DEFCON Level

Using Neural Networks..!

By Ranjith KS

“The Defense Readiness Condition (DEFCON) is an *Alert state* used by the *United States Armed Forces.*"

-- Joint Chiefs of Staff and Unified & Specified Combatant Commands.

INTRODUCTION

Revealing key points.

What is?

DEFCONs are a subsystem of a series of Alert Conditions, or LERTCONs, that also includes Emergency Conditions (EMERGCONs).

Who Commands?

DEFCONs vary between many commands and have changed over time, and the United States Department of Defense uses exercise terms when referring to the DEFCONs. This is to preclude possibility of confusing exercise commands with actual operational commands.

Which Level?

The DEFCON scale is a way of assigning a numerical value to the readiness of the American military. Higher DEFCON values are used for lower levels of readiness (during more peaceful situations), while Lower DEFCON values are used for higher levels of readiness.

PREPARE

As a Data Scientist, I would perform EDA on the Dataset to understand various Features, relations and how to Model it.

"What am I doing with Dataset?"

Given the Dataset of various combinations of Parameters along with it's DefCon Level, I am going to predict the DefCon Level for a Test set.

"Predict the Target i.e. DefCon Level"

The DefCon Level varies in the range of 1 to 5, hence it's a Multi Classification Target.

"Apply Deep Learning Techniques"

Using the Neural Networks in Deep Learning, I would prepare a ANN Model that takes inputs from the given Features and predict the Target Feature.

How I Start?

To Model the Dataset, I have applied Exploratory Data Analysis, Basic Statistics, Visualization and etc.

!

Operations

1. Understand the Dataset
2. Visualize the Dataset Features and Columns Relations
3. Perform Data Cleanup and Exploration
4. Apply Feature Engineering
5. Figure out the Insights
6. Modeling using Deep Learning NN

DEFCON Level Reference Chart

1	2	3	4	5
Maximum readiness; All forces ready for combat; nuclear war imminent or likely	High readiness; Armed forces ready to deploy in six hours	Armed forces readiness increased above normal levels; Air Force ready to mobilize in 15 minutes	Increased intelligence-gathering and security measures	Normal peacetime readiness

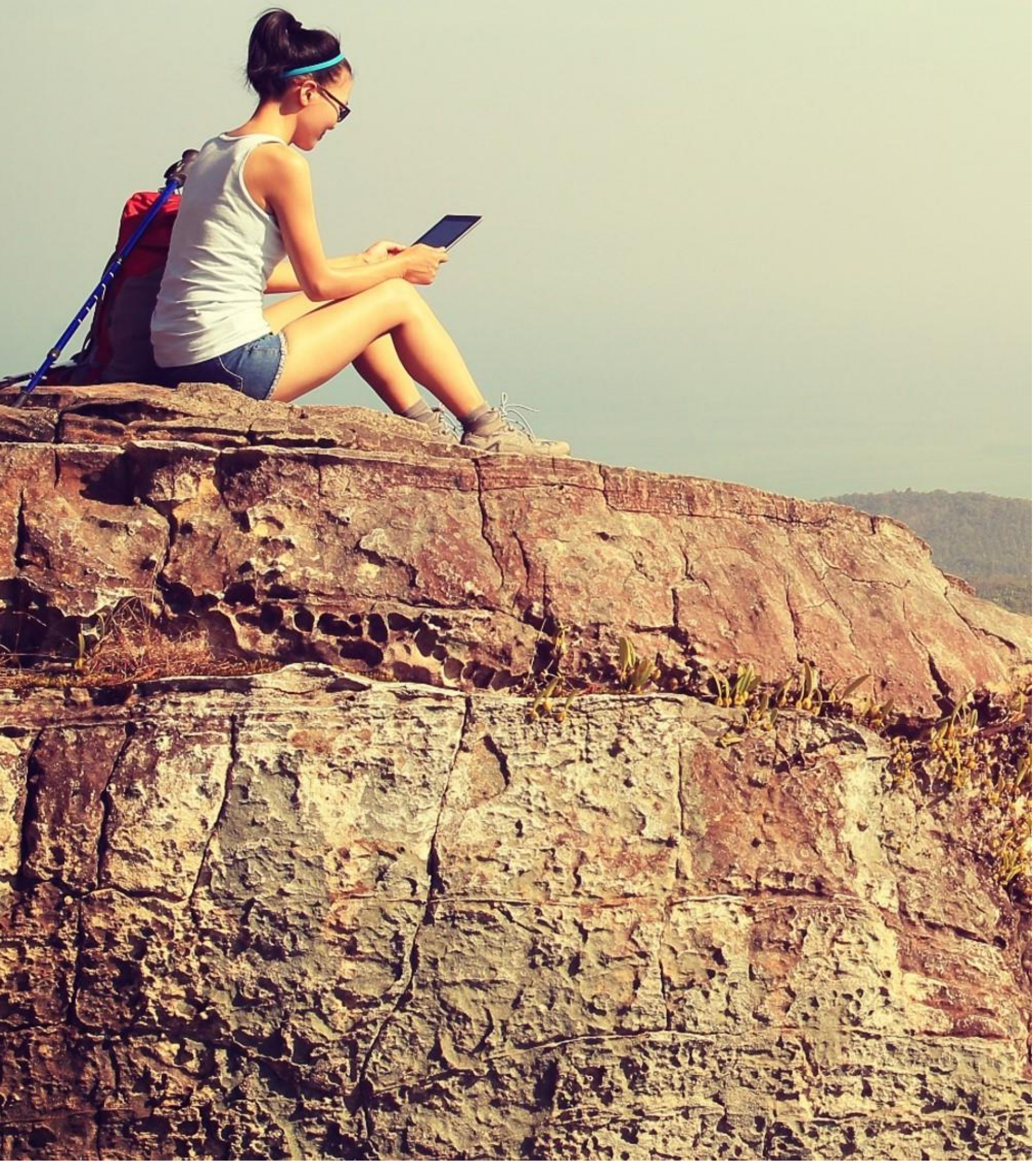
Reference : The basics you can find anywhere 5 Steps To Successful Storytelling Published on April 5, 2014 Featured in: Marketing & Advertising

5

A very good thing, used to designate normal peacetime military readiness

PEACE TIME

Note that DEFCON 5 isn't necessarily a sign that the world is at peace - conflicts, even major ones, may be occurring around the world during a DEFCON 5.



4

First level of readiness above baseline value of DEFCON 5 and thus constitutes a fairly mild increase in readiness

Heightened Alertness

In the modern world, it is thought that DEFCON 4 is sometimes issued after minor to moderate terrorist attacks and politically-motivated killings, or after would-be plots are uncovered

3

It’s a serious - though they may not pose an immediate threat to existence or stability of American state, they do call for significant vigilance

