Student Name: Chinnala Sathish Kumar

Student ID: 1002050768

MACHINE LEARNING - CSE 6363-005 - PROJECT 1

Linear Regression

Linear regression is a statistical method used for analyzing the relationship between a dependent variable and one or more independent variables. It is a supervised learning algorithm that is used to model the linear relationship between an independent variable (x) and a dependent variable (y). The goal of linear regression is to find the best-fitting line that represents the relationship between the dependent and independent variables.

The equation of the line is given by:

$$y = b0 + b1 * x$$

Simple Regression

Simple linear regression employs the standard slope-intercept form, with ‘m’ and ‘c’ denoting the variables that our system will attempt to "learn" in order to give the most accurate predictions. Our input data is denoted by the letter x, and our prediction is denoted by the letter y.

$$y=mx+c$$

Simple linear regression is useful for understanding the relationship between two variables and making predictions about the value of the dependent variable based on the value of the independent variable.

Multivariate Regression:

A more complicated, multi-variable linear equation looks like as follows, where w is the coefficients, or weights, that our model will attempt to learn.

$$f(x, y, z)=w1x+w2y+w3z$$

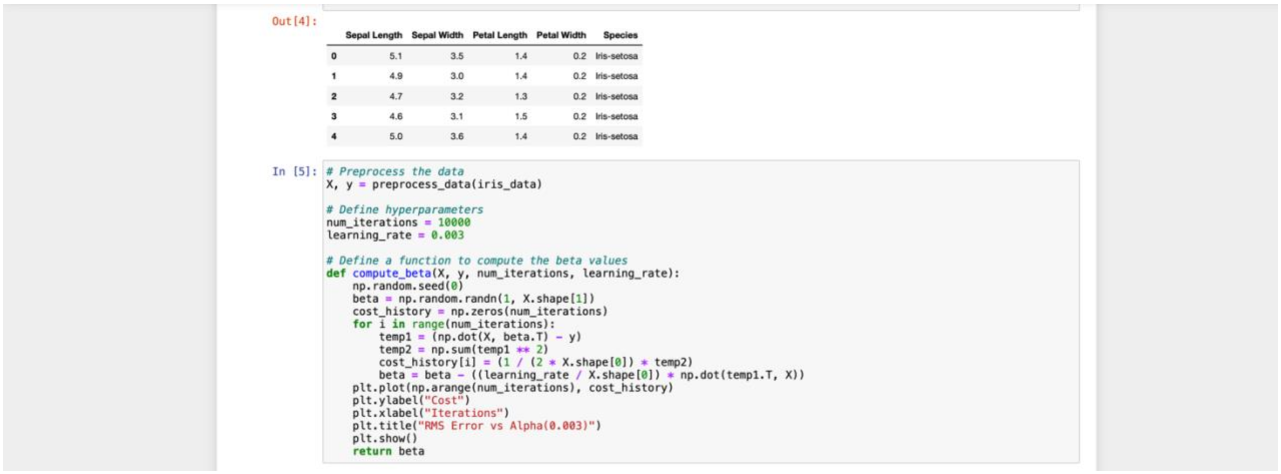
To train the model and perform classification, we need to utilize multivariate linear regression with cross-validation. There are a total of 5 columns in this data. The values of the fifth column, the label column will be determined by the value of the four feature columns

Iris-setosa, Iris-versicolor, and Iris-virginica are the three types of values in the label column.

Data Preprocessing:

Data preprocessing is an important step in the data analysis process, and it refers to the cleaning, transforming, and normalizing of the data prior to using it for modeling and analysis. The main goals of data preprocessing are to make the data consistent, remove any irrelevant or missing information, and ensure that the data is suitable for analysis and modeling

Data Preprocessing is the process of transforming, or encoding, data so that it may be easily parsed by the machine. First, we'll highlight and label the dataset. As a result, features are saved in X while labels are



Out[4]:

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [5]:

```python
# Preprocess the data
X, y = preprocess_data(iris_data)

# Define hyperparameters
num_iterations = 10000
learning_rate = 0.003

# Define a function to compute the beta values
def compute_beta(X, y, num_iterations, learning_rate):
    np.random.seed(0)
    beta = np.random.randn(1, X.shape[1])
    cost_history = np.zeros(num_iterations)
    for i in range(num_iterations):
        temp1 = (np.dot(X, beta.T) - y)
        temp2 = np.sum(temp1 ** 2)
        cost_history[i] = (1 / (2 * X.shape[0]) * temp2)
        beta = beta - ((learning_rate / X.shape[0]) * np.dot(temp1.T, X))
    plt.plot(np.arange(num_iterations), cost_history)
    plt.ylabel("Cost")
    plt.xlabel("Iterations")
    plt.title("RMS Error vs Alpha(0.003)")
    plt.show()
    return beta
```

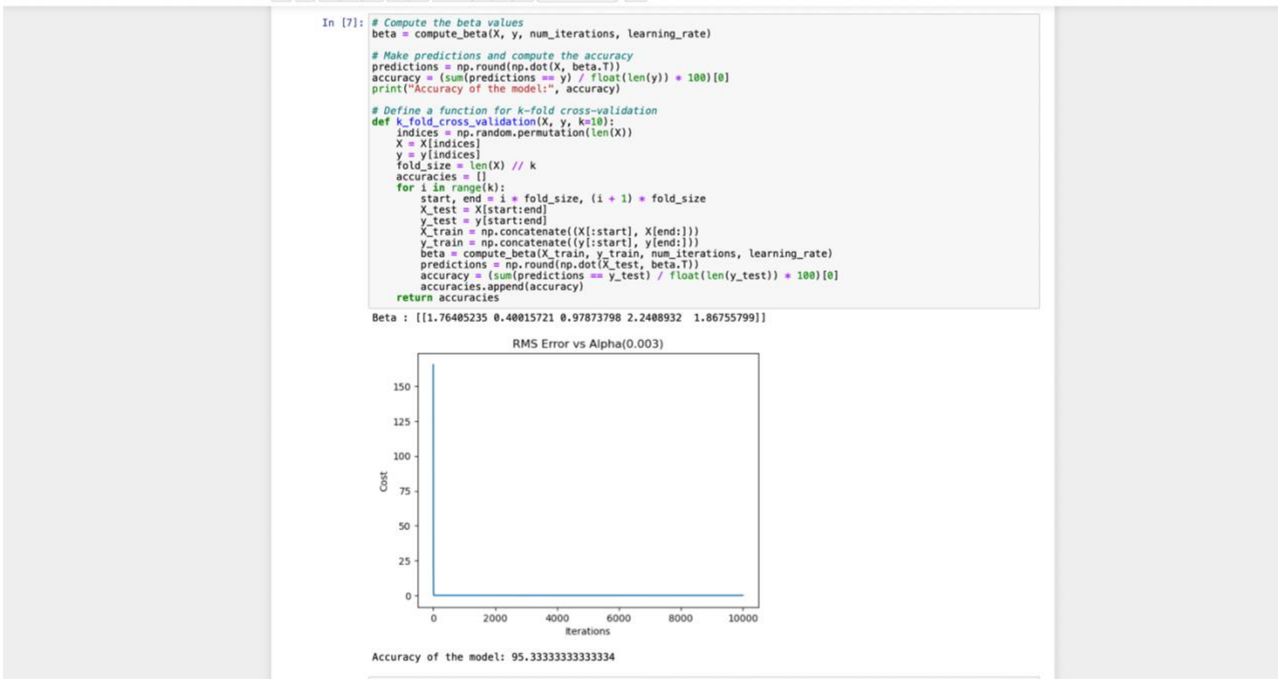
As a result, features are saved in X while labels are saved in Y. Because linear regression is a mathematical model, any categorical data in the label should be avoided. As a result, in our Y, we'll manually encode Iris-setosa, Iris-versicolor, and Iris-virginica into the integer values 1,2,3.

Training a model:

The process of training a model involves teaching it to identify patterns in the provided data, allowing it to make predictions on new, unseen data. In this case, we opt for the multivariate linear regression technique as it can handle multiple feature columns.

