UNIVERSITY OF ILLINOIS AT SPRINGFIELD

COURSE 570 BIG DATA ANALYTICS

Submitted by: -

Gaurav Karale(gkara2)

Gaurav Khandave(gkhan6)

Under the guidance: -

Dr. Elham Khorasani

A PROJECT REPORT

ON

Predictive analysis of data on water pumps from The Tanzanian Ministry of Water

Introduction: -

This project is part of curriculum for course Big data. In this project we used free data provided by Tanzanian ministry of government. Data was about survey made by Taarifa about water pump in Tanzania. Data file has 56000 rows and 41 columns for training set and 11200 rows and 40 columns for test file.

We used random forest classifier as our data was categorical. Spark mlib library is used as it implements random forest using the existing decision tree implementation. Random forest algorithm's pipeline, label indexer, feature indexer, label convertor and model fit this are used for prediction. For performance evaluation multi class classification evaluator is used. We have to predict three prediction labels that is functional, non-functional and function but needs repair that's why we used multi class classification evaluator.

Problem definition and project goal: -

The purpose of this project was to predict the pump which are functional, which need some repair and which don't work at all based on a number of variables. Main goal was to predict the operating condition of a water pump for each record in the dataset.

Data analysis: -

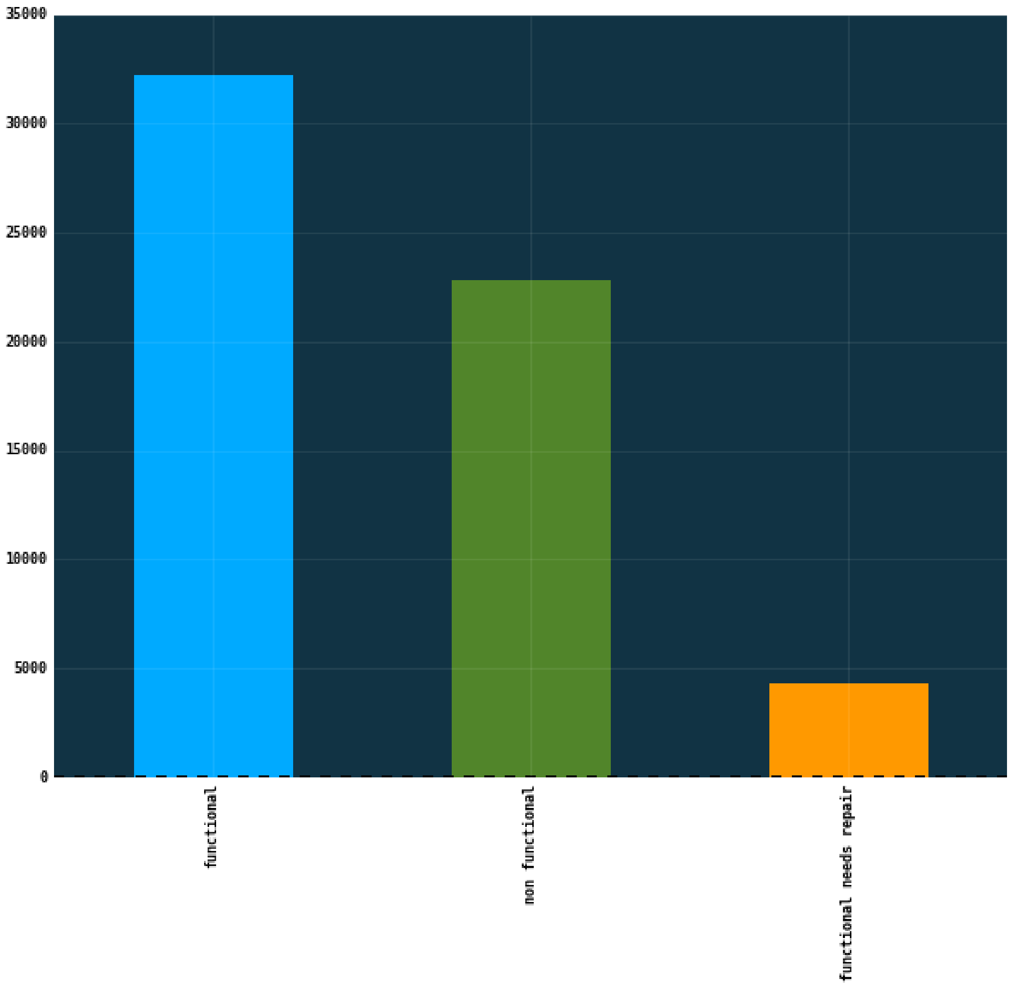
Exploratory data analysis on water pump data

For this we use ipython notebook as it provides better functions as well as graph plot features on the data. Graphical presentation gave feature importance of data. This analysis helped us in extracting categorical columns from data.