Tense Military/Political situations

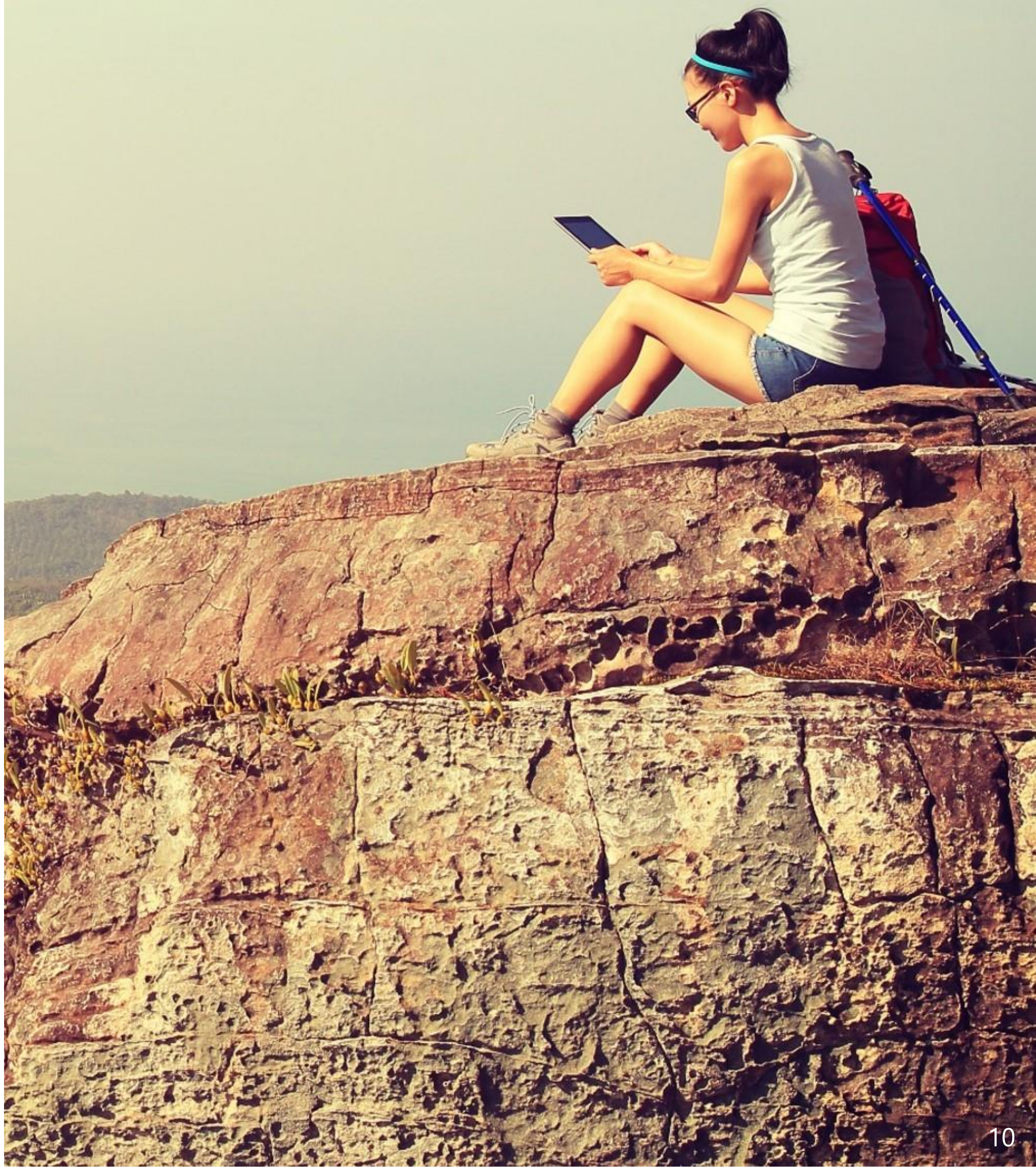
DEFCON 3 has usually corresponded to situations in which military action against the US or one of its allies was a distinct possibility

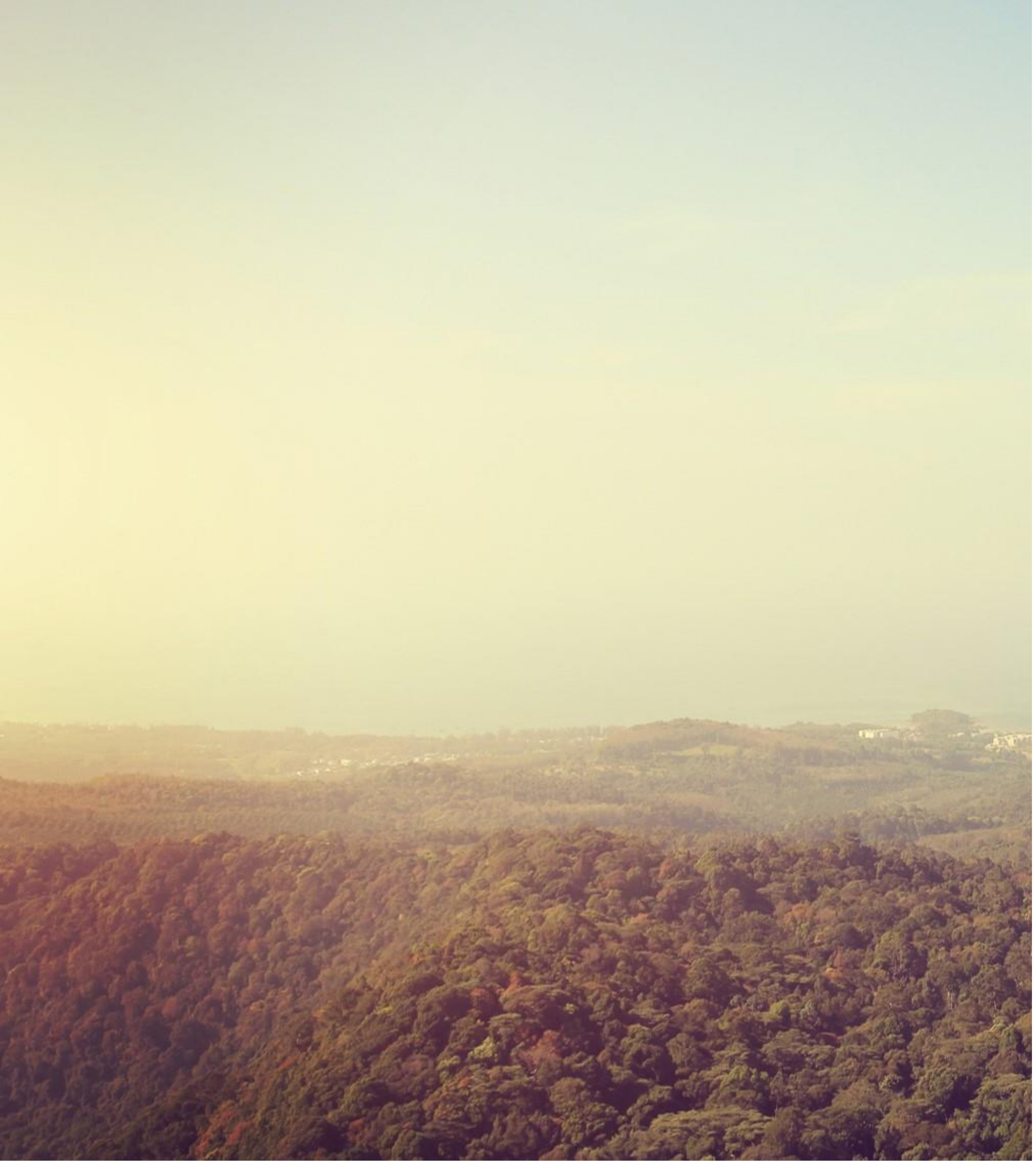
2

Refers to a further increase in force readiness just below maximum readiness

MAJOR THREAT

situations carry a significant risk of major military operations against the United States or its allies, including the use of nuclear weapons



