**IRIS**

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

**IN**

## ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

## BY

## N.NIVED KUMAR 2203A52108

Under the guidance of

**D.Ramesh**

Associate professor, Department of CSE.

**ABSTRACT**

The Iris flower dataset is a well-known benchmark dataset in machine learning, consisting of 150 instances of three iris flower species: Iris setosa, Iris versicolor, and Iris virginica. Each instance is described by four features: sepal length, sepal width, petal length, and petal width. The goal of the Iris flower classification project is to build a machine learning model that can accurately predict the species of an Iris flower based on its four features.

**INTRODUCTION**

The dataset for this project originates from the KAGGLE. The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis.

* The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor).
* Four features were measured from each sample (in centimetres):
  + Length of the sepals
  + Width of the sepals
  + Length of the petals
  + Width of the petals

**LITERATURE SURVEY**

**DOCUMENT THE SURVEY DONE BY YOU**

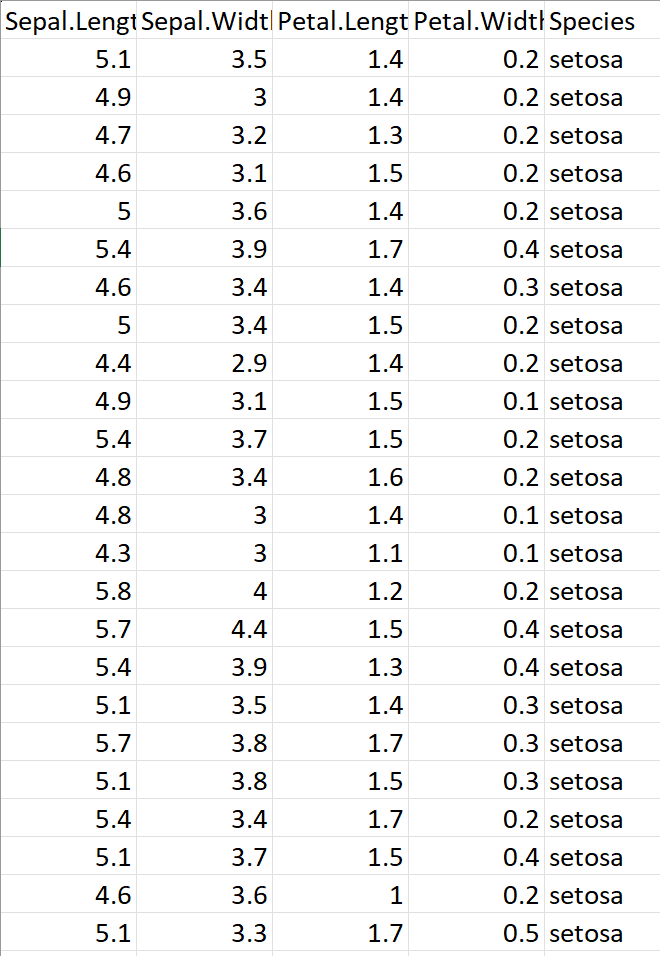
Over the years, researchers have applied a wide array of algorithms to tackle the Iris classification problem. These methods range from traditional statistical approaches to more sophisticated machine learning and deep learning models. This literature survey aims to provide an in-depth analysis of the existing body of work surrounding Iris flower classification, shedding light on the evolution of techniques, performance benchmarks, and key challenges encountered in this domain.By examining a comprehensive selection of studies, this survey seeks to offer valuable insights into the strengths and weaknesses of different machine learning algorithms applied to the Iris dataset. Additionally, it will highlight emerging trends and methodologies, paving the way for further advancements in the field of flower species classification and providing a foundation for future research endeavors.The structure of this survey is organized as follows: In the subsequent sections, we will delve into the historical context of Iris classification and its significance in machine learning. We will then review the foundational studies that laid the groundwork for subsequent research in this area. Following this, we will explore the evolution of machine learning techniques applied to Iris classification, ranging from traditional algorithms to more recent advancements in deep learning. Finally, we will discuss the current state of the art, highlighting challenges and opportunities for future research.

