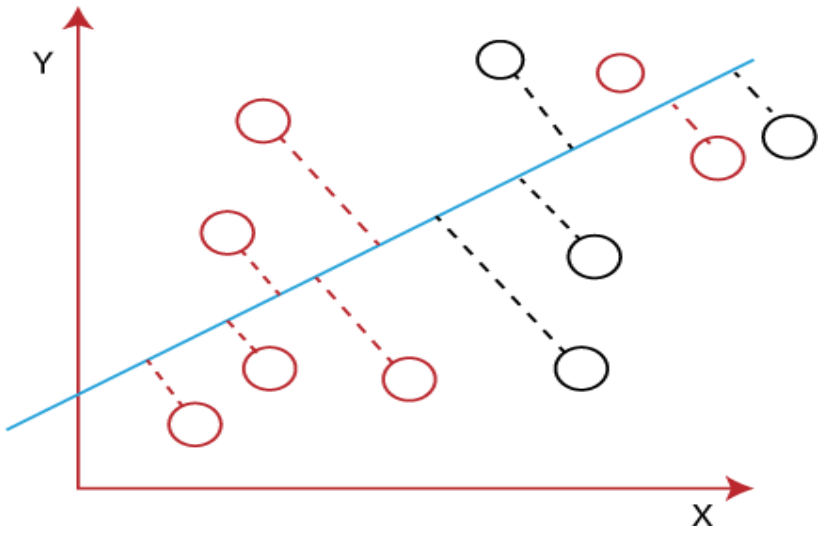
How Linear Discriminant Analysis (LDA) works?

Linear Discriminant analysis is used as a dimensionality reduction technique in machine learning, using which we can easily transform a 2-D and 3-D graph into a 1-dimensional plane.

Let's consider an example where we have two classes in a 2-D plane having an X-Y axis, and

we need to classify them efficiently. As we have already seen in the above example that LDA enables us to draw a straight line that can completely separate the two classes of the data points. Here, LDA uses an X-Y axis to create a new axis by separating them using a straight line and projecting data onto a new axis.

Hence, we can maximize the separation between these classes and reduce the 2-D plane into 1-D.



To create a new axis, Linear Discriminant Analysis uses the following criteria:

- It maximizes the distance between means of two classes.
- It minimizes the variance within the individual class.

Using the above two conditions, LDA generates a new axis in such a way that it can maximize the distance between the means of the two classes and minimizes the variation within each class.

In other words, we can say that the new axis will increase the separation between the data points of the two classes and plot them onto the new axis.

Why LDA?

- Logistic Regression is one of the most popular classification algorithms that perform well for binary classification but falls short in the case of multiple classification problems with well-separated classes. At the same time, LDA handles these quite efficiently.
- LDA can also be used in data pre-processing to reduce the number of features, just as PCA, which reduces the computing cost significantly.
- LDA is also used in face detection algorithms. In Fisher faces, LDA is used to extract useful data from different faces. Coupled with Eigen faces, it produces effective results.

Implementation:

- In this implementation, we will perform linear discriminant analysis using the Scikit-learn library on the Iris dataset.

CONCLUSION:

- In this implementation, we will perform linear discriminant analysis using the Scikit-learn library on the Iris dataset.

QUESTIONS:

1) What is the main objective of Linear Discriminant Analysis (LDA)?

2) Which technique would you use if you need to reduce the dimensionality of data while ignoring class labels?

3) What are differences between PCA and LDA?

4) If you need to reduce dimensionality and also consider class labels for better classification, which technique should you use PCA OR LDA?

ASSIGNMENT NO.2

Problem Statement:

Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks:

1. Pre-process the dataset.
2. Identify outliers.
3. Check the correlation.
4. Implement linear regression and ridge, Lasso regression models.
5. Evaluate the models and compare their respective scores like R2, RMSE, etc.

Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset

Tools / Environment:

Open Source Python, Programming tool like Jupyter Notebook,Pycharm, Spyder.

64-bit Open source Linux or its derivative

Related Theory:

What is a Regression Model/Analysis?