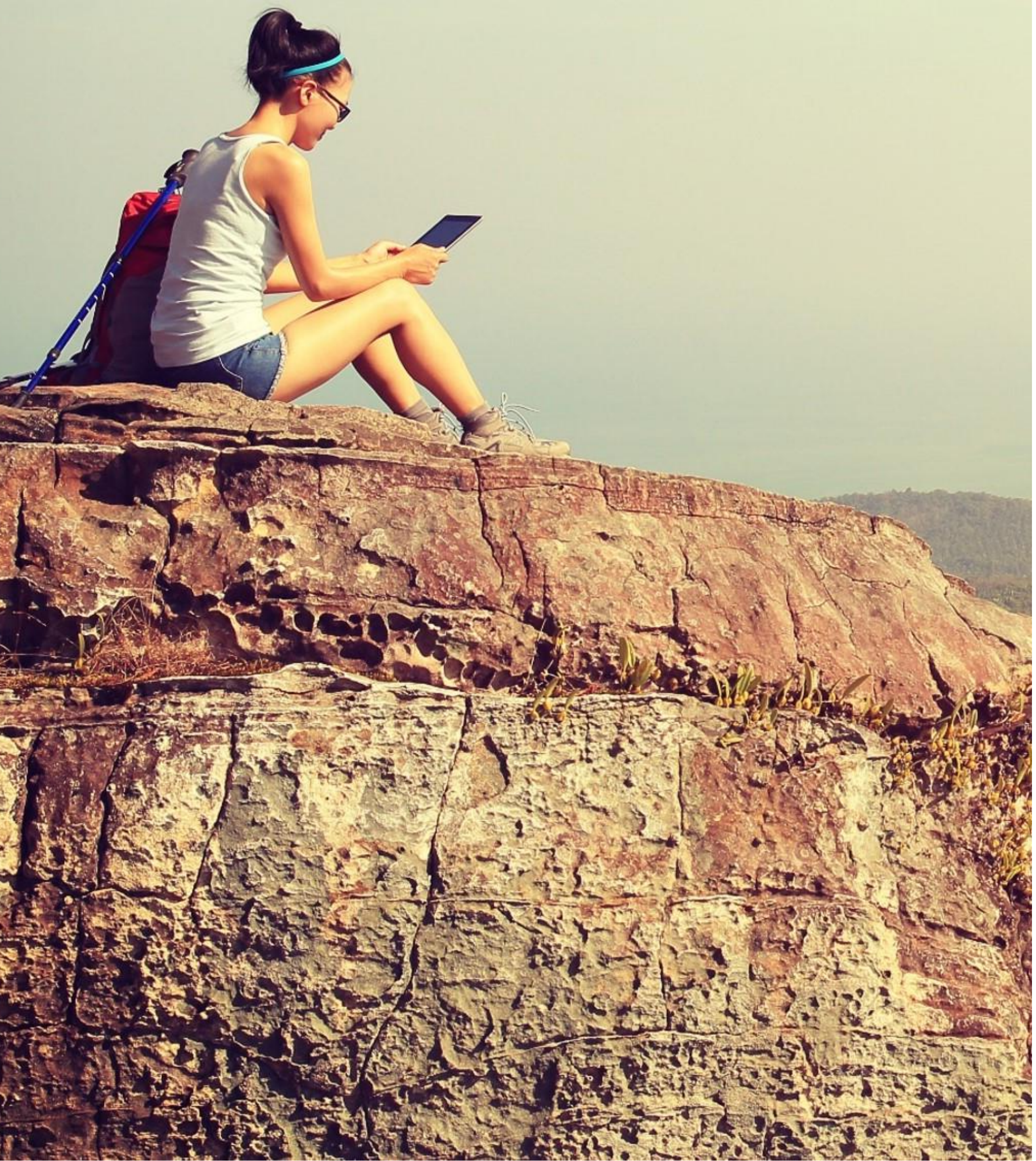


1

It is reserved for absolute most dangerous, severe situations, including ongoing nuclear war involving US or one of its allies

MAXIMUM ALERT

Levels are usually kept classified until after the fact, it is thought that DEFCON 1 has never been issued for a branch of the US military before.



TAKEAWAY

DefCon Level 1 is rarely is given.

LEARN FROM OTHERS

DefCon Level 5 is for Peace, and every Nation should have Friendship and positive Relationship among other Nations.

ACTION

A Common Agreement has to made without any Conflicts and Negative Face.

Given the Dataset with various Features and Entries,
what am I visualizing and predicting?

VISUALIZATION

Extracting the valuable Insights from the Dataset.

Variables

Dataset has 10000 Rows with 12 Features where 10 are Independent Features,1 Dependent Feature and an ID Feature which is ignored.

Describe

Dataset has Integer and Floating Point Features, does not have EMPTY values i.e. NaN and most of the Features are range in a limited values expect Troops_Mobilized Feature.

Features

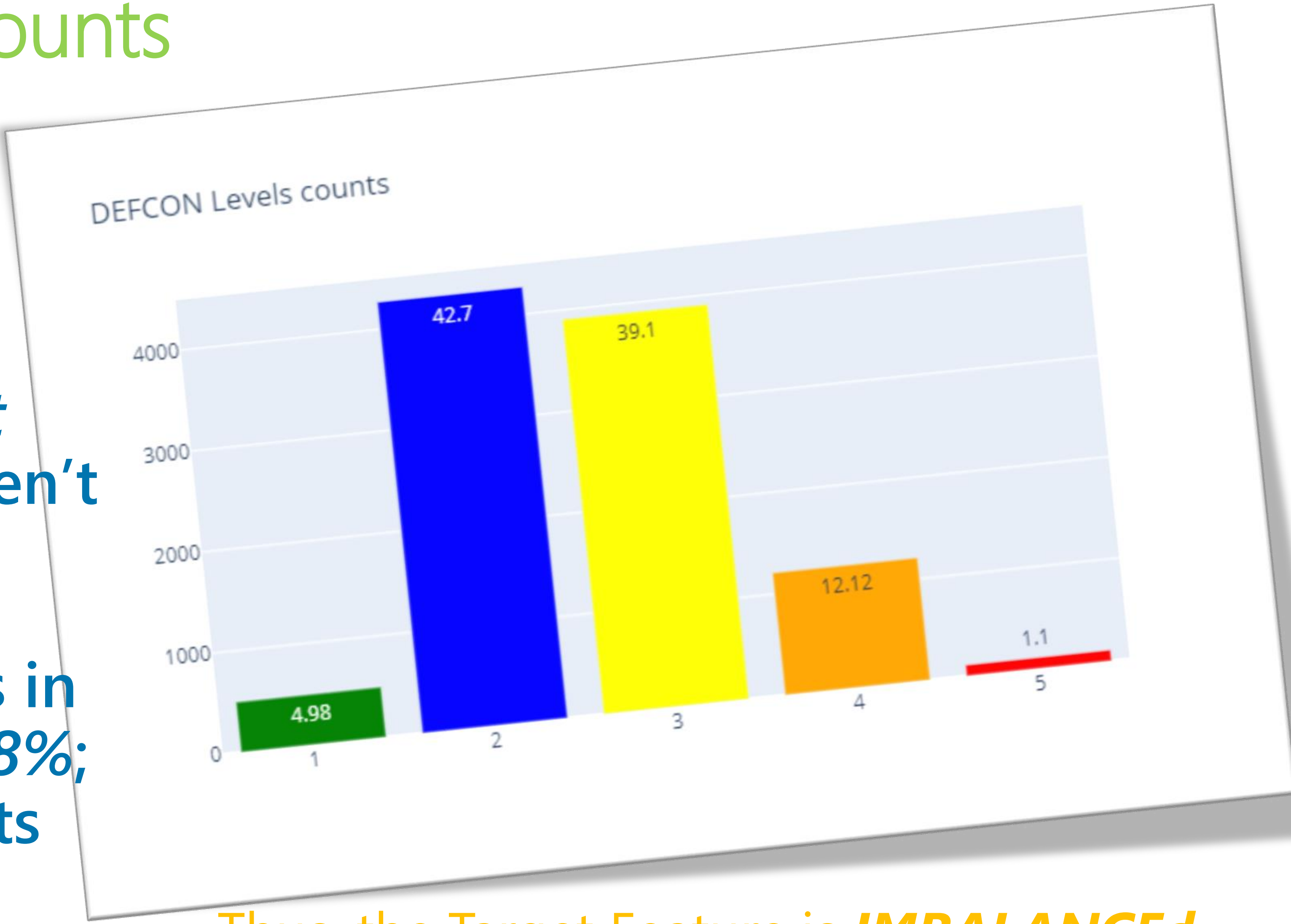
The DEFCON Level is the Target Feature, is Categorical and Discrete in nature.