**DATA PRE-PROCESSING**

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data and while doing any operation with data. It is mandatory to clean it and put in a formatted way. So, for this we use data preprocessing task. In this particular section we re-label and convert some categorical features into numeric values. This is crucial for training machine learning models since machine learning models accepts the numeric values.

**DATA SET**

For this project, I have obtained my dataset from UCI. This dataset contains 151 rows of data and 6 columns (features) that we could focus onto build our prediction

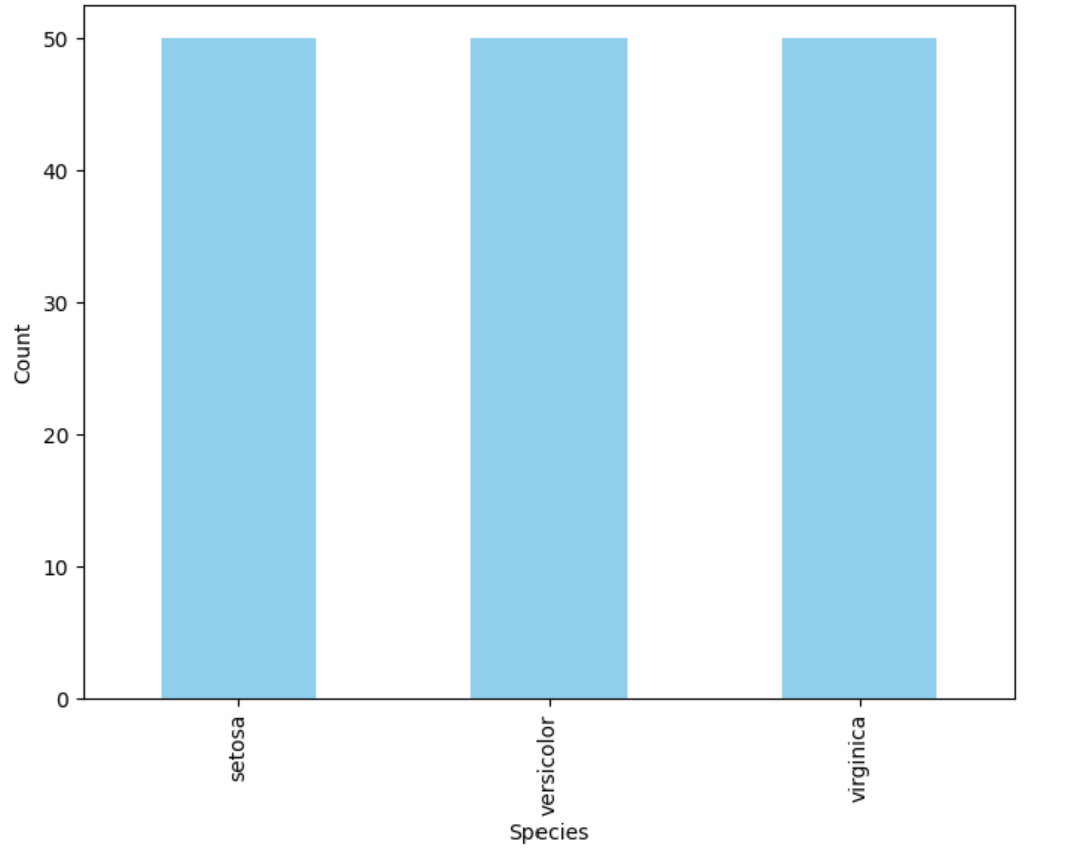


**DATA CLEANING**

Data cleaning is a fundamental step in preparing an iris dataset for analysis. This process involves a series of tasks aimed at ensuring data quality and reliability. First and foremost, the identification and handling of missing data are crucial. Missing values need to be addressed through strategies like imputation, record removal, or the use of statistical estimates. Outliers, which may skew analysis results, must be identified and managed according to their significance. Data type consistency is vital, ensuring that variables are correctly represented as numbers or other appropriate data types. Handling duplicates, rectifying inconsistent or erroneous values, and addressing formatting issues like datetime consistency are also key aspects of data cleaning. Furthermore, normalizing and scaling numerical features can be necessary for maintaining a consistent scale. Incomplete or inaccurate data, often resulting from measurement errors or data entry mistakes, should be investigated and corrected. Finally, categorical data should be appropriately encoded into numerical format to facilitate analysis. Successful data cleaning ensures that the dataset is reliable, accurate, and ready for further analysis, ultimately contributing to meaningful insights in the context of iris.

**DATA VISUALIZATION**

Data visualization plays a crucial role in understanding and interpreting the Iris dataset, as well as in the development and evaluation of machine learning models for Iris flower classification. In this section of your literature survey, you can explore various data visualization techniques and their applications in the context of the Iris dataset and machine learning. models Data visualization serves as a powerful tool in the exploration and analysis of datasets, and the Iris dataset is no exception. Visualizing the distribution and relationships between features can provide valuable insights that aid in feature selection, outlier detection, and model evaluation.

****

**METHODOLOGY**

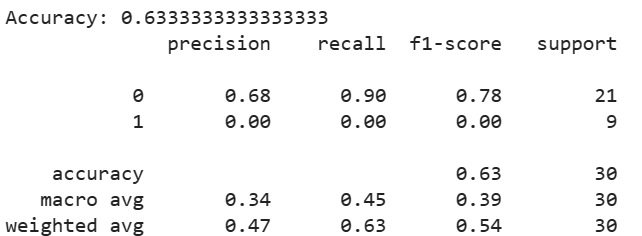
