Hyper-parameter Optimization through Reinforcement Learning

1 Problem Statement

Hyper-parameter Optimization through reinforcement learning

2 Explanation

In this problem we are using reward and action approach of reinforcement learning. Here increment in metric value corresponds to reward where as change in hyper-parameter are due to action taken.

3 Literature Review

Hyper-parameters in an algorithm are those parameters which remains unchanged during algorithm run. Its one of challenging task to get these parameters which suites are data. There are some optimization technique like grid search for these optimization but it takes lot of run time.

Random search is modified version of grid search where to reduce number of runs we constrain on time or iteration. One more method for this is bayesian search where we try to converse on best set of hyper-parameter using Gaussian process but this model also have its own hyper-paramter. Keep all scenarios in mind here we applied reinforcement learning approach to overcome above mentioned limitations.

4 Data

We are using tiktok dataset. Below is info of data

1. Columns = 9
2. Rows = 958
3. target_values = 0 or 1

5 Deliverable

To make a model which will give best set of hyper-parameter using reinforcement learning concept.

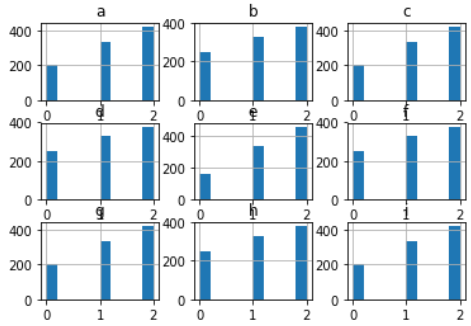
6 Evaluation

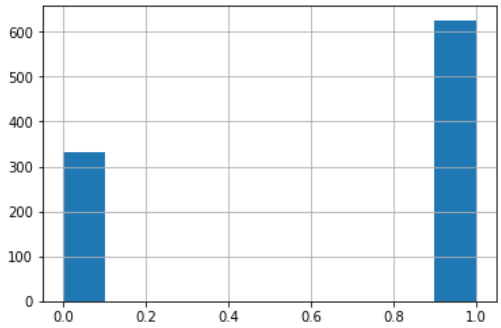
Accuracy metric is used.

7 Data Ingestion

We are reading data using pandas which give pandas data frame.

8 Data Analysis





Seeing above plot for features and target that data is balanced with ratio of 35:65.

	a	b	c	d	e	f	g	h	i
count	958.000000	958.000000	958.000000	958.000000	958.000000	958.000000	958.000000	958.000000	958.000000
mean	1.222338	1.133612	1.222338	1.133612	1.311065	1.133612	1.222338	1.133612	1.222338
std	0.775569	0.798966	0.775569	0.798966	0.740882	0.798966	0.775569	0.798966	0.775569
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000	1.000000
50%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
75%	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
max	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

9 Data Munging

Not Required.

10 Data Exploration

Not Required.

11 Feature Engineering

Not Required.

12 Modeling

Feed-forward Neural network is applied with below options.

1. Model with 1 layer
2. Model with 2 layer
3. Layer 1 range 4 to 10

13 Optimization

Award and action concept used for optimization.

14 Prediction

NA

15 Visual Analysis

Summary of model:

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 6)                 60
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 7
_________________________________________________________________
activation_4 (Activation)    (None, 1)                 0
=================================================================
```

16 Results

Best model is
1. model with 2 layer
2. number of node in layer1 is 6.