A Project Report On

Prediction of Cuisine using the recipe ingredients

(Big Data Application Development)

Submitted By

Pooja Anandani (16012121003)
Aawez Mansuri (16012121013)
Nikunj Thakkar (16012121016)

OCTOBER 2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
U.V. PATEL COLLEGE OF ENGINEERING
GANPAT UNIVERSITY

CERTIFICATE

TO WHOM SO EVER IT MAY CONCERN

This is to certify that students of B.Tech. Semester (VII) (COMPUTER SCIENCE & ENGINEERING) has completed his one full semester on project work titled "**Cuisine Prediction"** satisfactorily in the subject **"Big Data Application Development "** of **Computer Science & Engineering**, Ganpat University in the year 2019.

Pooja Anandani(16012121003)
Aawez Mansuri (16012121013)
Nikunj Thakkar (16012121016)

Internal Guide
Prof. Aniket Patel

Date:

HOD
CSE Department
Dharmesh Darji

Date:

ACKNOWLEDGEMENT

We have been constantly putting our efforts to make this project possible. However, it would not have been possible without the kind support and help of many individuals.
We would like to extend our sincere thanks to all of them.

We are highly indebted to **Prof. Aniket Patel** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude members of the **Institute of Computer Technology (ICT)** for their kind co-operation and encouragement which help us in the completion of this project.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

ABSTRACT

Over the years, people have tried to explore new ingredients and incorporate them into recipes or produce new recipes all together. One of the obvious relations that we would like to explore is the relationship between ingredients and cuisines. We use the Yummly data-set to study the problem of predicting the cuisine of a recipe based on its ingredients. On testing several classifiers we observed that SVM works best for this prediction task.

TABLE OF CONTENTS

INTRODUCTION

1. Project Definition

Our main goal of this project is to create a highly accurate machine learning model that should be able to predict cuisine based on the given recipe ingredients. It takes ingredients as input and predicts the category of cuisine as output.

PROJECT SCOPE

The main purpose of this model is to provide an auto-categorizing model for the restaurants and online food ordering applications such as Swiggy, Zomato and Uber Eats. Furthermore, it can also be used for creating a food recommendation system by finding similarities between the ingredients of the recipes.

SOFTWARE AND HARDWARE REQUIREMENTS

2. **Software Requirements**

   Python 3.6 or above

   Tableau

   Libraries Used

   - LabelEncoder
   - SVC
   - Pandas
   - Numpy
   - Json
   - One vs Rest Classifier
   - TfidVectorizer

3. **Hardware Requirements**

   A Machine with at least 8 GB & Quad-Core is recommended for faster computation.

PROJECT PLAN

Status	Finished
July 1st 3 weeks	Research regarding the subject.
July 4th week	Finding an appropriate algorithm.
August 1st week	Solving the problems faced with SVM.
August 2nd week	Facing Issues with Categorical Data in SVM.
September 2nd week	TFID Application in SVM.
September 3rd week	Training the model.
September 4th week	Testing the model.
October 1st and 2nd week	Finishing.

Table 1: PROJECT PLAN.

IMPLEMENTATION DETAILS

4. **Data Cleaning.**
   As we took the dataset from the kaggle, the data was already preprocessed and cleaned. But following the are basic preprocessing step to ensure that our data is ready for model training.

   - Getting rid of extra spaces.
   - Converting all characters into lowercase.
   - Removing stop-words. i.e. Removing common words such as A, The, An, etc.
   - not helpful for prediction.
   - Removing the duplicate values.
   - Removing words like salt, water, and spices that are used commonly in each cuisine.

