**COMP4651 – Cloud Computing and Big Data Systems**

**Project Report**

**Group Members:**

Zeng Yueran 20591873 yueransutd

Naik Hiong Chiang 20591885 frankienhc

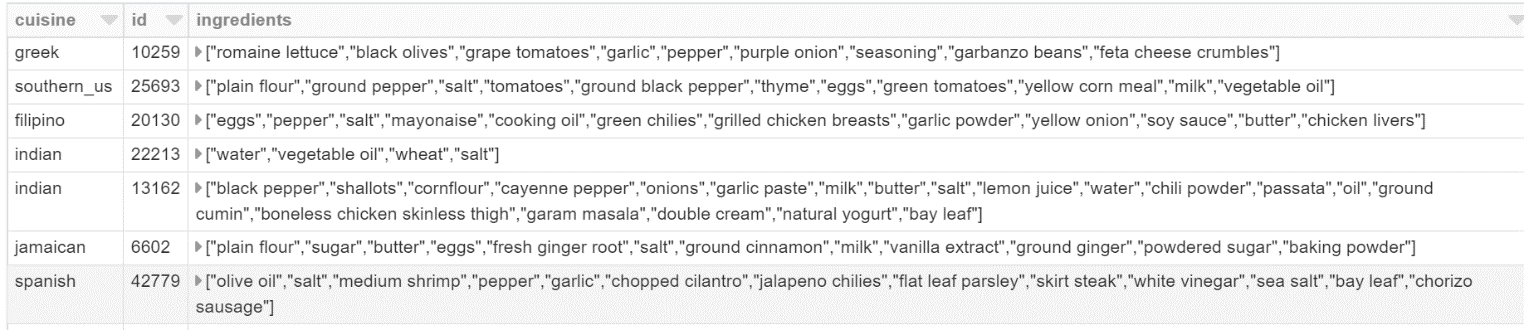
Tan Ting Yu 20591861 tytanab

1 | Introduction

2 | Data Exploration & Processing

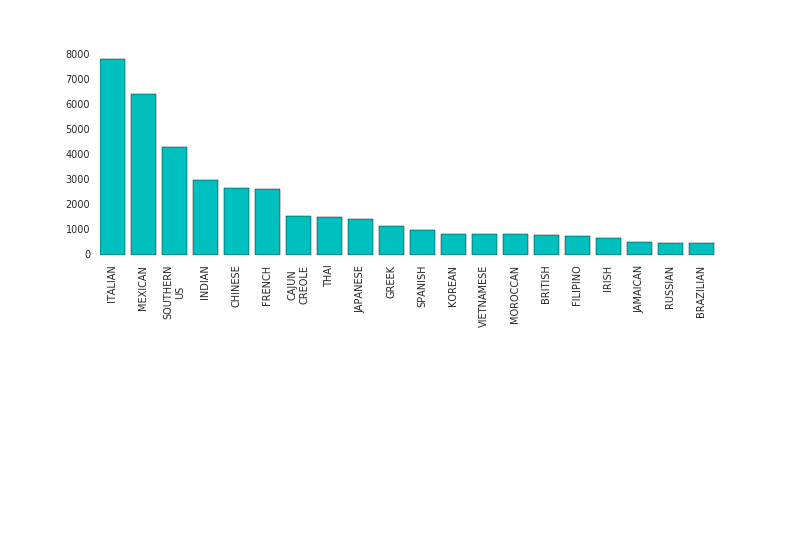
Before formulating our evaluation strategy, it is necessary to examine the given dataset, in order understand its limitations and ensure that is sufficiently well-balanced. The dataset is publicly available on Kaggle (<https://www.kaggle.com/c/whats-cooking/data>). There is a total of 39, 774 recipes across 20 cuisines, containing 6714 unique ingredients, with each recipe stored as JSON objects in the form:

**{"cuisine": "indian", “id”: 24717, "ingredients": ["tumeric", "vegetable stock", … ]}**

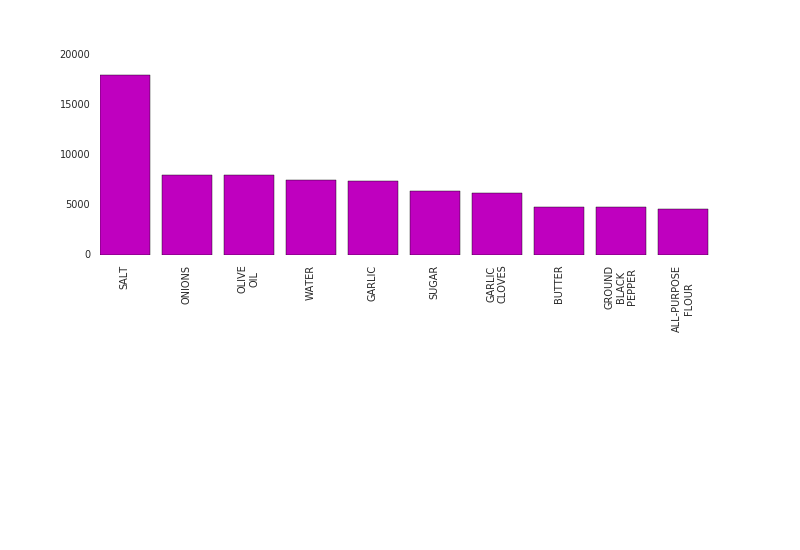


Frequency Analysis

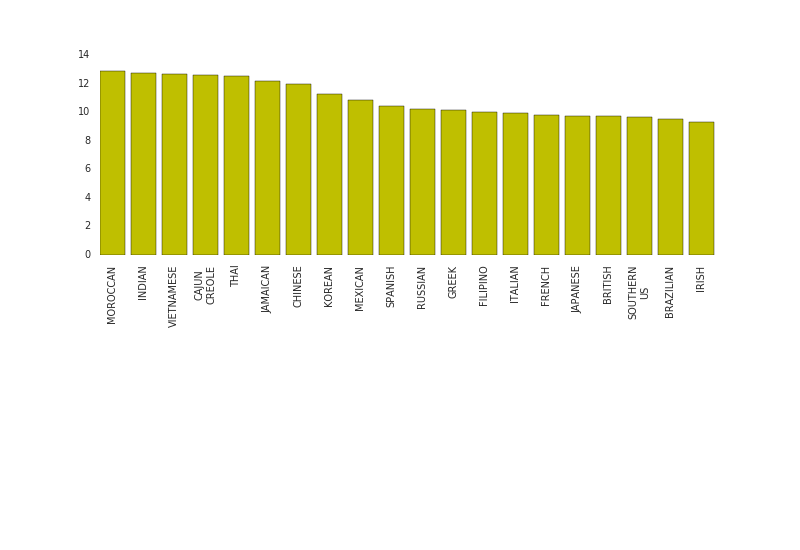
Applying MapReduce transformations with Spark, helpful visualizations that describe the characteristics and distribution of data can be constructed, as shown below:



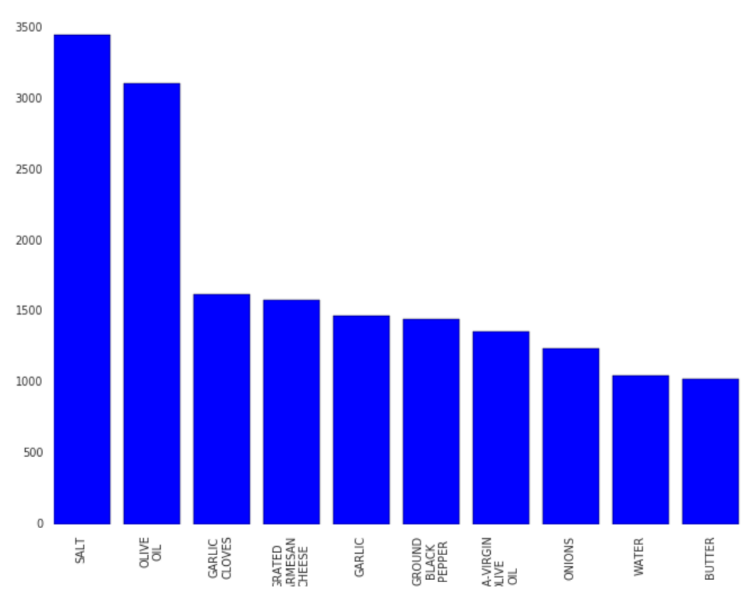
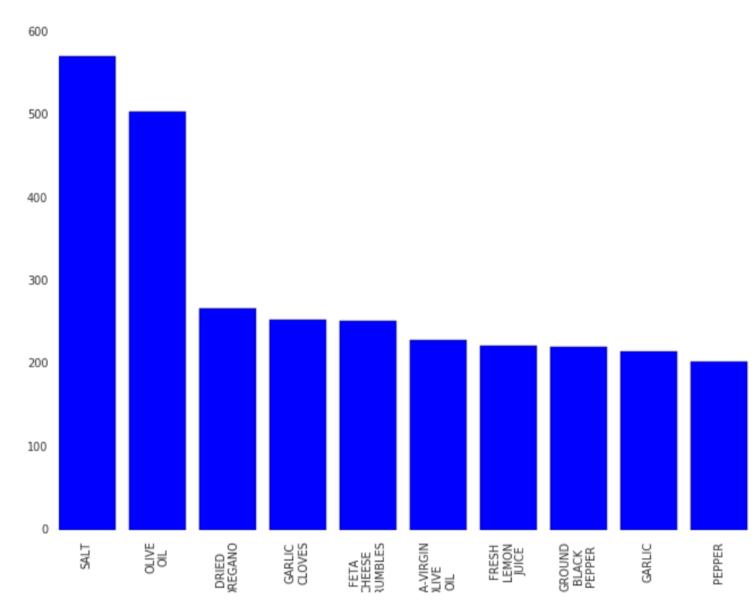
*No. of Recipes By Cusine*

