****

**Data Science Programming Project Report**

**Yelp Image Classifier**



**Submitted By,**

* Sudesh V Khillare **U42719953**
* Uday Sai Karnam **U91670152**
* Mohit Dedhe **U97009028**

Table of Contents

1. Problem………………………………………………………………………………… 3
2. Data Description………………………………………………………………………. 4
3. Traditional Machine Learning…………………………………………………………..5
4. Description
5. Results
6. CNN Approach………………………………………………………………………… 8
7. With Pooling
8. Without Pooling and Batch Normalization
9. One vs Rest
10. Overview ………………………………………………………………… ….…… 13
11. References……………………………………………………………………………. 14
12. **Problem Statement**

Yelp provides the image-dataset of food items and restaurant ambiance and want them to classify into the different classes so that it will be trigger as the recommendation system for all the food lovers.

Yelp is giving out dataset to build a modeling technique that without manual intervention tags restaurants with multiple photos using a dataset of images which are submitted by the users. Now, restaurant labels are manually selected by Yelp users when they submit a review. Selecting the labels is optional, leaving some restaurants un- or only partially-categorized.

Nowadays, anywhere we go, we take selfies and food-fies (food photos) and post those things with the story line and it is very obvious that even on the Yelp users are uploading the photos of food item and tagging those with the restaurant and giving their reviews. If we consider only photos can we using food photos and ambiance can we predict something and turn it into the story? This statement is a big challenge in front of Yelp. Because, getting to know the inner sense on the basis of the food and restaurant ambiance photos can really make the change in the future business of the restaurant as well as recommendation system for Yelp.

We have narrowed down the problem statement to classifying the images into the 6 different classes. For, future scope of the project we are thinking to enhancing it and including the NLP for a user review section and use both image and review for the next pipe lining for the recommendation so that user can get the more accurate prediction on the food of their choice.

1. **Data Description:**

This database is created and maintained by the Yelp website. Yelp is a social media site just as Facebook where people find the restaurant on the basis of their preference of food and give reviews with food and restaurant images. Yelp, provide the dataset of the photos which consists of the image’s food items such as Pizza, Tacos, Chicken, Burger and Drinks etc. This dataset consists of the 200000 images from the 6685990 reviews. All these images do have the different categories.

When we got dataset, we first studied it. We understood the parameters in conceptual way, how they will affect our model, which model can be built so that they can categorized the images more precisely. One of the important tasks was manually start giving the labels for the train dataset with the 386 images. For the labelling, we have done the label encoding and insert the data into the X and Y sets. we have to create the test dataset with the 30 images so that we can test our model using prediction function. Hence, we have only 386 data points. We have divided these data points into the test and train and validation set. Where we have 286 training data points, 72 test data points and 30 in test data points. We have tried to use the larger set of data points for all the section. However, it was very hard to compute and process the things because of the computation limitations. Dataset, we have created has a similar distribution as the original dataset do. Here we have higher number count for the burger and pizza whereas for rest of the classes it is less because it is very much similar to what we have seen the original dataset which is parent set for this data. We have done the pre-processing before moving to any of the classification modelling on it. Here we have turn the image into the shape of 256\*256.

1. **Traditional Machine Learning**

3.1 Description

We have started our project with traditional approaches for the classification. Approaches that we have used are mentioned below:

* SGD Classifier
* Polynomial SVM
* KNN classifier
* Random Forest Classifier
* Decision Tree Classifier
* GaussianNB
* Voting Classifier

Description on the each classifier is given below including their results at the end of the description.

**SGD Classifier Tree:**

We have started our approach of classification with the SGD classification model.We are using 80% of dataset for the training and internally it is also used for the validation purpose and 20% for the testing. We have kept the random state of the classifier as 12. As this is categorical variable, we decided to go with SGD classifier algorithm.

**Polynomial SVM**

We have implemented the 2nd approach with the Linear SVM classification model. Here, we have classify the categories on the basis of linear. Here we have kept the random state 42. Because of the linear SVM, accuracy drops here as compare to SGD classifier.

**KNN Classification**

We have used 3rd approach with the KNN (K- Nearest Neighbor) classifier. This approach will classify the data points on the basis of Euclidian distance from their nearest neighbors. Here we have selected the number of neighbors 6 because of the classification

classes are 6.

**Random Forest Classifier**

For the 4th approach, we have selected the Random Forest Classifier, where we have only set the random state =42. Random Forest will create multiple decision trees and gives out the best fit and shows the accuracy that is why accuracy is going to be higher in this classifier.