5. **Reading the Dataset.**
   Our kaggle dataset is in JSON format, thus we have used python library for it.

```
import json
def read_dataset(path):
        return json.load(open(path))
train = read_dataset('train.json')
test = read_dataset('test.json')
```

6. **Convert the Data into Simple Text from JSON.**
   We have to convert our JSON data into simple data in a python list. The reason for it is that we have to perform TF-IDF Vectorization for using multi-value categorical data in SVM.

```
def generate_text(data):
        text_data = [" ".join(doc['ingredients']).lower() for doc in data]
        return text_data

train_text = generate_text(train)
test_text = generate_text(test)
target = [doc['cuisine'] for doc in train]
```

7. TF-IDF on text data.

The term TF-IDF is used for multi-value data classification.
TF: Term frequency, which measures how frequently a term occurs in a document. Since every record is different in length, It is possible the single term may occur in the lengthy records a number of times than shorter ones. This is the way of normalization.

TF(t) – Number of terms appears in a doc/total number of terms
IDF- Measures the importance of the term in the Document.
IDF(t) – log_e(total number of documents / number of documents in term t in it.)

We have used sklearn for Tfidf Vectorization.

```
from sklearn.feature_extraction.text import TfidfVectorizer
print ("TF-IDF on text data ... ")
tfidf = TfidfVectorizer(binary=True)
def tfidf_features(txt, flag):
    if flag == "train":
            x = tfidf.fit_transform(txt)
    else:
              x = tfidf.transform(txt)
    x = x.astype('float16')
    return x
X = tfidf_features(train_text, flag="train")
X_test = tfidf_features(test_text, flag="test")
```

8. Label Encoding

Label encoding will convert our training data class which is categorical into a numerical value. A Unique value will be generated for each class label.

We have used sklearn for label encoding.

```
from sklearn.preprocessing import LabelEncoder
lb = LabelEncoder()
y = lb.fit_transform(target)
```

9. Model Training

We will now train our SVM model using a sklearn support vector classifier.

```
from sklearn.svm import SVC
from sklearn.multiclass import OneVsRestClassifier
classifier = SVC(C=100, # penalty parameter
                 kernel='rbf', # kernel type, rbf working fine here
                 degree=3, # default value
                 gamma=1, # kernel coefficient
                 coef0=1, # change to 1 from default value of 0.0
                 shrinking=True, # using shrinking heuristics
                 tol=0.001, # stopping criterion tolerance
                 probability=False, # no need to enable probability estimates
                 cache_size=200, # 200 MB cache size
                 class_weight=None, # all classes are treated equally
                 verbose=False, # print the logs
                 max_iter=-1, # no limit, let it run
                 decision_function_shape=None, # will use one vs rest explicitly
                 random_state=None)
model = OneVsRestClassifier(classifier, n_jobs=4)
model.fit(X, y)
```

10. Predicting

As we have finished training the data, now we will use the model for prediction on our testing dataset.

```
y_test = model.predict(X_test)
y_pred = lb.inverse_transform(y_test)
```

TESTING RESULTS

As we have finished execution, we have successfully predicted the cuisine for the testing data. The result is as below.

```
id,cuisine
18009,irish
28583,southern_us
41580,italian
29752,cajun_creole
35687,italian
38527,southern_us
19666,greek
41217,chinese
28753,mexican
22659,british
21749,italian
44967,greek
42969,indian
44883,italian
20827,british
23196,french
35387,mexican
33780,southern_us
19001,mexican
16526,southern_us
42455,japanese
47453,indian
42478,irish
11885,vietnamese
16585,italian
29639,southern_us
26245,thai
38516,korean
47520,italian
26212,southern_us
23696,mexican
14926,thai
13292,indian
27346,japanese
1384,chinese
15959,mexican
```

Figure 1: Output

INSTALLATION MANUAL

For the installation and execution of this model, all you require is Python 3.6 or later, and the prerequisite libraries.

You can install the python libraries using pip command.
The following are the required libraries.

- Pandas
- Numpy
- Sklearn
- JSON

CONCLUSION AND FUTURE SCOPE

In conclusion, we were able to successfully apply the model using Support Vector Machine Algorithm. We were able to classify the cuisine on the testing data. For Future scope, this model can be integrated into online food ordering or restaurant applications.