DEFCON Level Value Counts

Insight

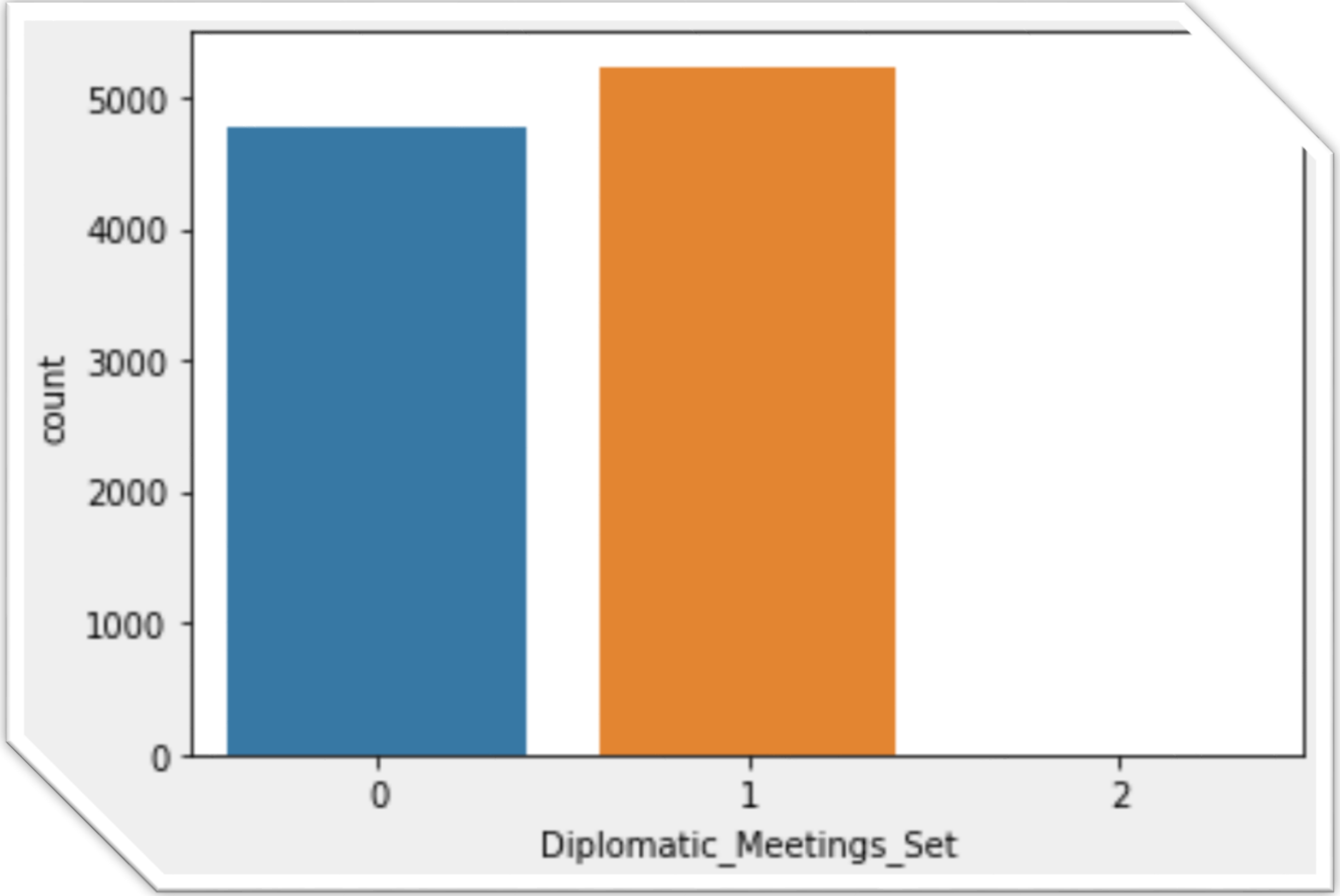
DEFCON Level **is a** ***Target*** **Feature, it's Value Counts aren't spread equally.**

The Levels ***2*** **&** ***3*** **does exists in majorly of Dataset, i.e.** ***~81.8%*****; and Levels** ***5*** **&** ***1*** **does exists minimally in Dataset, i.e.** ***~06.1%***



Thus, the Target Feature is ***IMBALANCEd***.