Predictive modelling techniques such as regression model/analysis may be used to determine the relationship between a dataset's dependent (goal) and independent variables. It is widely used when the dependent and independent variables are linked in a linear or non-linear fashion, and the target variable has a set of continuous values. Thus, regression analysis approaches help establish causal relationships between variables, modelling time series, and forecasting. Regression analysis, for example, is the best way to examine the relationship between sales and advertising expenditures for a corporation.

What is the purpose of a regression model?

Regression analysis is used for one of two purposes: predicting the value of the dependent variable when information about the independent variables is known or predicting the effect of an independent variable on the dependent variable.

Types of Regression Models

There are numerous regression analysis approaches available for making predictions. Additionally, the

choice of technique is determined by various parameters, including the number of independent variables, the form of the regression line, and the type of dependent variable.

Let us examine several of the most often utilized regression analysis techniques:

1. Linear Regression

The most extensively used modelling technique is linear regression, which assumes a linear connection between a dependent variable (Y) and an independent variable (X). It employs a regression line, also known as a best-fit line. The linear connection is defined as Y = c+m*X + e, where 'c' denotes the intercept, 'm' denotes the slope of the line, and 'e' is the error term.

The linear regression model can be simple (with only one dependent and one independent variable) or complex (with numerous dependent and independent variables) (with one dependent variable and more than one independent variable).

2. Logistic Regression

When the dependent variable is discrete, the logistic regression technique is applicable. In other words, this technique is used to compute the probability of mutually exclusive occurrences such as pass/fail, true/false, 0/1, and so forth. Thus, the target variable can take on only one of two values, and a sigmoid curve represents its connection to the independent variable, and probability has a value between 0 and 1.

3. Polynomial Regression

The technique of polynomial regression analysis is used to represent a non-linear relationship between dependent and independent variables. It is a variant of the multiple linear regression model, except thatthe best fit line is curved rather than straight.

4. Ridge Regression

When data exhibits multicollinearity, that is, the ridge regression technique is applied when the independent variables are highly correlated. While least squares estimates are unbiased in multicollinearity, their variances are significant enough to cause the observed value to diverge from the actual value. Ridge regression reduces standard errors by biassing the regression estimates.

The lambda (λ) variable in the ridge regression equation resolves the multicollinearity problem.

$$= \underset{\beta \in \mathbf{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

Lasso Regression

As with ridge regression, the lasso (Least Absolute Shrinkage and Selection Operator) technique penalizes the absolute magnitude of the regression coefficient. Additionally, the lasso regression technique employs variable selection, which leads to the shrinkage of coefficient values to absolute zero.

5. Bayesian Linear Regression

Bayesian linear regression is a form of regression analysis technique used in machine learning that uses Bayes' theorem to calculate the regression coefficients' values. Rather than determining the least-squares, this technique determines the features' posterior distribution. As a result, the approach outperforms ordinary linear regression in terms of stability.

6. Elastic Net Regression

Elastic net regression combines ridge and lasso regression techniques that are particularly useful when dealing with strongly correlated data. It regularizes regression models by utilizing the penalties associatedwith the ridge and lasso regression methods.

Why we require Evaluation Metrics?

Most beginners and practitioners most of the time do not bother about the model performance. The talk is about building a well-generalized model, Machine learning model cannot have 100 per cent efficiency otherwise the model is known as a biased model. which further includes the concept of overfitting and under fitting.

It is necessary to obtain the accuracy on training data, but it is also important to get a genuine and approximate result on unseen data otherwise Model is of no use.

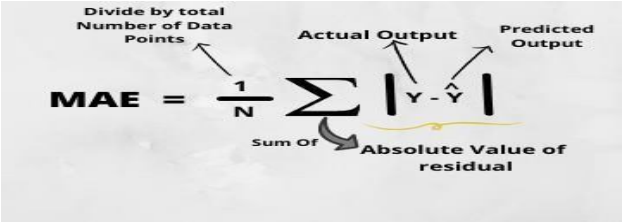
So to build and deploy a generalized model we require to Evaluate the model on different metrics.

Mean Absolute Error (MAE)

MAE is a very simple metric which calculates the absolute difference between actual and predicted values.

To better understand, let's take an example you have input data and output data and use Linear Regression, which draws a best-fit line.