- amount_tsh - Total static head (amount water available to waterpoint)
- date_recorded - The date the row was entered
- funder - Who funded the well
- gps_height - Altitude of the well
- installer - Organization that installed the well
- longitude - GPS coordinate
- latitude - GPS coordinate
- wpt_name - Name of the waterpoint if there is one
- num_private -
- basin - Geographic water basin
- subvillage - Geographic location
- region - Geographic location
- region_code - Geographic location (coded)
- district_code - Geographic location (coded)
- lga - Geographic location
- ward - Geographic location
- population - Population around the well
- public_meeting - True/False
- recorded_by - Group entering this row of data
- scheme_management - Who operates the waterpoint
- scheme_name - Who operates the waterpoint
- permit - If the waterpoint is permitted
- construction_year - Year the waterpoint was constructed
- extraction_type - The kind of extraction the waterpoint uses
- extraction_type_group - The kind of extraction the waterpoint uses
- extraction_type_class - The kind of extraction the waterpoint uses
- management - How the waterpoint is managed
- management_group - How the waterpoint is managed
- payment - What the water costs
- payment_type - What the water costs
- water_quality - The quality of the water

- quality_group - The quality of the water
- quantity - The quantity of water
- quantity_group - The quantity of water
- source - The source of the water
- source_type - The source of the water
- source_class - The source of the water
- waterpoint_type - The kind of waterpoint
- waterpoint_type_group - The kind of waterpoint
- status_group- prediction label

The labels in this dataset (data distribution in training data)



Experimental results: -

1) Getting dummies for categorical data: -
   To convert categorical column to its equivalent dummy value which should of type double, we used MurmurHash scala utility.

2) Vector assembler: -
   We combine given input columns into desired output columns that is features.

3) Label indexer: -
   As prediction labels were of type string we converted them to equivalent index values using string indexer API.

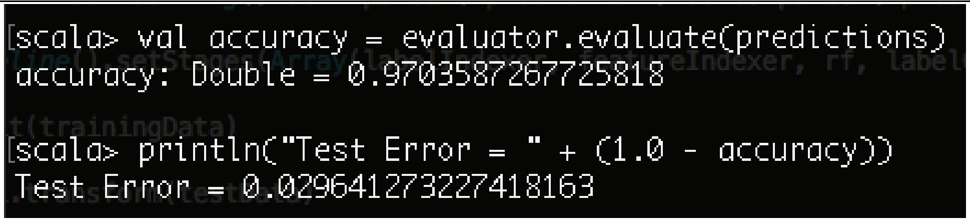
4) Feature indexer: -
   In this step we did indexing of feature.

5) Data split: -
   We split data into training and test set in the ratio of 70 and 30 percent respectively.

6) Random forest classification: -
   We used pipeline feature of spark ML to set stages, random forest classifier was one of the stages in pipeline. We used 100 trees for classification, we trained the model on training dataset and tested it on test data set.

7) Evaluation: -
   Multiclass Classification evaluator is used for evaluating the result as predication labels were multiclass.

8) Accuracy: -
   Accuracy we got by following above steps was about 0.97 i.e. 97%.



```
[scala> val accuracy = evaluator.evaluate(predictions)
 accuracy: Double = 0.9703587267725818

[scala> println("Test Error = " + (1.0 - accuracy))
 Test Error = 0.029641273227418163
```

Result and Conclusion: -

In this report we explained the methodology and procedures for predictive analysis of categorical data using Spark MLIb. And explored random forest classification on data provided by Tanzanian ministry of government. We successfully predicted data with 97 percent accuracy.

References: -

- https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/
- https://spark.apache.org/docs/2.1.0/mllib-ensembles.html
- https://ipython.org/ipython-doc/3/notebook/