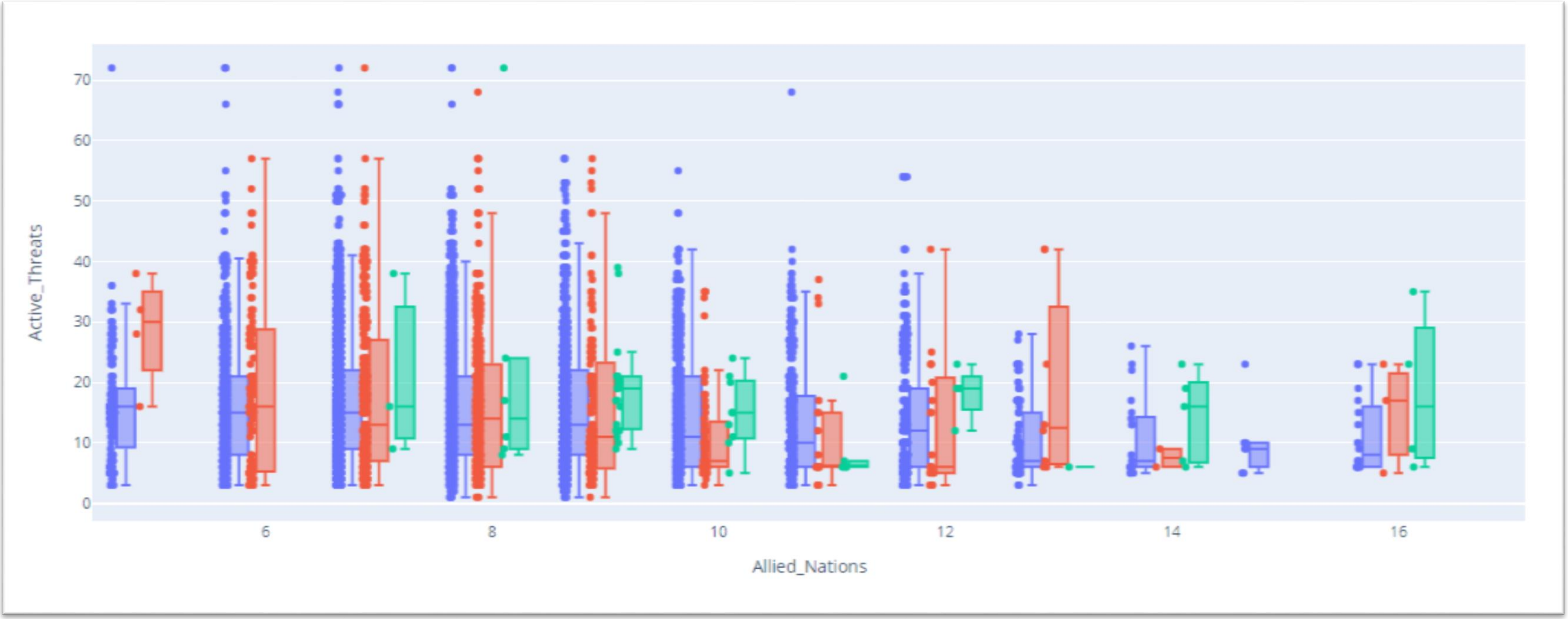
Diplomatic Meetings Value Counts



Insight

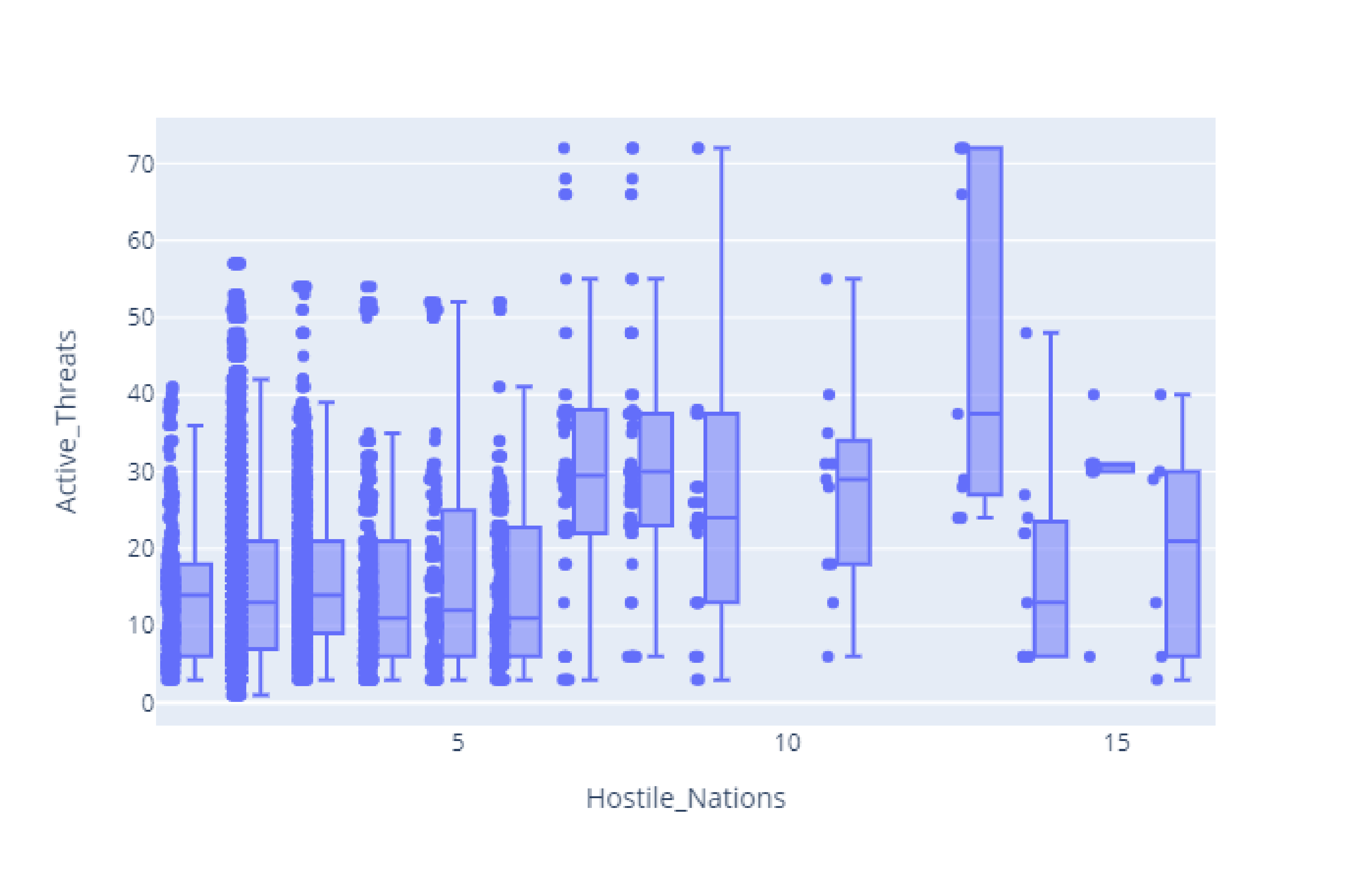
The no. of Diplomatic Meetings set among Nations are either 1 or None.

Allied Nations vs Active Threats



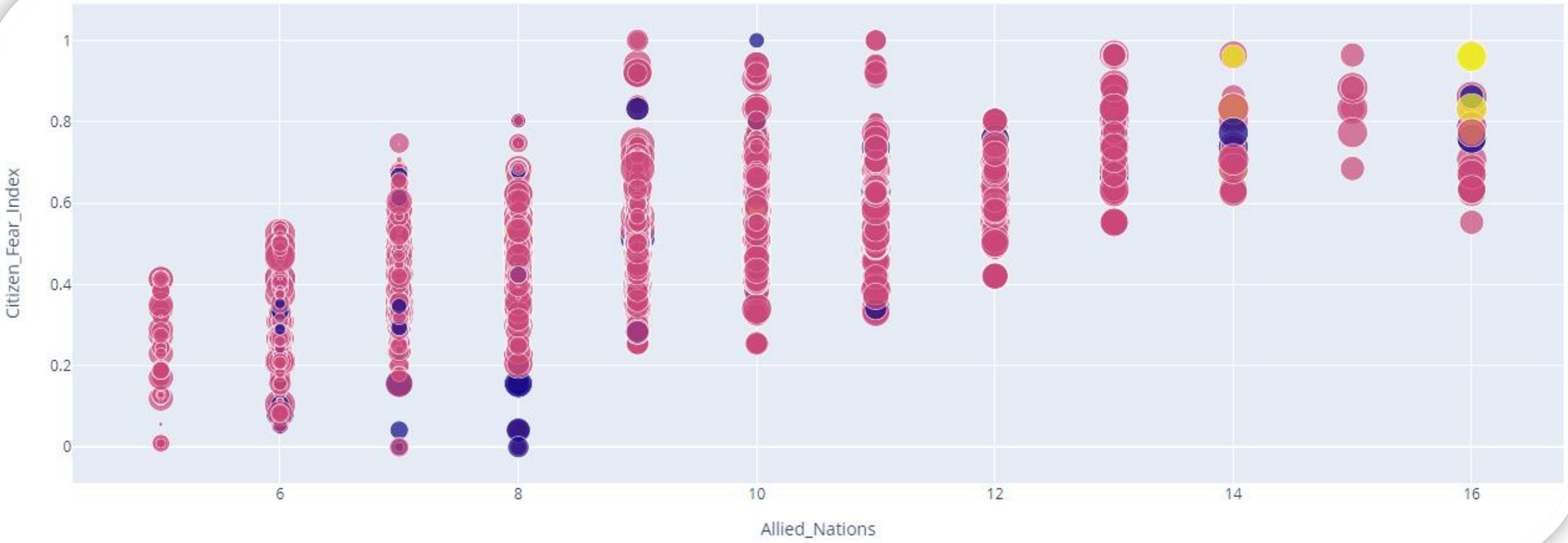
As more Nations Allied, the no. of Active Threats gradually reduces.

Hostile Nations vs Active Threats



Active Threats gradually increases as the no. of Hostile Nations increases.

Allied Nations vs Citizen Fear Index



Citizen Fear Index increases as the no. of Nations are Allied together.

MODEL

Applying Supervised Machine Learning, Deep Learning Modeling techniques on the Dataset to predict the Target Class.

In this Project, I am applying Deep Learning Techniques on the Dataset to predict the Multi Class Feature, i.e. DEFCON Level

Modeling

Using Keras Tensor Flow Deep Learning APIs, created a Model with Optimizers, Early Stopping, Drop Out, Regularization etc. to predict the Target Feature.

“Base Model”

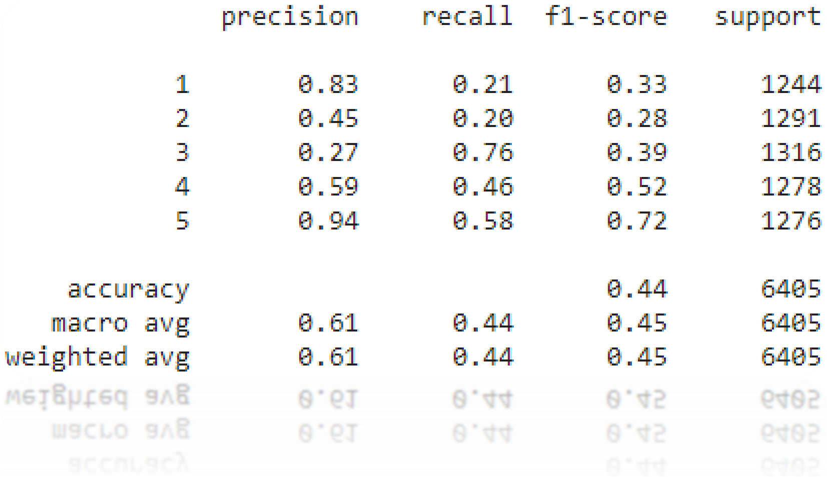
The Base Model did not perform better in predicting the Multi Class Target Feature.

The Accuracy, Precision and F1 Score did not meet threshold.

The Plot of Loss Function in during Training and Validation is not IDLE.

“Reason for Less Accuracy”

The Dataset Target Class has highly Imbalance in nature. Hence Up Sampling has to be performed to increase the no. of majority values.



```
              precision    recall  f1-score   support

           1       0.83      0.21      0.33      1244
           2       0.45      0.20      0.28      1291
           3       0.27      0.76      0.39      1316
           4       0.59      0.46      0.52      1278
           5       0.94      0.58      0.72      1276

    accuracy                           0.44      6405
   macro avg       0.61      0.44      0.45      6405
weighted avg       0.61      0.44      0.45      6405
```

“Model – After handling Imbalance”

Applied SMOTE sampling technique to Up Sample the Target Class's Minority values.