Now you have to find the MAE of your model which is basically a mistake made by the model



known as an error.

Mean Squared Error (MSE)

MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

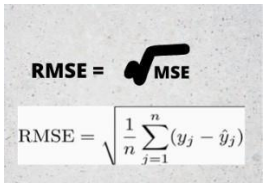
So, above we are finding the absolute difference and here we are finding the squared difference.

What actually the MSE represents? It represents the squared distance between actual and predicted values. we perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)}_{\text{The square of the difference between actual and predicted}}^{2}$$

Root Mean Squared Error (RMSE)

As RMSE is clear by the name itself, that it is a simple square root of mean squared error.



Root Mean Squared Log Error(RMSLE)

Taking the log of the RMSE metric slows down the scale of error. The metric is very helpful when you are developing a model without calling the inputs. In that case, the output will vary on a large scale.

To control this situation of RMSE we take the log of calculated RMSE error and resultant we get as RMSLE.

To perform RMSLE we have to use the NumPy log function over RMSE. print("RMSE",np.log(np.sqrt(mean_squared_error(y_test,y_pred))))

It is a very simple metric that is used by most of the datasets hosted for Machine Learning competitions.

R Squared (R2)

R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform.

In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context.

So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides. The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically R2 squared calculates how must regression line is better than a mean line.

$$\textbf{R2 Squared} = 1 - \frac{SSr}{SSm}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

Adjusted R Squared

The disadvantage of the R2 score is while adding new features in data the R2 score starts increasing or remains constant but it never decreases because It assumes that while adding more data variance of data increases.

But the problem is when we add an irrelevant feature in the dataset then at that time R2 sometimes starts increasing which is incorrect.

Hence, To control this situation Adjusted R Squared came into existence.

$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right) \times (1 - R^2)\right]$$

where:
n = number of observations
k = number of independent variables
R_a^2 = adjusted R^2

CONCLUSION:

We have implemented Regression Analysis with linear regression and ridge, Lasso regression models.

QUESTIONS:

1) What is regression analysis?

2) What is the difference between linear regression and logistic regression?

3) What is the purpose of a regression line?

4) How do you interpret the coefficients in a linear regression model?

ASSIGNMENT NO.3

Problem Statement:

Implementation of Support Vector Machines (SVM) for classifying images of handwritten digits into their respective numerical classes (0 to 9).

Tools / Environment:

Open Source Python, Programming tool like Jupyter Notebook,Pycharm, Spyder.
64-bit Open source Linux or its derivative

Related Theory:

For this problem, we use the MNIST data which is a large database of handwritten digits. The 'pixel values' of each digit (image) comprise the features, and the actual number between 0-9 is the label.

Since each image is of 28 x 28 pixels, and each pixel forms a feature, there are 784 features. MNIST digit recognition is a well-studied problem in the ML community, and people have trained numerous models (Neural Networks, SVMs, boosted trees etc.) achieving error rates as low as 0.23% (i.e. accuracy = 99.77%, with a convolutional neural network).
Before the popularity of neural networks, though, models such as SVMs and boosted trees were the state-of-the-art in such problems.
We'll first explore the dataset a bit, prepare it (scale etc.) and then experiment with linear and non-linear SVMs with various hyper parameters.

We'll divide the analysis into the following parts:

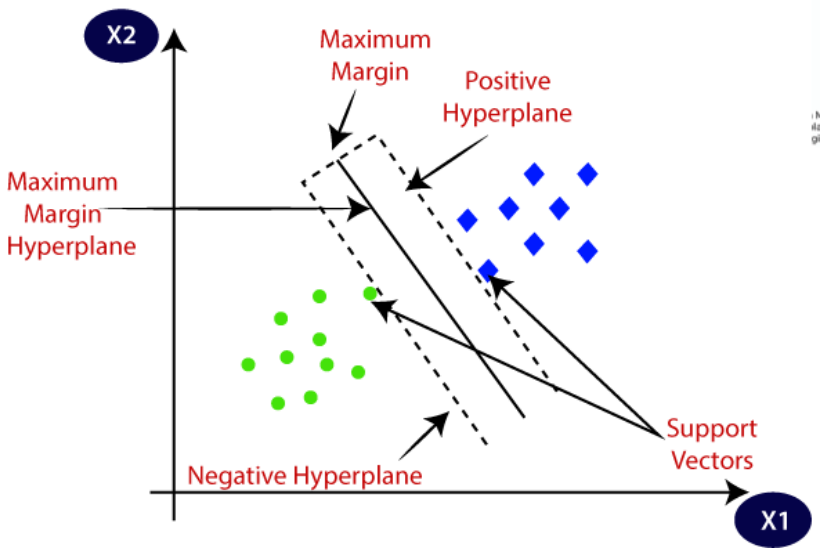
Data understanding and cleaning, Data preparation for model building, Building an SVM model - hyper parameter tuning, model evaluation etc.