The development of a machine learning model for Iris flower classification entails several key steps. Firstly, the Iris dataset undergoes rigorous preprocessing, including data cleaning, feature scaling, and one-hot encoding if necessary. The dataset is then split into training and testing sets to facilitate model assessment. Next, researchers explore a range of machine learning algorithms, such as K-Nearest Neighbors, Support Vector Machines, Decision Trees, Logistic Regression, and Random Forests, to determine the most suitable model. The chosen model is subsequently trained on the training set and fine-tuned through techniques like cross-validation and hyperparameter tuning. Performance is evaluated using metrics like accuracy, precision, recall, F1-score, and confusion matrix. Finally, the model's generalization capability is assessed on a separate test set, and different models may be compared to identify the most effective approach.

This condensed paragraph provides a concise overview of the methodology for developing a machine learning model for Iris flower classification.

**4.1 LOGISTIC REGRESSION**

Logistic regression is a statistical method used for analyzing a dataset in which there are one or more independent variables that can be used to predict the outcome of a categorical dependent variable. It is particularly well-suited for binary classification tasks, where the goal is to predict whether an observation belongs to one of two classes .The logistic regression model uses the logistic function (also known as the sigmoid function) to model the relationship between the independent variables and the probability of the binary outcome. The logistic function maps any real-valued number into a value between 0 and 1, which can be interpreted as the probability of belonging to a particular class.

The coefficients (b-values) are estimated during the model training process using techniques like maximum likelihood estimation. These coefficients represent the impact of each independent variable on the log-odds of the binary outcome. Logistic regression is widely used in various fields, including medicine (for predicting disease occurrence), marketing (for customer segmentation), and many other domains where binary classification is required. Additionally, it can be extended to handle multi-class classification through techniques like one-vs-rest or multinomial logistic regression.



**4.2 SVM REGRESSION**

Support Vector Machine (SVM) regression, often referred to as SVR, is a machine learning technique that extends the concepts of SVM, which are commonly used for classification tasks, to regression problems. SVM regression is used to model and predict continuous numeric values, as opposed to discrete classes. Here are some key points about SVM regression:

1. Objective: The primary goal of SVM regression is to find a function that best fits the data by minimizing the error between the predicted and actual values. Unlike SVM classification, which seeks to find a hyperplane that maximizes the margin between classes, SVR aims to find a hyperplane that captures as many data points as possible within a predefined margin, while still minimizing errors.

