**PROJECT REPORT**

**Ajith Sivadas**

**SID- 1001829098**

**Problem Statement:**

To create a linear regression model that can classify different species of the Iris flower from the given dataset and predict the test dataset with maximum accuracy

**Problem Solving:**

The problem solving method which we will be using to implement for the given dataset is as follows,

* Create the dataset
* Build the model
* Train the model
* Make predictions
* Compare the predictions with two different models (Linear regression )

1. **Create the dataset**

In-order to classify the dataset the different species, initially, we need to prepare the labels and features of the Iris flower. However, **sklearn** comes with an in built libraries for the iris classification problem. We do not use sklearn here, instead we use **K-Fold** to split the dataset first, then train and test, and finally, cross validate the dataset.

To make it more clear, let’s understand that the dataset consists of:

* **150 instances**
* **3 labels: Species of Iris (*Iris setosa, Iris virginica, Iris versicolor*)**
* **4 features: Sepal length, Sepal width, Petal length, Petal width**

**Table

Description automatically generated**

1. **Build the model**

We have a training set in machine learning, which is made up of attributes (inputs, independent variables) and labels (response, target, dependent variables). We use an algorithm to train a set of models with different hyperparameter values and then choose the model that best minimizes a cost (a loss or objective) function. We will use k-fold technique to feed the model for that we will randomly choose the indices and split the main data into multiple training sets and testing sets according to the number of folds.

In machine learning, linear regression is a supervised learning method derived from classical statistic. However, since neural networks with linear (multilayer perception) layers perform regression, their use has risen in tandem with the rapid rise of machine learning and deep learning. This regression is normally linear, but when non-linear activation functions are used, it becomes non-linear.

To solve the problem, we use classification algorithms. We use the Linear Regression formula to train the dataset with the beta value since it is supervised learning. I have used numpy to write the code to predict it.

Shape

Description automatically generated with medium confidence

The above formula is implemented for linear regression in the code below

Graphical user interface, text, application

Description automatically generated

Another technique to optimize the results is Gradient Descent by calculating and minimizing the error. The change of all weights in relation to the change in error is determined by a gradient. A gradient is also known as the slope of a function. The steeper the slope and the faster a model can learn, the higher the gradient. If the slope is zero, however, the model will stop learning. A gradient is a partial derivative with respect to its inputs in mathematics.

A picture containing text, clock, watch, gauge

Description automatically generated

Gradient descent is defined by the equation below: b represents our climber's next position, while a represents his current position. The minus sign denotes the aspect of gradient descent that requires minimization. The middle gamma is a waiting element, and the gradient term ( f(a) ) is literally the steepest descent path.

Graphical user interface, text, application

Description automatically generated

1. **Train the model**

For training the dataset, we will be having train and test for both x and y variable as x\_train, x\_test and y\_train, y\_test.

A subset which is used to train the model is indicated as **training set** and a subset which is used to test the trained model is called as **test data.**

To make sure that our test sets meet the following requirements, always check the following:

1. If it is large enough to produce the statistically meaningful results.
2. If it is representative of the dataset as a whole. In other words, do not choose a test set with different characteristics than the training set.

Your aim is to construct a model that generalizes well to new data, assuming your test set meets the two conditions stated above

The process of predicting the class of given data points is known as Classification. Targets/ labels or divisions are other names for groups. The role of approximating a mapping function (f) from discrete input variables (x) to classification predictive modeling is known as classification predictive modeling (y).

Classification is a form of supervised learning in which the objective are also given the input data.

A statistical approach for estimating the ability of machine learning models is cross validation. It is widely used in applied machine learning to compare and choose a model for a given predictive modeling problem because it is simple to understand, execute, and produces skill elements with lower bias than other approaches.

The method of k-fold cross validation is used to estimate the model’s ability on new data. You may use a number of methods to choose the value of k for your dataset. In scikit-learn, there are many widely used cross-validation variants, such as stratified and repeated.

The k-fold is done here with the aid of the code below:

Diagram

Description automatically generated

**Graphical user interface, text, application

Description automatically generated**

1. **Predictions**

We are calculating accuracies for 3 fold, 5 fold and 7 fold with different seed value and the output is as follows:

**A picture containing text, receipt

Description automatically generated**

**A picture containing text, receipt

Description automatically generated**

The accuracy improvement while using gradient descent algorithm is as follows:

Table

Description automatically generated with medium confidence

The error Vs Iterations Graph is as follows for 3 fold

Graphical user interface, application

Description automatically generated\

The error vs iteration clearly tells us that when the number of iteration increase the cost aka the error decreases which is beneficial to predict the training model at much higher accuracy.

**Conclusion:**

From the above accuracies and the graph, it is clear that by using the gradient descent method, there are significant improvement in accuracy. It gives slightly higher accuracies than the previous one. The change in the beta value in gradient descent helps the model to predict at a much higher accuracy.

**References:**

<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html>

<http://ethen8181.github.io/machine-learning/model_selection/model_selection.html>

<https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3>