SVM -

Support Vector Machine or SVM is a Supervised Learning algorithm, which is used for Classification and Regression problems. but it is mostly used for Classification problems in Machine Learning.

An SVM separates data across a decision boundary (plane) determined by only a small subset of the data (feature vectors). The data subset that supports the decision boundary is called the support vector. And this best decision boundary is called a hyper plane. Consider the below

diagram in which two different categories are classified using a decision boundary or hyperplane:



The distance between the decision–boundary and the closest data points is called the margin. SVM algorithm can be used for **image classification, Face detection, text categorization,** etc.**SVM can be of two types:**

- **Linear SVM:** if a dataset can be classified into two classes by using a single straight line, then such data is linearly separable data, and the classifier is used called a Linear SVM classifier. and the kernel is also used “linear”

- **Non-linear SVM:** if a dataset cannot be classified by using a straight line, then such data is non-linear data, and the classifier used is called a Non-linear SVM classifier. and the kernel is used “poly” or “rbf”.

Python Implementation of Support Vector Machine of the Image Classification problem

Now we will implement the SVM algorithm using Python. Here we will use the digit recognizerdata .you can download the data-set from the kaggle itself

We used 42000 samples and 28000 samples from the training and test data sets just to reduce thetime of computation

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as pltrain_df = pd.read_csv("train.csv")
test_df = pd.read_csv("test.csv")
```

Input the data

Extracting Independent and dependent Variable**.** Now we separate label and pixel columnsand, the label is the 1st column of the data frame.

```
X = train_df.drop('label', axis=1)
y = train_df['label']
```

Model Training

Then I have separate training and test data with 30% samples reserved for test data.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 100)
```

Data preprocessing

I have used StandardScaler which standardizes features by removing the mean and scaling it to unit variance. StandardScaler is that it will transform your data such that its distribution will havea mean value of 0 and a standard deviation of 1

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()# fit_transform use to do some calculation and then do transformation
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Model construct

Now the training set will be fitted to the SVM classifier and construct the model. To create

```
from sklearn.svm import SVC
rbf_model = SVC(kernel='rbf')
rbf_model.fit(X_train, y_train)
```

the SVM classifier, we will import the **SVC** class from s**klearn.svm** library. Below is the code for it:

In the above code, we have used **kernel='rbf'**, as here we are creating SVM for linearly separable data. and the Fit function is used to adjust weights according to data values

Model Testing

Now, we will predict the output for the test set. For this, we will create a new variable y_rbf_pred. Below is the code for it:

```
y_rbf_pred = rbf_model.predict(X_test)
```

After getting the y_rbf_pred vector, we can compare the result of **y_rbf_pred** and **y_test** tocheck the difference between the actual value and the predicted value.

```
print("Predictad Values :\n ",y_rbf_pred[10:15])
print ("Actual Values :\n",y_test[10:15])
```

Output:-

```
Predictad Values :
  [3 9 6 7 1]
Actual Values :
 24273    3
32691    9
34526    6
11625    7
6614     1
Name: label, dtype: int64
```

The accuracy of the model can be calculated using the **accuracy_score()** method from sklearn.metrics from sklearn import metrics

acc_rbf= metrics.accuracy_score(y_test,

y_rbf_pred)

print("accuracy:","{:.2f}".format(acc_rbf*100),"%")