**Decision Tree Classifier**

5th approach starts with the Decision Tree classification modelling technique. This classifier split the data classification on the basis of 90-degree split, however because of that accuracy goes down. For this classifier we have used the random state as 42.

**Naïve Bayes (Gaussian N.B.)**

6th approach we have started using the Naïve Bayes algorithm for the classification. It will very powerful algorithm to classify the binary data points with the good amount of precision.

**Voting Classifier**

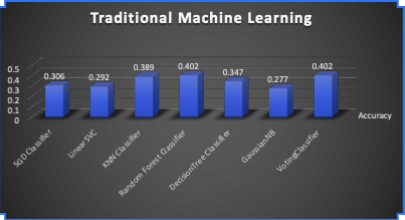
7th approach, we have used the Voting classifier, it is EnsembleVoteClassifier. A meta-classifier for combining similar or conceptually different machine learning classifiers for classification via majority or plurality voting. ... In hard voting, we predict the final class label as the class label that has been predicted most frequently by the classification models. Here we estimated the classifier which are SGD, Polynomial SVM, KNN, Random Forest, Decision Tree.

On the basis of above modelling techniques, we got the highest accuracy with the Random Forest classifier and the voting classifier also dominated because of the Random Forest dataset.

3.2 Results

Please find the table for the accuracy as well as histograms in the figure:

|  |  |  |
| --- | --- | --- |
| SNo | Classification Model Name | Accuracy |
| 1 | SGD Classfier | 30.6% |
| 2 | Polynomial SVM | 29.2% |
| 3 | KNN Classifier | 38.8% |
| 4 | Random Forrest Classifier | 40.2% |
| 5 | Decision Tree Classifier | 34.7% |
| 6 | Naïve Bayes | 27.7% |
| 7 | Voting Classifier | 40.2% |



1. **CNN Approach**

A Convolutional Neural Network, also known as CNN, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. By using a CNN, one can enable sight to computers.

With the CNN approach, we have done the three different modelling technique

1. Without Pooling
2. With Pooling and Batch Normalization
3. One Vs Rest Modelling.

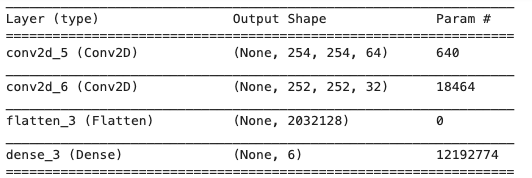
Please find the information on the both methods including results:

1. **Without Pooling**

In this approach we have spilt the dataset using 80%-20% and keep the random state as 42. Here we have use the **Keras** library to import the convolution neural network and other important parameters such as Dense, Dropout, Flatten and Batch-Normalization.

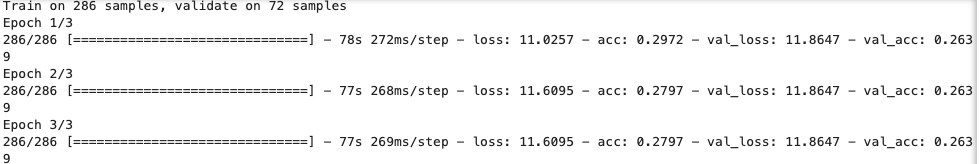
In this model we have created the filter with the size of 64 with the 3\*3 matrix and shape the images with the 256\*256 format. On top of that we have create the same dimension filter as above of 32 filters. However, we are not using continuous pooling and batch normalization. After adding the filters into the CNN, we have used Flatten() function in the classifier, this function changed the whole multi dimension array into the single column linear array. After the flatten step we have classify the output for all the 6 classes using the Dense function inside the Dense function we have used the activation function as ‘SoftMax’ as well as ‘sigmoid’. However, for the sigmoid the accuracy was comparatively low than the ‘SoftMax’ activation function.

Please find the description of the CNN given below :



After adding activation function and filter layers, we have compiled the CNN with the different optimizers such as **sgd**, **adam** , **rmsprop**. However, we have found the highest accuracy and precision with the **adam** optimizer. we also tried different transfer functions on the same CNN architecture such as **binary\_crossentropy, categorical\_crossentropy** and **sparsed\_crossentropy,** on these we found the categorical\_crossentropy given the better performance.

After all the compilation, we fit the model on X\_train and Y\_train data points where we have seen the accuracy around 26.3% which is same while predicting the X\_test and Y\_test. Please find the below screenshot for the same :



**X\_Train and Y\_Train accuracy details**

****

**X\_Test and Y\_Test accuracy details**

1. **With Pooling and Batch Normalization**

In this approach we already used the spilt dataset with the same split (80%-20%) and keep the random state as 42. Here we have use the **Keras** library to import the convolution neural network and other important parameters such as Dense, Dropout, Flatten and Batch-Normalization.

