Water Quality Prediction

INTRODUCTION

We all know water is one of the most essential resource for our living. We can't survive without water. But as the development is increasing, we are exploiting water by wasting it and treating it with harmful materials which makes water impure and unfit for use. This is the reason it is very important to know the quality of water. This project is based on water quality prediction. In this project, water quality index (WQI) quality status of water is predicted through some parameters that affects water quality.

1. Data Source

The data has been taken from https://data.gov.in/ which is the official website for data related to India. It contained data of different rivers, for example https://data.gov.in/resources/water-quality-river-brahmaputra-2013 this link contains from where the river Brahmaputra is flowing and what are the values of all the important factors that affect water quality at a particular area. There are 8 parameters out of which 6 are the most important parameters and that have been used to predict. Similarly, there are data for different rivers which I have downloaded and combined them so that the model works more efficiently.

1.1 Justification

I have chosen https://data.gov.in/ because it is the official and trusted website for the data related to India. Also, the data is not fake and it will give more accurate and better result.

2. Feature Creation

2.1 Data Cleansing

2.1.1 Conversion of datatypes

The data which has been uploaded contains all the values in string data type.

But for the calculation of water quality index we need the data in float format so for that we converted all the required columns in the float data type.

2.1.2 Emptiness

The data which has been uploaded has few null values in all those required columns which should not be present as it will reduce the accuracy of our model. So, we will remove all those rows which has any null value in it.

2.2 Calculation of water quality index

The data which we want to predict is not already present in the data. So, before we create a model to train, we need what we want to train. So, now we will calculate WQI.
The WQI has been calculated by using the standards of drinking water quality recommended by the World Health Organisation (WHO), Bureau of Indian Standards (BIS) and Indian Council for Medical Research (ICMR).
The Water Quality Index was calculated by aggregating the quality rating with the weight linearly,

$$WQI=\sum (q_n \times W_n)$$

where q_n =Quality rating for the nth Water quality parameter,
W_n= unit weight for the nth parameters.
Although for calculation q_n we have standard formula but it was not possible in this case, so we applied a standard method for calculating quality rating for each parameter. Whereas W_n is the standard value which shows how much that parameter affects the WQI and has the fixed value.

So, first we calculated quality rating of each of the parameter. Then we multiplied all the quality rating with its weight and summed all the values and got the value of WQI for each row.

2.3 Status of water quality

According to the water quality index the status of water is given as: -

WATER QUALITY INDEX LEVEL	WATER QUALITY STATUS
0-25	Excellent water quality
26-50	Good water quality
51-75	Poor water quality
76-100	Very Poor water quality
>100	Unsuitable for drinking

3. Model Definition

Now it's time to create our model and predict our data using machine learning and deep learning algorithm. The data which we have is in supervised context and so we have chosen algorithm accordingly. There are three models in our project. Each of them are explained below: -

3.1 Non-Deep Learning Based Linear Regression Model

The first model which have been created is non-deep learning algorithm based linear regression model. A linear regression model is a model in which the output quantity has a linear relationship with some parameters. All those parameters are multiplied by their weight and are added to produce the output.

Here first we converted our data which are required to predict WQI into vector form by using VectorAssembler. Then we normalize our data by using Normalizer and at last we import LinearRegression from pyspark.ml.regression and applied it to our normalized data. Afterthat, we import Pipeline from pyspark.ml and include all those steps in the pipeline that we have done.

3.1.1 Justification

I am creating a Linear Regression model because I have to predict a continuous value and not a discrete or binary value. Also, data have a linear relationship between the input parameters and output. Also, the model is working very well with a very high accuracy.

3.2 Deep Learning Based Linear Regression Model

The second model which have been created is deep learning algorithm based linear regression model. Here, we are using Keras to create a model. In Keras there are two types of model available, Sequential and Non-Sequenetial. Here we are creating a Sequential Model. As we all know each model has a bunch of layers so the layers in my model are: -

There are 4 dense layers in the model. The first layer has given with output length 350 and the input-shape is 6 with activation function relu. The second and third layer has given with output length 350 and activation function relu. The last layer has given with 1 output class and activation function is linear.

3.2.1 Justification

I have chosen Keras because it has an intuitive high-level API that makes it very easy to create a model. Also, the model works very well with a very high accuracy and there is no overfitting also.

3.3 Logistic Regression Model

The last model which have been created is a non-deep learning algorithm based logistic regression model. Here the water quality is classified according to their WQI value.

In this, first the quality column in data contains value in string format so that column is converted and each string value is assigned a unique value using StringIndexer. Then we converted our data which are required to predict WQI into vector form by using VectorAssembler. Then we normalize our data by using Normalizer and at last we import LogisticRegression from pyspark.ml.classification and applied it to our normalized data. Afterthat, we import Pipeline from pyspark.ml and include all those steps in the pipeline that we have done.

3.3.1 Justification

I am using logistic regression model because I need to predict a discrete value and not a continuous value. Also, the model performs very well in predicting the quality of the water.

4. Model Training

Before training our data, first data is randomly split in two parts i.e. train and test data. This is done so as to reduce the chances of overfitting. Then according to the model it is trained as explained below.

4.1 Non-Deep Learning Based Linear Regression Model

This model is trained very easily. First, we use our created pipeline and fit it to the train data and then the model is transformed. This way training is done in non-deep learning algorithms.

4.2 Deep Learning Based Linear Regression Model

In this before training our model we first need to compile our model. Here, the model is compiled with loss function = 'mean_squared_error', optimizer = 'Adam' and evaluation metrics = 'mse'. Afterwards, we fit our model to the training data that we provide with the required number of epochs and batch size. This way training is done in deep learning algorithms.

4.3 Logistic Regression Model

This model is trained similarly like non-deep learning based linear regression model. First, we use our created pipeline and fit it to the train data and then the model is transformed. This way training is done.

5. Model Evaluation

The model evaluation is one of the most important step as this tells us how accurate our model is working.

5.1 Non-Deep Learning Based Linear Regression Model

The model is evaluated by using R-squared method and the accuracy is displayed. The result comes out to be 0.9710 which means our model is working with 97% accuracy.

5.2 Deep Learning Based Linear Regression Model

Here, the graph of mean squared error is plotted which shows that the value of mean squared error after 50 epochs is almost 0.0030 which means our model is working properly. Also, the graph between actual and predicted data is plotted which shows that almost all the values have been predicted correct and there is no overfitting.

5.3 Logistic Regression Model

The model is evaluated by using Multiclass specific evaluator and we get the accuracy of our model. The result comes out to be 0.945 which means our model is working with 94% accuracy.