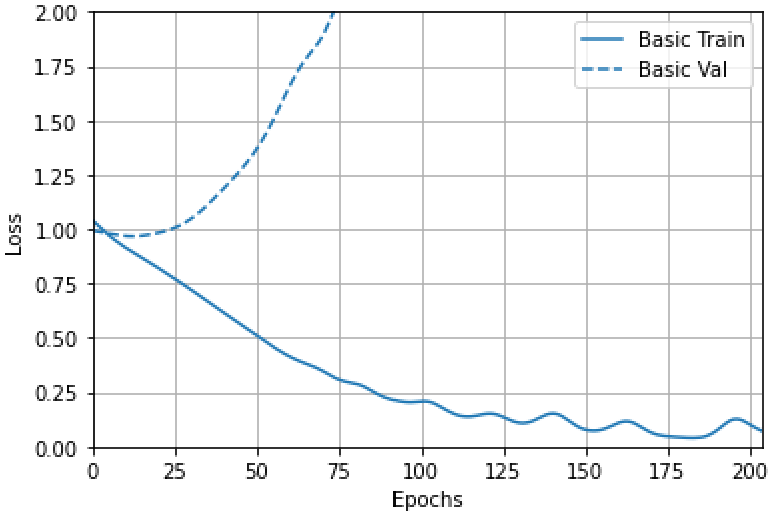
Upon handling the Imbalance, the Model Accuracy is bit improved, i.e. 43%

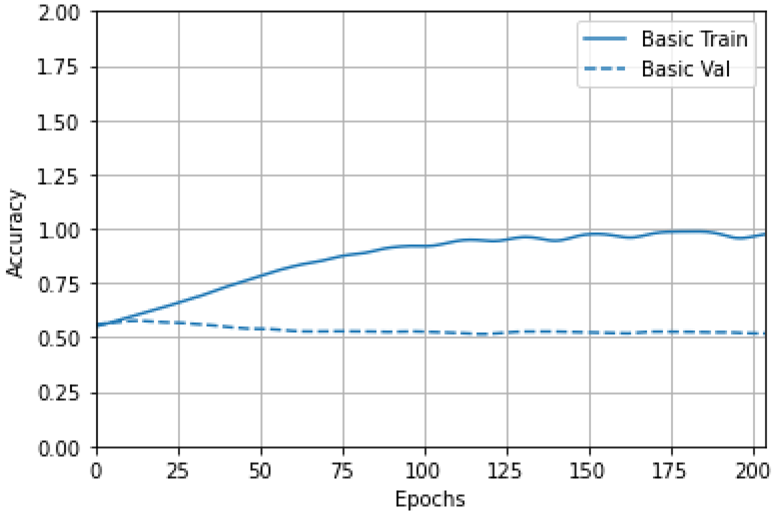
The Plot of Loss Function in during Training and Validation is now IDLE.

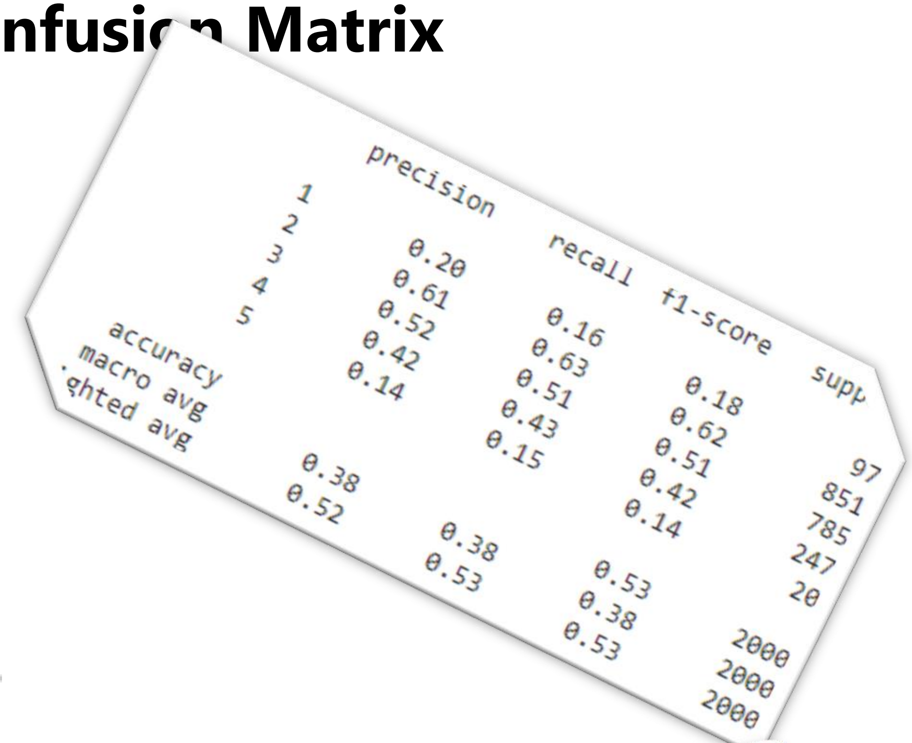
Plot

Plot Loss Function, Accuracy & Confusion Matrix

"Base Model"