In this model we have continuously do the pooling and batch normalization that is why it reduced the computational time. We have added the multiple layers of filters in this model. Please find the screenshot for the new CNN architecture below:



We have also tried different loss functions, optimizer and activation function. On those **categorical\_crossfunction** as loss function, **adam** as optimizer and **softmax** as activation function outperform the other functions and shows the better accuracy.

Please find the screenshot for the same:

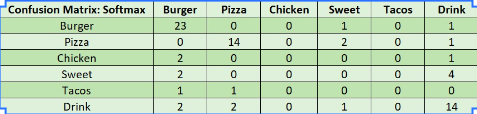


It is very clear that, we have achieved the 95% + accuracy on the train set. However, test set shows the accuracy of 41%. This may be the reason of overfitting the data. We have took the precautionary steps to reduce the chances for over fitting :

1. We have tried the early stopping.
2. We have studied the lecture from the Percyi Liang on reducing the overfitting chances from tweaking the parameters.
3. Data augmentation can solve the overfitting but because of the high computation environment limitation. It can’t reduce over fitting what we have expected.

Important Points and Table

Please find the below screenshot for the confusion matrix:



1. **One Vs Rest**

In the above classification modelling we have done one vs one classification that is the reason confusion matrix has 6X6 table. Here, in this model, we have done the one vs rest classes.

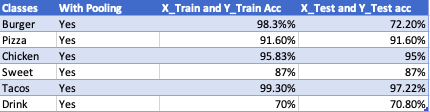
We have started with the Burger class and marked it 0 and kept the rest 5 classes as the binary 1. After loading, resizing it split the data into test and train.

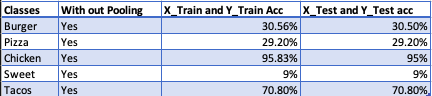
Here, also we have create two neural network architecture

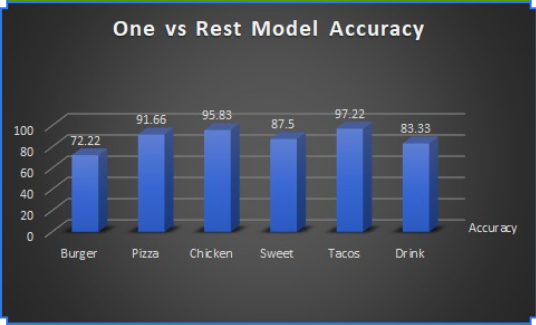
1. Without Pooling
2. With Pooling

Same as above Here also, we have used the different function such as loss functions, optimizer and activation function with the best performance values to get the higher accuracy.

Please find the accuracy table for one class vs rest classes.





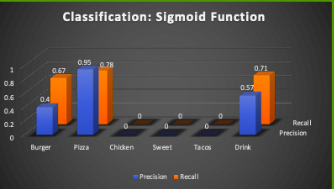


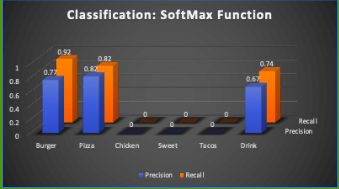
On the basis of above graph it is very much clear that the model one vs rest can accurately predict the photos into the classes but Here, we have seen that pizza is the best possible class which is classified properly. Burger’s classifier, total percentage of the accuracy is low because photos which are included burger do included people and other object that is why it is very hard to define the classes.

1. **Overview**

* Firstly, We have started started the classification using the traditional Machine Learning algorithms where we have seen the algorithms such as KNN, Random Forest and Voting classifier. On those Random Forest was the best to predict the classes comparatively.

* Secondly, we went ahead and create the CNN algorithms with pooling and without pooling. On those basis we have tried acceleration function as sigmoid vs softmax where softmax out perform the sigmoid because sigmoid only predict the binary values between 0 and 1 whereas softmax can predict the multi class classification properly. Please find the attached screen shots for the reference :





* On the basis of the CNN model we have implemented the one vs rest modelling technique which actually classify the data points into binary classes and see whether the one class is classifies properly.
* Our whole main is to try the Machine Learning algorithms and CNN on the Yelp Image classification so that it will trigger the photos into the most probable classes and after pipelining the classification we can turn those classification into recommendation.

1. **References**

* <https://keras.io/>
* <https://www.youtube.com/watch?v=1DBhzQgB5xw>
* <https://www.yelp.com/dataset/challenge>
* <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05>