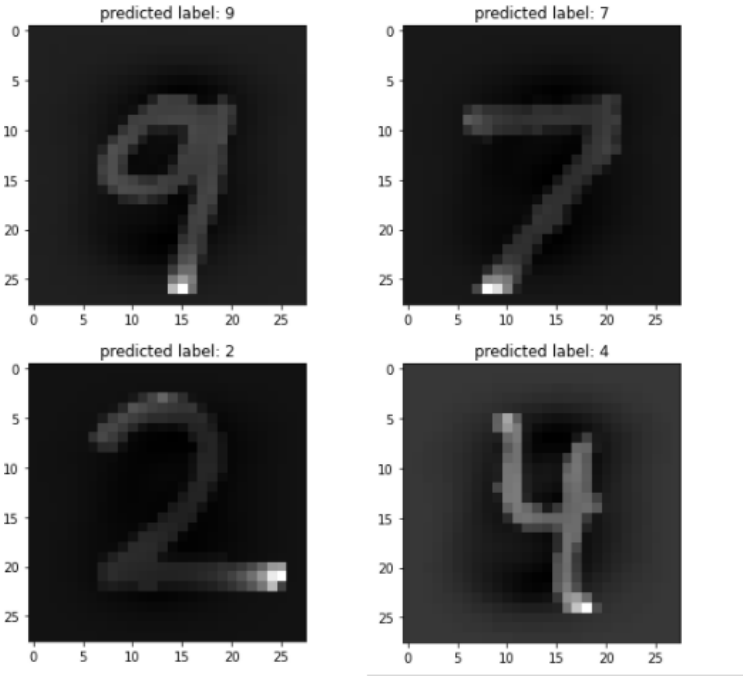
Output:- accuracy: 95.66 %

We can now plot the digits using python matplotlib.pyplot . We use the prediction list and thepixel values from the test list for comparison.

```
for i in (np.random.randint(0,270,4)):
   two_d = (np.reshape(X_test[i], (28, 28)))
   plt.title('predicted label: {0}'. format(y_rbf_pred[i])
   plt.imshow(two_d, cmap='gray')
   plt.show()
```

Let me briefly explain the second line of the code. As the pixel values are arranged in a row with 784 columns in the data set, first we use numpy 'reshape' module to arrange it in 28 X 28 array

Output:-



CONCLUSION:

Hence here we implementation of Support Vector Machines (SVM) for classifying images of handwritten digits into their respective numerical classes (0 to 9).

QUESTIONS:

1) What are support vectors in SVM?

2) What is the difference between a hard margin and a soft margin in SVM?

3) Name some common kernel functions used in SVM.

4) When would you use a linear kernel versus a non-linear kernel?

ASSIGNMENT NO.4

Problem Statement:

Implement K-Means clustering on Iris.csv dataset. Determine the number of clusters using the elbow method.

Tools / Environment:

Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder.
64-bit Open source Linux or its derivative

Related Theory:

What Is the Elbow Method in K-Means Clustering?

The elbow method is a graphical representation of finding the optimal 'K' in a K-means clustering. It works by finding WCSS (Within-Cluster Sum of Square) i.e. the sum of the square distance between points in a cluster and the cluster centroid.

Let's go through the steps involved in K-means clustering for a better understanding:

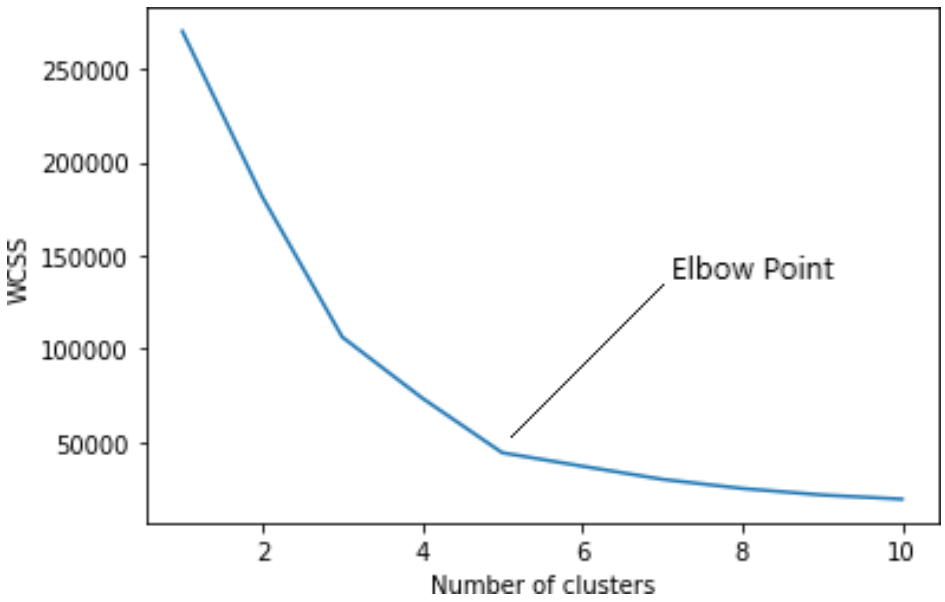
- Select the number of clusters for the dataset (K)
- Select the K number of centroids randomly from the dataset.
- Now we will use Euclidean distance or Manhattan distance as the metric to calculate the distance of the points from the nearest centroid and assign the points to that nearest cluster centroid, thus creating K clusters.
- Now we find the new centroid of the clusters thus formed.

● Again reassign the whole data point based on this new centroid, then repeat step 4. We will continue this for a given number of iterations until the position of the centroid doesn't change, i.e., there is no more convergence.

Finding the optimal number of clusters is an important part of this algorithm. A commonly used method for finding the optimum K value is Elbow Method.

K Means Clustering Using the Elbow Method

In the Elbow method, we are actually varying the number of clusters (K) from 1 – 10. For each value of K, we are calculating WCSS (Within-Cluster Sum of Square). WCSS is the sum of the squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1. When we analyze the graph, we can see that the graph will rapidly change at a point and thus creating an elbow shape. From this point, the graph moves almost parallel to the X-axis. The K value corresponding to this point is the optimal value of K or an optimal number of clusters.



Clustering analysis is a machine learning technique used to group similar data points together based on certain similarity or distance criteria. It is an unsupervised learning method where the algorithm identifies patterns or structures in the data without any predefined labels. Clustering can be useful for various tasks, such as customer segmentation, image compression, anomaly detection, and more.

CONCLUSION:

Here, we implement overview of clustering analysis and how to perform it using Python's scikit-learn library with the K-Means algorithm as an example with elbow method.

QUESTIONS:

1) How does the K-Means algorithm work?

2) What methods can be used to initialize the centroids in K-Means?

3) How do you choose the number of clusters (K) in K-Means?

4) What are some limitations of the K-Means algorithm?

ASSIGNMENT NO.5

Problem Statement:

Implement Random Forest Classifier model to predict the safety of the car.

Dataset link: https://www.kaggle.com/datasets/elikplim/car-evaluation-data-set

Tools / Environment:

Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder.
64-bit Open source Linux or its derivative

Related Theory:

1. Introduction to Random Forest algorithm

Random forest is a supervised learning algorithm. It has two variations – one is used for classification problems and other is used for regression problems. It is one of the most flexible and easy to use algorithm. It creates decision trees on the given data samples, gets prediction from each tree and selects the best solution by means of voting. It is also a pretty good indicator of feature importance.

Random forest algorithm combines multiple decision-trees, resulting in a forest of trees, hence the name Random Forest. In the random forest classifier, the higher the number of trees in the forest results in higher accuracy.

2. Random Forest algorithm intuition

Random forest algorithm intuition can be divided into two stages.

In the first stage, we randomly select "k" features out of total m features and build the random forest. In the first stage, we proceed as follows:-

1. Randomly select k features from a total of m features where k < m.
2. Among the k features, calculate the node d using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until l number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for n number of times to create n number of trees. In the second stage, we make predictions using the trained random forest algorithm.

1. We take the test features and use the rules of each randomly created decision tree to predict the outcome and store the predicted outcome.

