AI in Recruitment

Megathon 2018

Manojit Chakraborty Nilabja Bhattacharya Sayan Ghosh Shubham Das

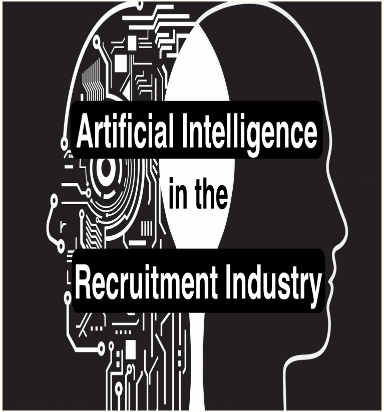
International Institute of Information Technology, Hyderabad

M.Tech Computer Science

Group Name : **Illuminati**

Introduction

- AI for recruiting is a data driven HR technology designed to reduce, or even remove time consuming activities of the recruitment process.
- This new technology is designed to streamline or automate some part of the recruiting workflow, especially repetitive, high-volume tasks.
- It improves quality of hire through standardized job matching and automating high-volume tasks
- It objectively assess a candidate's ability and skills while removing the inherent biases found throughout the sourcing and selection process.



AI helps in recruitment

- Machine Learning for Resume Parsing
- Detecting Fraud Clicks on Ads
- Chatbots for easy job matching
- Recommendation System for job assignment

Our Challenge

Problem Statement

Identify Bot Clicks from a real-world Ad Clicks dataset.

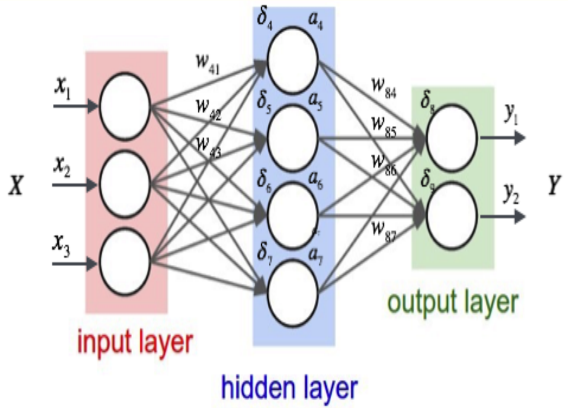
Relevance

Ad fraud is particularly important for marketers to understand. If you don't detect and avoid fraud, it will poison all other areas of optimization: context (brand safety), viewability and performance

Approach Outline

- A real world dataset containing bot click information
- Preprocessing of the raw data-set
- Training the preprocessed dataset using deep neural network
- Calculating accuracy of the trained machine learning model from the test dataset

Language(s) Used : Python (Pandas, Scikit-learn, Keras)



Input Dataset

- The input dataset used in our algorithm contains 1 million entries of Ad clicks.
- The dataset has a total of 17 attributes. Some of them are as follows :
  - *Clicks, BotClicks, LatentClicks*
  - *Title, Category, City, State*
  - *Device, Operating System, IP, Location*
- The attribute **botClicks** is the Class Label for our machine learning algorithm.

Preprocessing

This is one of the most important steps for a machine learning algorithm to achieve better accuracy. The preprocessing steps for the input dataset is as follows :

- Drop all the rows which contains NULL values.
- Drop the columns that are unnecessary for our algorithm. We dropped *userAgent*, *eventId*, *publisher*, *operatingSystem*, *clicks* columns.
- The idea behind this is, we want to focus on these aspects :
  - **Number of clicks/Unique IP addresses**
  - **Number of clicks/Unique devices**
  - **Number of clicks within a short timestamp interval**
- Convert all the categorical attributes to numerical coding.
- Standard scaling of the attributes in order to reduce noisy training data in the dataset.

Machine Learning Algorithm

For training the preprocessed dataset containing 827528 rows and 11 columns, we used the following algorithm :

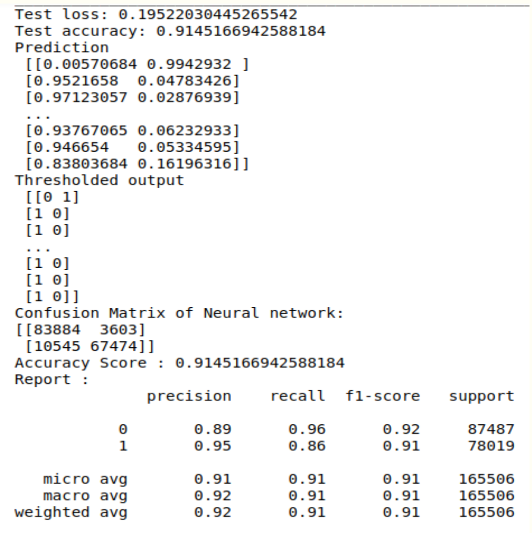
- **Support Vector Machine** with RBF (Radial Basis Function) Kernel using Scikit-learn.

- A **3 hidden layer Deep Neural Network** using Keras (which uses Tensorflow in the backend)
- Input layer contains 11 neurons, each of the 3 next hidden layers contain 10,9 and 9 neurons respectively, and Output layer contains 2 neurons for 2 output classes (0 and 1)
- Hidden layer neurons have ***RELU*** activation function. Output layer neurons have ***Softmax*** activation function.
- We used **Batch Processing System** in order to reduce time complexity (256 rows per batch).

Results

- First, we did a 50% random sampling on the dataset and ran SVM algorithm with RBF Kernel to achieve an **Accuracy Score of 90.46%**
- In order to achieve more accuracy score, we used the **Deep Neural Network** mentioned before, on the whole dataset.
- **This algorithm gives a much better accuracy score of *91.45%***, which is a pretty high value with respect to this dataset.
- Let's look at the output more clearly.

Output Screenshot



```
Test loss: 0.19522030445265542
Test accuracy: 0.9145166942588184
Prediction
 [[0.00570684 0.9942932 ]
 [0.9521658  0.04783426]
 [0.97123057 0.02876939]
 ...
 [0.93767065 0.06232933]
 [0.946654   0.05334595]
 [0.83803684 0.16196316]]
Thresholded output
 [[0 1]
 [1 0]
 [1 0]
 ...
 [1 0]
 [1 0]
 [1 0]]
Confusion Matrix of Neural network:
[[83884  3603]
 [10545 67474]]
Accuracy Score : 0.9145166942588184
Report :
              precision    recall  f1-score   support

           0       0.89      0.96      0.92     87487
           1       0.95      0.86      0.91     78019

   micro avg       0.91      0.91      0.91    165506
   macro avg       0.92      0.91      0.91    165506
weighted avg       0.92      0.91      0.91    165506
```

Thank you