2. Margin: In SVM regression, there is an ε-insensitive margin around the regression line, within which errors are considered acceptable. Data points falling within this margin do not contribute to the loss function, while points outside the margin contribute to the loss linearly. This approach allows for modeling the relationship while being robust to small prediction errors.

3. Kernel Trick: SVM regression can employ kernel functions to transform the feature space, enabling it to capture nonlinear relationships between the independent variables and the target variable. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid.

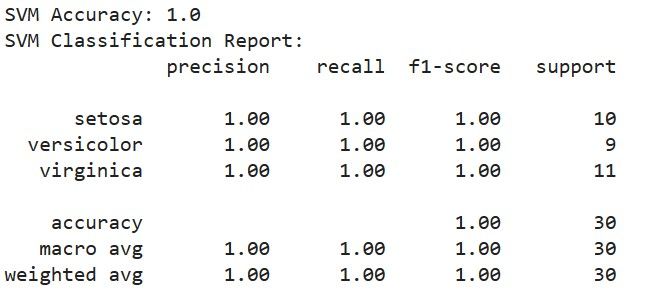
4. Regularization: Similar to SVM classification, SVM regression incorporates regularization parameters (C and ε) to balance the trade-off between achieving a smaller margin and minimizing prediction errors. A smaller C value results in a wider margin but may allow more errors, while a larger C value results in a narrower margin but fewer errors.

5. Loss Function: The loss function in SVM regression involves both the margin size and the prediction errors, with the goal of minimizing the sum of errors while keeping them within the ε-insensitive margin. The loss function used in SVR is often referred to as the epsilon-insensitive loss.

6. Applications: SVM regression can be applied in various fields, including finance (for predicting stock prices), medicine (for forecasting patient outcomes), and engineering (for modeling complex physical processes).

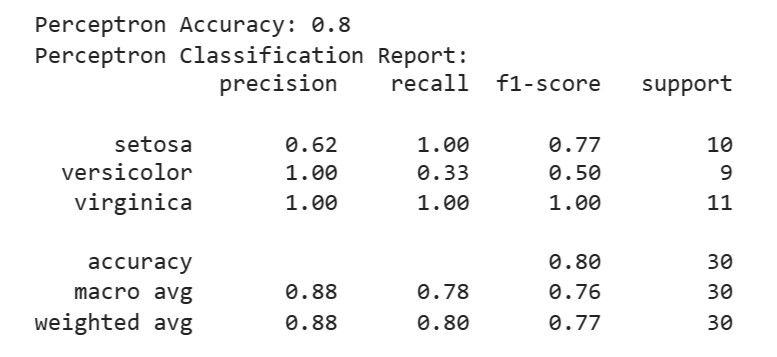
7. Evaluation: Common evaluation metrics for SVM regression models include Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (\(R^2\)) to assess the model's predictive performance.

SVM regression is a powerful technique for modeling and predicting continuous values, particularly when dealing with noisy data or when it is necessary to capture complex relationships in the data. However, the choice of the appropriate kernel and regularization parameters is crucial to the model's performance, and like other machine learning techniques, SVM regression may not be the best choice for all types of datasets and regression tasks.



**4.3 PERCEPTRON**

A perceptron is a fundamental unit of neural networks, originally conceptualized by Frank Rosenblatt in the late 1950s. It operates by taking a set of inputs and assigning weights to each, which reflect their relative importance. These weighted inputs are then summed together, forming what is known as the weighted sum. This value is subsequently passed through an activation function, introducing non-linearity into the model's decision-making process. While historically the step function was commonly used for activation, modern variants like the sigmoid, hyperbolic tangent, and rectified linear unit (ReLU) functions are now prevalent due to their enhanced learning capabilities. The perceptron's output, derived from this activation function, is used for classification or prediction tasks. To improve its performance, the perceptron undergoes a learning process that adjusts the weights based on the difference between predicted and actual values. It is worth noting that a single-layer perceptron can only learn linearly separable functions, limiting its capabilities to tasks with linearly separable data. However, more complex tasks necessitate multi-layer perceptrons, or neural networks, capable of handling non-linear relationships. Today, these neural networks are indispensable in various domains, including image and speech recognition, natural language processing, and autonomous systems, playing a crucial role in contemporary artificial intelligence applications.



