horizontal line

**AIL302M - MACHINE LEARNING**

TRAN THANH DUONG

SE160185

Viewers' feeling of Youtuber's style Prediction using Neural Network

**13th June 2022**

[COLAB NOTEBOOK](https://colab.research.google.com/drive/1BYEDv2gh8Xh3MIbmQjPefhaxNzJ_seBB?usp=sharing)

# INTRODUCTION

YouTube is the most famous and loved video platform. Nowadays, many people have seen YouTube as a place to work and become an extremely hot profession in the multimedia and entertainment industry. For the AI and Data industry, YouTube is an abundant source of video data, which serves to solve problems to improve social life. In this report, I am going to build a model to predict viewers' feeling of a YouTuber's video style of street food in Vietnam through how they describe them in the video, using Neural Network. The feeling is rated with 5 levels from 1 (worst) to 5 (best).

# NEURAL NETWORK VS. OTHER MODELS

Neural network is a model in the field of deep learning, built on the topology of neurons in brain cells. In recent years, it has shown impressive efficiency and accuracy with problems of Classification and even Regression.

Since this is a problem of Classification, we can build a NN model with hidden layers, the output layer uses the softmax activation function to evaluate the probability of each level.

Linear Regression is not a good idea for classification problems, since its output is a continuous value.

Logistic Regression is used for binary classification, whose output has two values only (0 - 1 or True - False), it is completely capable of creating a Logistic Regression model for this problem by using one-vs-rest method and building 5 separated models for the dataset. It costs time and is susceptible to be biased due to an imbalanced dataset.

# IMPLEMENTATION PROCESS

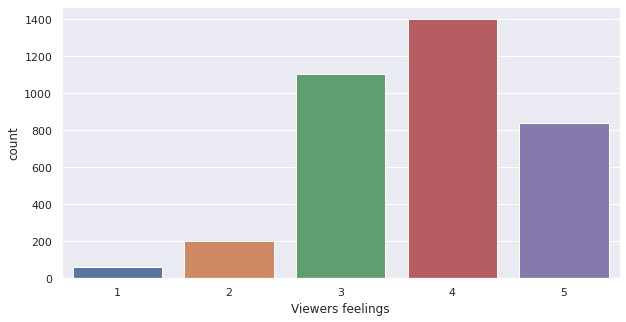
## Data cleaning

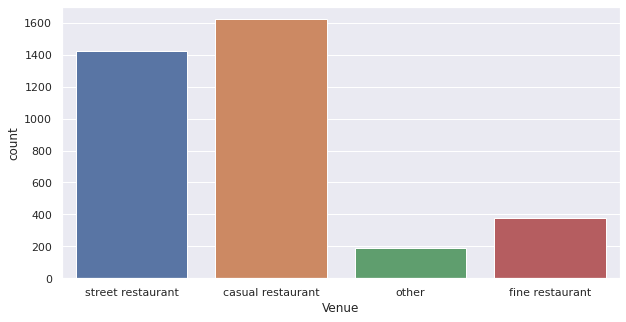
After diving into the dataset, I found out some following errors:

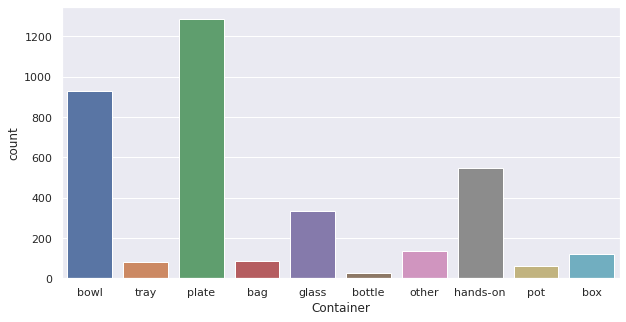
* There are many rows with null values.
* The start time and end time contains some special cases (ex: ***1.30****,* ***20;30****,* ***13 : 20***) and some unknown values (ie: ***End****,* ***end****,* ***x***)
* Some data points have start start time and end time are equal (which means duration is zero!)
* The container and venue have values which are unspecified values (ie: ***clay bot, Bag, no, home, boat restaurant, etc.***)

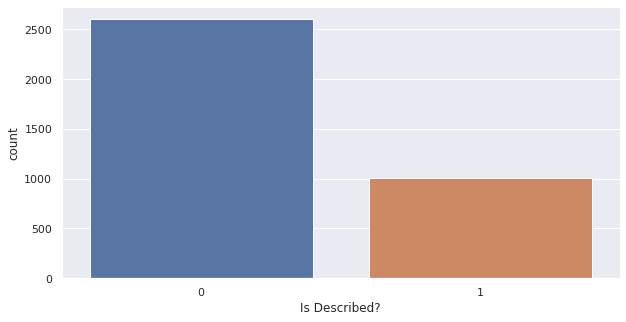
## Data analysis

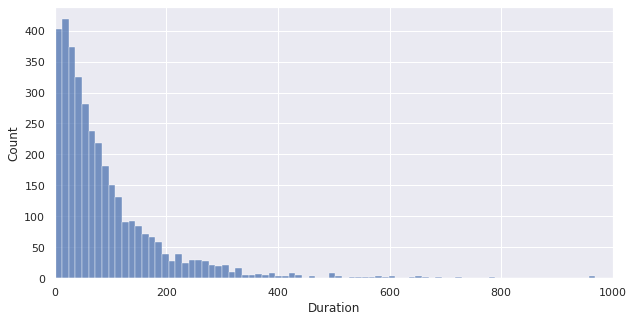
After cleaning data, the dataset will have some of following charts and distributions:

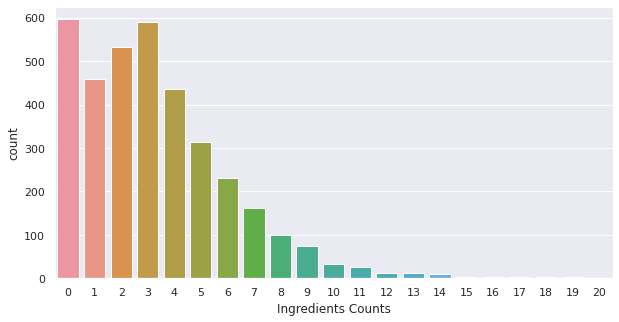












## Data preprocessing

My data processing is about 4 steps:

1. **One-hot encoder**: Converting **Container** and **Venue** feature from categorical value to one-hot array.
2. **Oversampling**: Because the current dataset is imbalanced (the Viewers' feeling shows that the number of samples of level 1 and 2 is smaller than level 4 very much), I use an oversampling method to duplicate un-majority labels.
3. **Train/Test split**: Splitting with ratio of 0.8 for training set and 0.2 for test set.
4. **Standard Scaler**: Standardizing the **Duration** and **Count** feature to bring down all the features to a common scale without distorting the differences in the range of the values. In particular, I didn't standardize whole datasets but the training dataset, then applied that standard scaler to the testing dataset due to data leaking.

## Building Neural Network Model

### Model Architecture

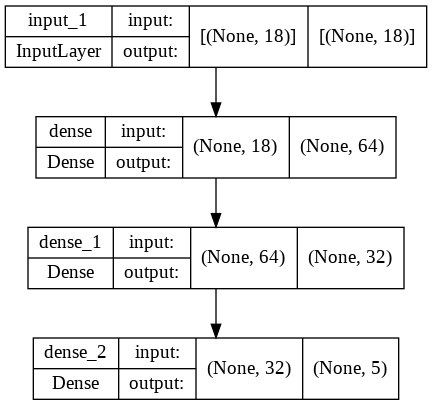
I built a NN model with 1 input layer, 2 hidden layers and 1 output layer with number of neurons respectively:

* 18 neurons (The number of features after preprocessing)
* 64 neurons (using ReLU activation)
* 32 neurons (using ReLU activation)
* 5 neurons (Softmax Activation - Output Layer).

**Optimizers:** Gradient Descent with Momentum (*learning rate = 0.001, momentum = 0.9*)

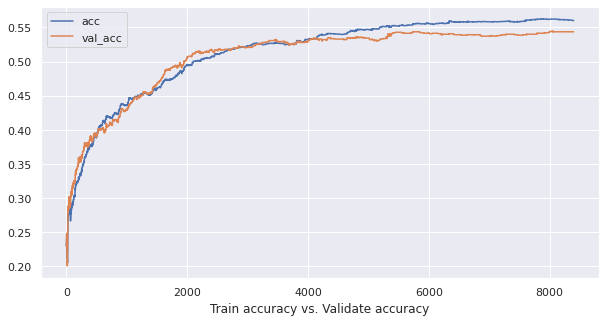
**Loss function:** Categorical Cross-entropy

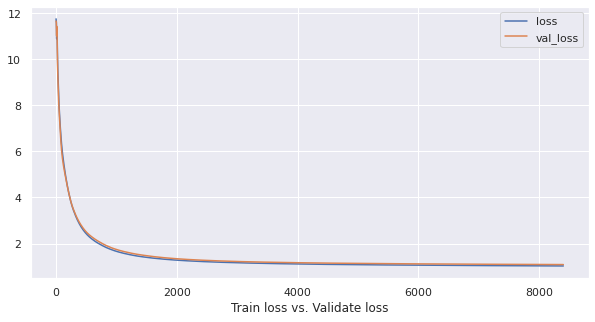
**Number of epochs**: 8,400 epochs



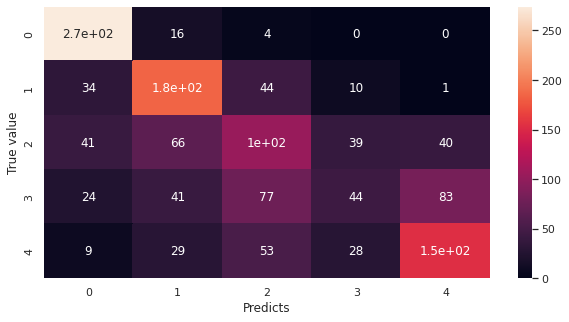
### Model Evaluation

After training on train set and evaluating on test set, these are some plots of results:





**Confusion matrix:** the matrix shows that on testing dataset, the model mis-predicted on label 3 the most.



**Accuracy and Loss on Training and Testing:**

|  | **Accuracy/F1-Score** | **Loss** |
| --- | --- | --- |
| **Training** | 0.5614 | 1.0306 |
| **Testing** | 0.5436 | 1.0902 |

# CONCLUSION

The accuracy of the model is just above the average level (> 0.5), this is an acceptable accuracy when the dataset is too unbalanced and has many errors. Using a fancy and heavy model such as Neural Network cannot make the prediction better, so the data needs to be polished and processed properly from the very first step in order to create a perfect model.