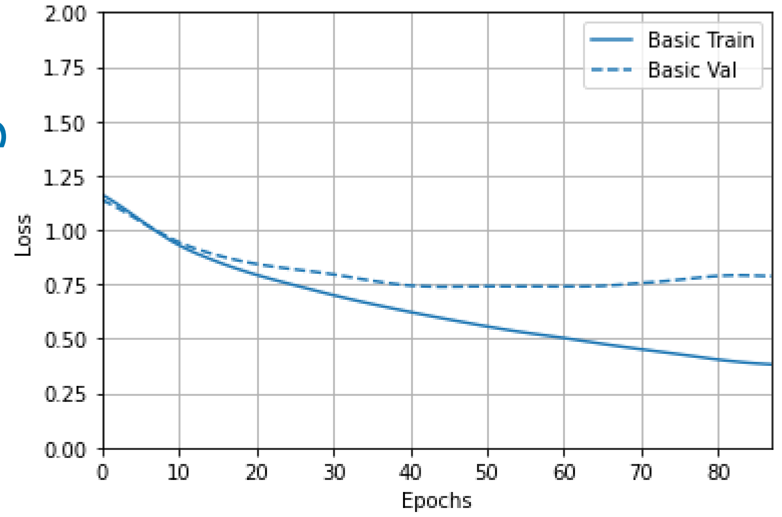


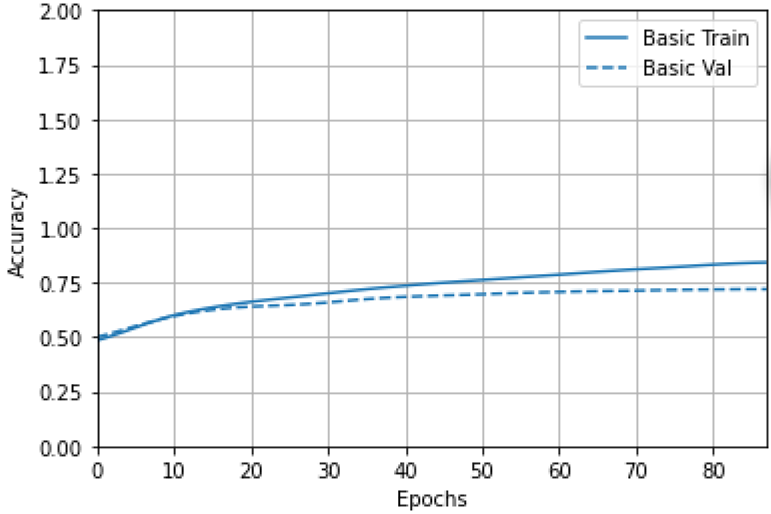


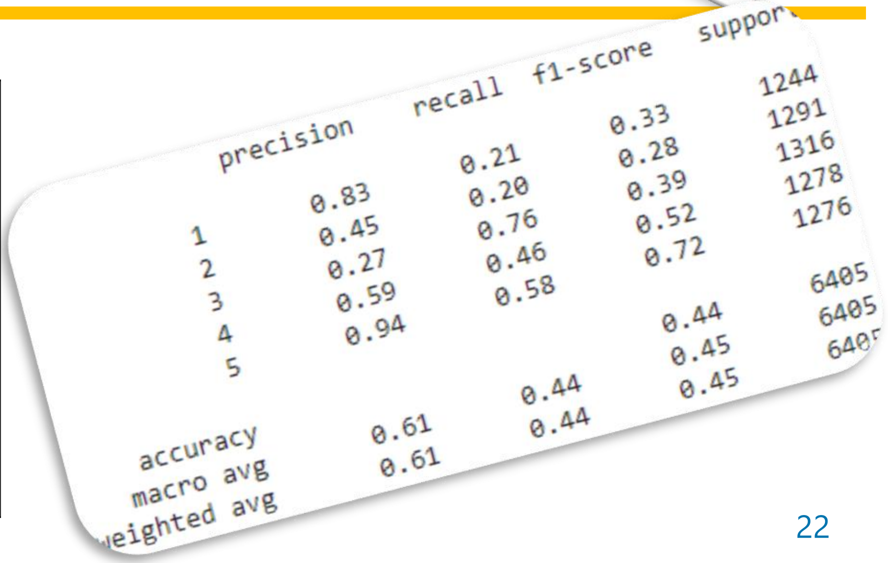


```
              precision    recall  f1-score   supp
           1       0.20      0.16      0.18        97
           2       0.61      0.63      0.62       851
           3       0.52      0.51      0.51       785
           4       0.42      0.43      0.42       247
           5       0.14      0.15      0.14        20

    accuracy                           0.53      2000
   macro avg       0.38      0.38      0.38      2000
 ghted avg       0.52      0.53      0.53      2000
```

"After handling Imbalance"







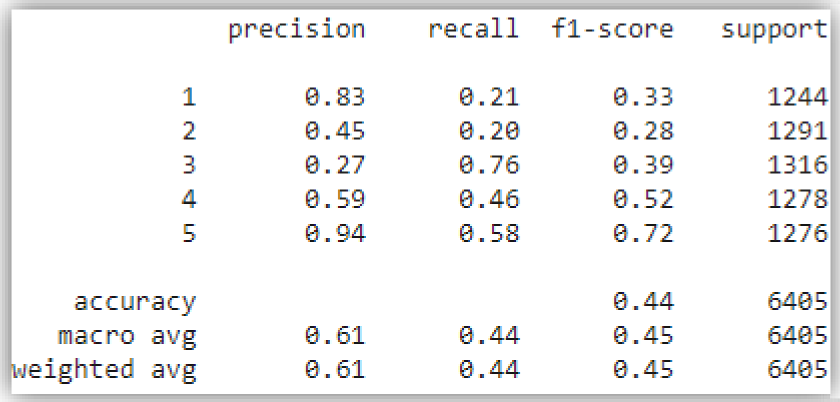
```
              precision    recall  f1-score   suppor
           1       0.83      0.21      0.33      1244
           2       0.45      0.20      0.28      1291
           3       0.27      0.76      0.39      1316
           4       0.59      0.46      0.52      1278
           5       0.94      0.58      0.72      1276

    accuracy                           0.44      6405
   macro avg       0.61      0.44      0.45      6405
weighted avg       0.61      0.44      0.45      640
```

Modeling

Using Keras Tensor Flow Deep Learning APIs, created a Model with Optimizers, Early Stopping, Drop Out, Regularization etc. to predict the Target Feature.

“Label Encoding”

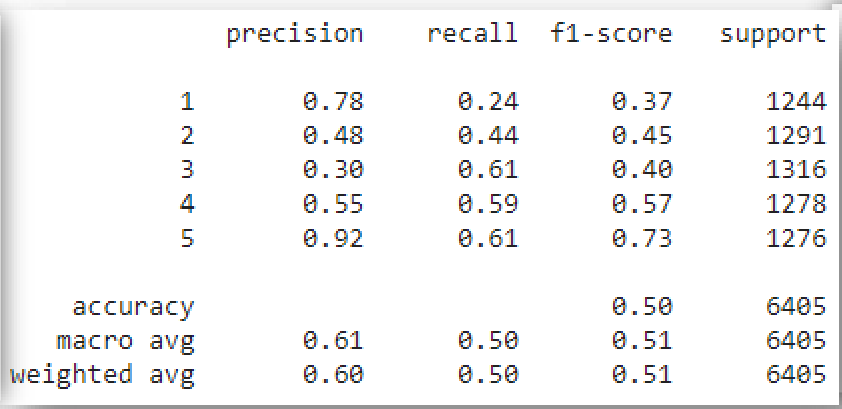
Performed Label Encoding using get_dummies() on multiple features, with which there is observed improvement in the Accuracy



```
              precision    recall  f1-score   support

           1       0.83      0.21      0.33      1244
           2       0.45      0.20      0.28      1291
           3       0.27      0.76      0.39      1316
           4       0.59      0.46      0.52      1278
           5       0.94      0.58      0.72      1276

    accuracy                           0.44      6405
   macro avg       0.61      0.44      0.45      6405
weighted avg       0.61      0.44      0.45      6405
```

“Using Drop Out”

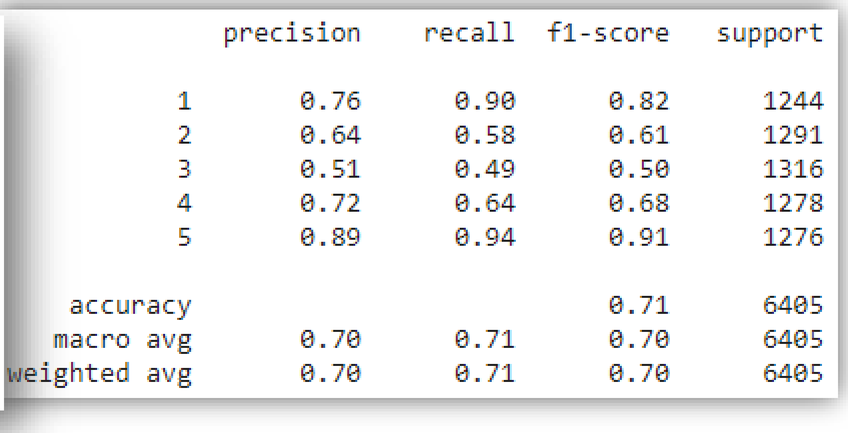
Used Drop Out in Layers to achieve better Consistency and Fitting.



```
              precision    recall  f1-score   support

           1       0.78      0.24      0.37      1244
           2       0.48      0.44      0.45      1291
           3       0.30      0.61      0.40      1316
           4       0.55      0.59      0.57      1278
           5       0.92      0.61      0.73      1276

    accuracy                           0.50      6405
   macro avg       0.61      0.50      0.51      6405
weighted avg       0.60      0.50      0.51      6405
```

“Using Regularization”