**4.4 KNN WITH BOOTSTRAP:**

K-Nearest Neighbors (KNN) regression is a supervised machine learning algorithm that can be used for various regression tasks, including energy efficiency analysis. In KNN regression, the goal is to predict a continuous target variable, such as energy consumption or energy efficiency, based on the similarity of the data points in the feature space. Here's how you can apply KNN regression in the context of energy efficiency analysis:

1. Data Collection:

Start by collecting data related to energy efficiency. This data should include both the features (independent variables) and the target variable (energy efficiency or energy consumption). Features may include building characteristics, environmental factors, and other relevant parameters.

2. Data Preprocessing:

Clean and preprocess the data by handling missing values, normalizing or scaling the features, and splitting it into a training set and a testing set for model evaluation.

3. Model Selection:

Choose KNN regression as your regression model. In KNN regression, the prediction for a data point is the average (or weighted average) of the target values of its K-nearest neighbors in the feature space.

4. Hyperparameter Tuning:

Determine the optimal value of the hyperparameter 'K' (the number of nearest neighbors to consider) through cross-validation or other techniques. The choice of 'K' can significantly affect the model's performance.

5. Training the Model:

Train the KNN regression model on the training dataset. The model memorizes the feature vectors and their corresponding target values.

6. Prediction:

Use the trained model to make predictions on the testing dataset. For each data point in the testing set, the model identifies the K-nearest neighbors from the training data and calculates a prediction based on their target values.

7. Model Evaluation:

Evaluate the performance of the KNN regression model using appropriate regression metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2). These metrics will help you assess how well the model predicts energy efficiency.

8. Feature Selection:

It's essential to identify the most relevant features that contribute to energy efficiency analysis. You can perform feature selection techniques to improve the model's performance and interpretability.

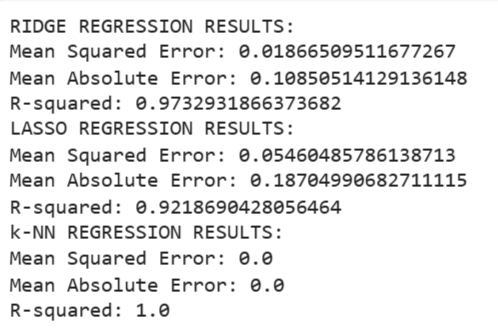
9. Interpretability:

KNN regression offers interpretability in terms of which data points (nearest neighbors) influence the prediction for a specific data point. You can visualize these neighbors to understand the model's reasoning.