Cross validation:

Cross-validation is a technique for evaluating the performance of a machine learning model by partitioning the dataset into complementary subsets, training the model on one subset and evaluating it on the other, and repeating this process multiple times to ensure that the model is not overfitting to the data. The accuracy of the designed module is provided.



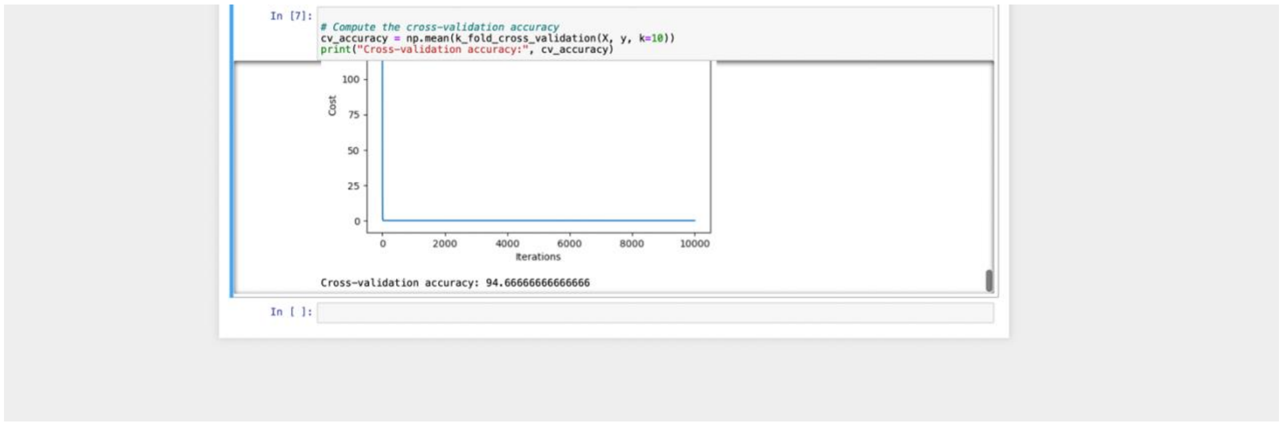
```
In [7]: # Compute the beta values
        beta = compute_beta(X, y, num_iterations, learning_rate)

        # Make predictions and compute the accuracy
        predictions = np.round(np.dot(X, beta.T))
        accuracy = (sum(predictions == y) / float(len(y)) * 100)[0]
        print("Accuracy of the model:", accuracy)

        # Define a function for k-fold cross-validation
        def k_fold_cross_validation(X, y, k=10):
            indices = np.random.permutation(len(X))
            X = X[indices]
            y = y[indices]
            fold_size = len(X) // k
            accuracies = []
            for i in range(k):
                start, end = i * fold_size, (i + 1) * fold_size
                X_test = X[start:end]
                y_test = y[start:end]
                X_train = np.concatenate((X[:start], X[end:]))
                y_train = np.concatenate((y[:start], y[end:]))
                beta = compute_beta(X_train, y_train, num_iterations, learning_rate)
                predictions = np.round(np.dot(X_test, beta.T))
                accuracy = (sum(predictions == y_test) / float(len(y_test)) * 100)[0]
                accuracies.append(accuracy)
            return accuracies
```

```
Beta : [[1.76405235 0.40015721 0.97873798 2.2408932  1.86755799]]
```

```
Accuracy of the model: 95.33333333333334
```


```
In [7]:
# Compute the cross-validation accuracy
cv_accuracy = np.mean(k_fold_cross_validation(X, y, k=10))
print("Cross-validation accuracy:", cv_accuracy)
```

```
Cross-validation accuracy: 94.66666666666666
```

```
In [ ]:
```

Results:

The developed model is capable of predicting the test data’s class. The model has been well trained with training data and is capable of producing results with test data. The accuracy is increases with the beta value. The accuracy of the model is developed and also the cross validation accuracy is produced.

References:

https://machinelearningmastery.com/k-fold-cross-validation/

https://scikit-learn.org/stable/modules/cross_validation.html