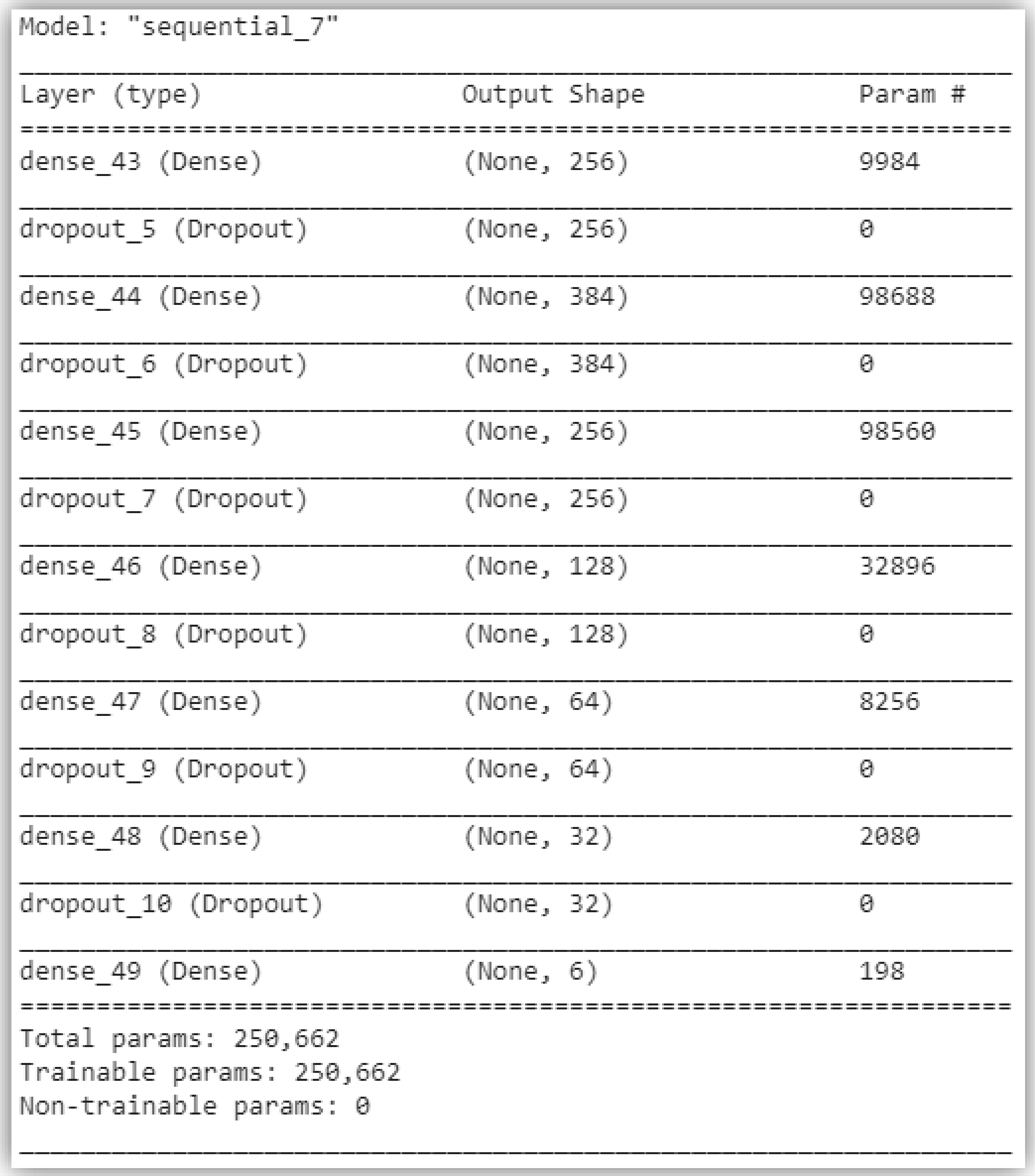
Did Normalize the Weights in between Layers as the Dataset Inputs are already Normalized in nature.



```
              precision    recall  f1-score   support

           1       0.76      0.90      0.82      1244
           2       0.64      0.58      0.61      1291
           3       0.51      0.49      0.50      1316
           4       0.72      0.64      0.68      1278
           5       0.89      0.94      0.91      1276

    accuracy                           0.71      6405
   macro avg       0.70      0.71      0.70      6405
weighted avg       0.70      0.71      0.70      6405
```

Summary

Model Summary



```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_43 (Dense)             (None, 256)               9984      
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0         
_________________________________________________________________
dense_44 (Dense)             (None, 384)               98688     
_________________________________________________________________
dropout_6 (Dropout)          (None, 384)               0         
_________________________________________________________________
dense_45 (Dense)             (None, 256)               98560     
_________________________________________________________________
dropout_7 (Dropout)          (None, 256)               0         
_________________________________________________________________
dense_46 (Dense)             (None, 128)               32896     
_________________________________________________________________
dropout_8 (Dropout)          (None, 128)               0         
_________________________________________________________________
dense_47 (Dense)             (None, 64)                8256      
_________________________________________________________________
dropout_9 (Dropout)          (None, 64)                0         
_________________________________________________________________
dense_48 (Dense)             (None, 32)                2080      
_________________________________________________________________
dropout_10 (Dropout)         (None, 32)                0         
_________________________________________________________________
dense_49 (Dense)             (None, 6)                 198       
=================================================================
Total params: 250,662
Trainable params: 250,662
Non-trainable params: 0
_________________________________________________________________
```

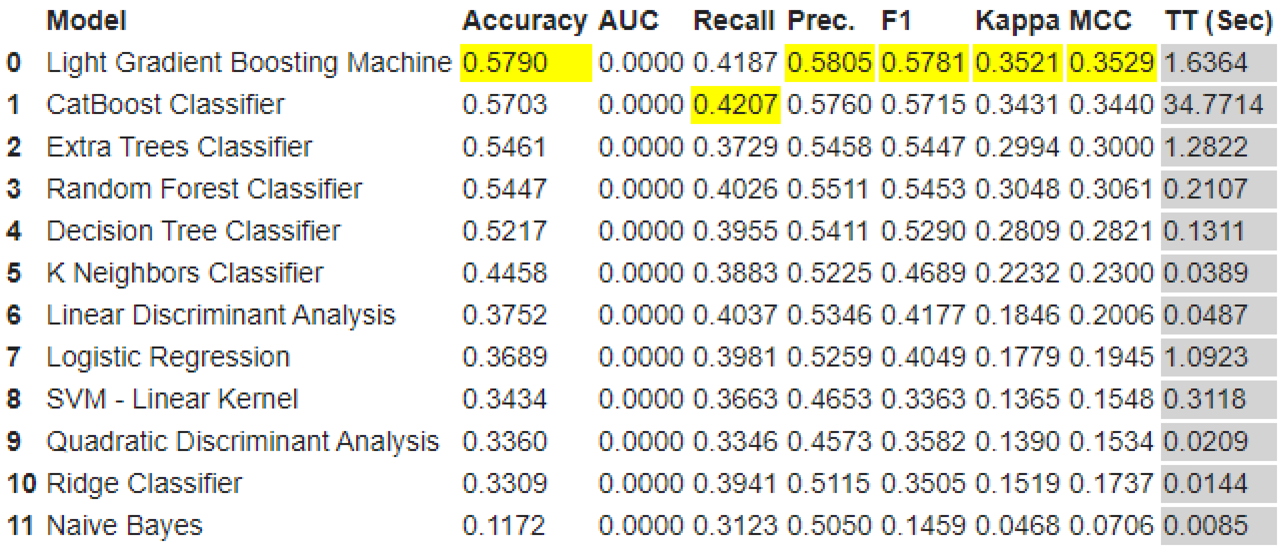
Accuracy	:	71%
Precision	:	70%
Recall	:	71%
F1 Score	:	70%

Applied PyCaret

PyCaret is an open source, low-code machine learning library in Python that allows you to go from preparing your data to deploying your model within minutes in your choice of notebook environment

PyCaret

I have applied PyCaret on the Dataset to check how does the Machine Learning Algorithms are performing on the Dataset, but does not beat Deep Learning Neural Network Model.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Light Gradient Boosting Machine	0.5790	0.0000	0.4187	0.5805	0.5781	0.3521	0.3529	1.6364
1	CatBoost Classifier	0.5703	0.0000	0.4207	0.5760	0.5715	0.3431	0.3440	34.7714
2	Extra Trees Classifier	0.5461	0.0000	0.3729	0.5458	0.5447	0.2994	0.3000	1.2822
3	Random Forest Classifier	0.5447	0.0000	0.4026	0.5511	0.5453	0.3048	0.3061	0.2107
4	Decision Tree Classifier	0.5217	0.0000	0.3955	0.5411	0.5290	0.2809	0.2821	0.1311
5	K Neighbors Classifier	0.4458	0.0000	0.3883	0.5225	0.4689	0.2232	0.2300	0.0389
6	Linear Discriminant Analysis	0.3752	0.0000	0.4037	0.5346	0.4177	0.1846	0.2006	0.0487
7	Logistic Regression	0.3689	0.0000	0.3981	0.5259	0.4049	0.1779	0.1945	1.0923
8	SVM - Linear Kernel	0.3434	0.0000	0.3663	0.4653	0.3363	0.1365	0.1548	0.3118
9	Quadratic Discriminant Analysis	0.3360	0.0000	0.3346	0.4573	0.3582	0.1390	0.1534	0.0209
10	Ridge Classifier	0.3309	0.0000	0.3941	0.5115	0.3505	0.1519	0.1737	0.0144
11	Naive Bayes	0.1172	0.0000	0.3123	0.5050	0.1459	0.0468	0.0706	0.0085

Light Gradient Boosting Machine Model is the tend to be the Best Model with a Accuracy of ~57.9% and F1 Score of ~57.8%

Thank You!

- Yours Ranjith