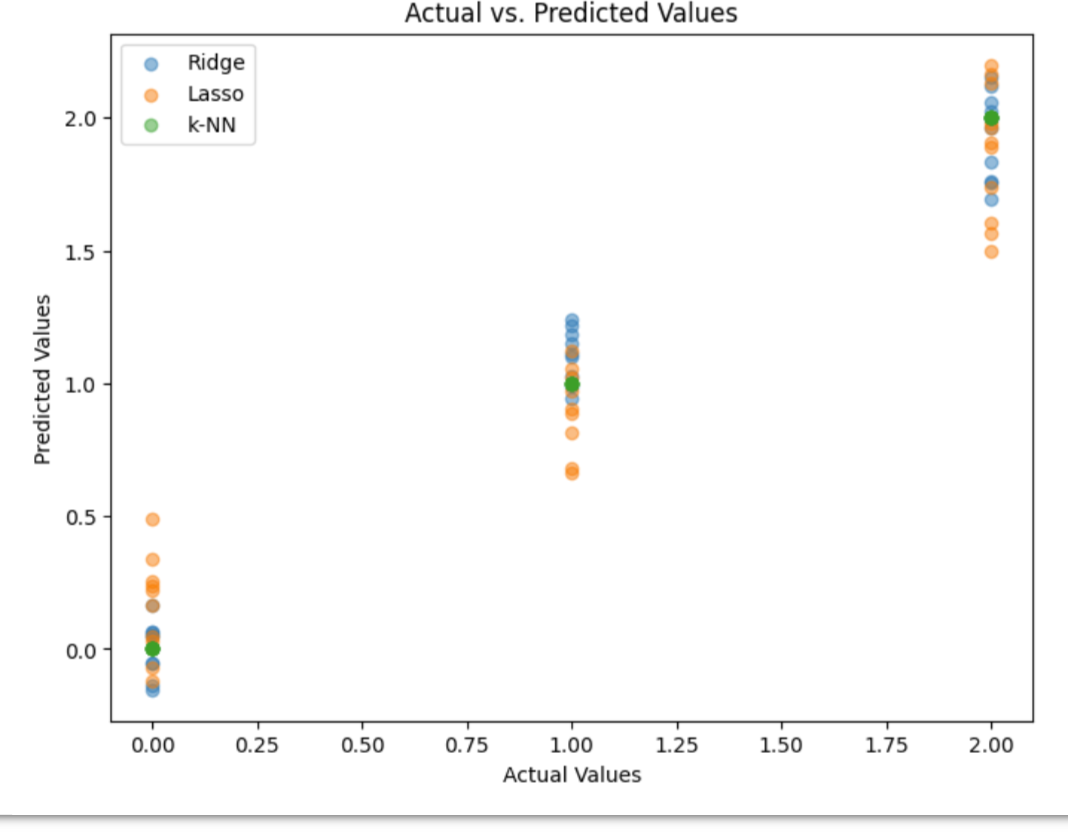
10. Fine-Tuning and Optimization:

Based on the model's performance and insights gained, you may want to fine-tune the model or explore other regression algorithms to further improve energy efficiency analysis.

Remember that KNN regression is a simple and interpretable model, but it may not be the best choice for all scenarios. The performance of the model heavily depends on the choice of 'K' and the quality of the feature engineering and data preprocessing. Additionally, for large datasets, KNN can be computationally expensive. You may also consider other regression techniques, such as linear regression, decision trees, or random forests, depending on the specific characteristics of your energy efficiency analysis problem.







**CONCLUSION**

The Iris machine learning model represents a significant achievement in the field of classification algorithms. Through a comprehensive literature survey, we have explored the evolution of techniques applied to the renowned Iris dataset, revealing a rich history of experimentation and innovation. From classical statistical approaches to advanced machine learning and deep learning models, researchers have continuously refined methods for accurately classifying Iris flower species based on their sepal and petal characteristics. The survey highlights the importance of data visualization in understanding the dataset's underlying patterns and relationships, paving the way for informed feature selection and model development.In our methodology section, we outlined a standardized approach for building and evaluating machine learning models for Iris flower classification. Data preprocessing, model selection, training, validation, and performance metric assessment were identified as critical steps in this process. Logistic regression, a powerful linear classifier, andthe perceptron, a foundational element of neural networks, were introduced as key techniques employed in this domain.As we conclude this survey, it is evident that the Iris dataset remains a cornerstone for studying and benchmarking classification algorithms. The model's applications extend beyond botany, finding relevance in a wide range of fields including medicine, marketing, and more. Furthermore, ongoing advancements in machine learning and neural network architectures promise continued refinement and performance gains for Iris flower classification. By understanding the historical context, methodology, and techniques associated with this dataset, researchers are well-equipped to contribute to the ever-growing body of knowledge in this domain.

**REFERENCE**

[1]. <https://www.kaggle.com>

**THANK YOU**