2. Then, we calculate the votes for each predicted target.

3. Finally, we consider the high voted predicted target as the final prediction from the random forest algorithm.

Advantages and disadvantages of Random Forest algorithm

The advantages of Random forest algorithm are as follows:-

1. Random forest algorithm can be used to solve both classification and regression problems.

2. It is considered as very accurate and robust model because it uses large number of decision-trees to make predictions.

3. Random forests takes the average of all the predictions made by the decision-trees, which cancels out the biases. So, it does not suffer from the overfitting problem.

4. Random forest classifier can handle the missing values. There are two ways to handle the missing values. First is to use median values to replace continuous variables and second is to compute the proximity-weighted average of missing values.

5. Random forest classifier can be used for feature selection. It means selecting the most important features out of the available features from the training dataset.

The disadvantages of Random Forest algorithm are listed below:-

1. The biggest disadvantage of random forests is its computational complexity. Random forests are very slow in making predictions because large number of decision-trees is used to make predictions. All the trees in the forest have to make a prediction for the same input and then perform voting on it. So, it is a time-consuming process.

2. The model is difficult to interpret as compared to a decision-tree, where we can easily make a prediction as compared to a decision-tree.

CONCLUSION:

Thus, we have implemented Random Forest Classifier model to predict the safety of the car.

QUESTIONS:

1) How does the Random Forest algorithm work?

2) What are the key hyper parameters of a Random Forest Classifier?

3) Why does Random Forest use random subsets of features?

4) How would you explain the difference between Random Forest and a single decision tree to a non-technical person?

Assignment 6

Problem Statement:

Build a Tic-Tac-Toe game using reinforcement learning in Python by using following tasks

a. Setting up the environment

b. Defining the Tic-Tac-Toe game

c. Building the reinforcement learning model

d. Training the model

e. Testing the model

Tools / Environment:

Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder.

64-bit Open source Linux or its derivative

Related Theory:

Introduction-

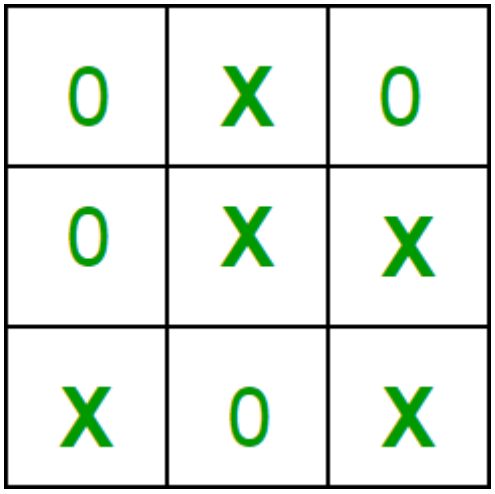
In this, we will learn how to create an agent that learns to play the game by trial and error, taking actions and receiving rewards or penalties depending on whether the action led to a win, loss, or draw.

Prerequisites-

- Basic knowledge of Python programming
- Familiarity with the fundamentals of machine learning, specifically reinforcement learning
- Familiarity with the TensorFlow library

Rules of the Game

● The game is to be played between two people (in this program between HUMAN and COMPUTER).
● One of the player chooses 'O' and the other 'X' to mark their respective cells.
● The game starts with one of the players and the game ends when one of the players has one whole row/ column/ diagonal filled with his/her respective character ('O' or 'X').
● If no one wins, then the game is said to be draw



Implementation In our program the moves taken by the computer and the human are chosen randomly. We use rand() function for this.

What more can be done in the program?
The program is in not played optimally by both sides because the moves are chosen randomly. The program can be easily modified so that both players play optimally (which will fall under the category of Artificial Intelligence). Also the program can be modified such that the user himself gives the input (using scanf() or cin). The above changes are left as an exercise to the readers.

Winning Strategy – An Interesting Fact

If both the players play optimally then it is destined that you will never lose ("although the match can still be drawn"). It doesn't matter whether you play first or second. In another ways – "Two expert players will always draw". Isn't this interesting?

CONCLUSION:

Thus, we have built a Tic-Tac-Toe game using reinforcement learning.

QUESTIONS:

1) What are the key components of a Reinforcement Learning system?

2) What is Reinforcement Learning?

3) What is the difference between model-free and model-based reinforcement learning?