**

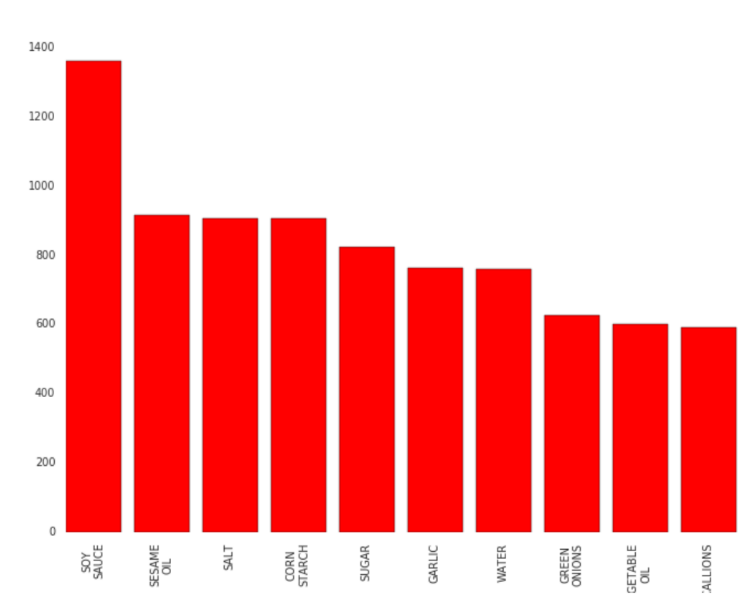
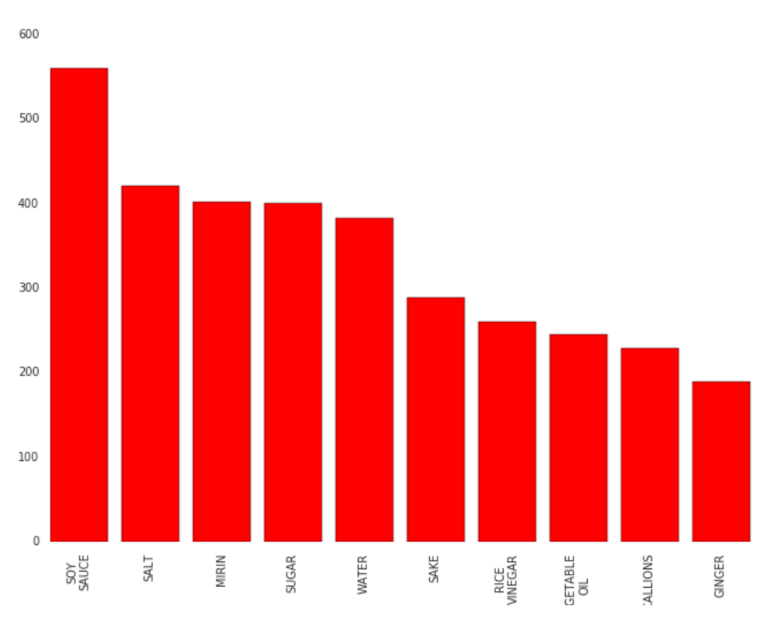
*No. of Occurrences Per Ingredient Across All Recipes (Top 10)*



*Average No. of Ingredients Per Recipe, By Cuisine*

**

*No. of Ingredients Occurrence for Italian vs. Greek (Top 10)*



*No. of Ingredients Occurrence for Japanese vs. Chinese (Top 10)*

Key observations include the following:

1. Italian and Mexican recipes are overly-represented.
2. Single-occurrence ingredients are too specific (e.g. ‘Kraft Original Barbecue Sauce’, ‘Boned Skinned Duck Breast Halves’, ‘Pineapple Chunks In Natural Juice’, etc.) and will not be particularly helpful. These ingredients have very low occurrence, and half of the ingredients occurs less than 5 times.
3. The average number of ingredients used per recipe is reasonably uniform across cuisines (≈ 11).

Using Spark’s Word2Vec machine learning API, we search for and sieve out ingredients that are essentially or close to identical. To demonstrate, we compare 5 ingredients that appear only once with all other ingredients, so that we can extract those that are most alike, and if possible, to rename them accordingly, hence reducing the number of ‘features’.

|  |  |  |
| --- | --- | --- |
| **Ingredient** | **Most Alike** | **Cosine Similarity \*** |
| TOPSIDE STEAK | STEAK | 0.824 |
| CALAMARI STEAK | 0.749 |
| RUMP STEAK | 0.705 |
| ICING MIX | SOUR MIX | 0.857 |
| STUFFING MIX | 0.827 |
| CAKE MIX | 0.817 |
| KRAFT ORIGINAL BARBECUE SAUCE | BARBECUE SAUCE | 0.928 |
| HOISIN SAUCE | 0.915 |
| ASIAN BARBECUE SAUCE | 0.914 |
| SKINLESS CHICKEN BREAST FILLETS | SKINLESS CHICKEN FILLETS | 0.992 |
| CHICKEN BREAST FILLETS | 0.989 |
| BONELESS CHICKEN BREAST | 0.987 |
| BONED SKINNED DUCK BREAST HALVES | SKINNED BONED DUCK BREAST HALVES | 1.0 |
| BONED DUCK BREAST HALVES | 0.987 |
| DUCK BREAST HALVES | 0.975 |
| \* Values range from -1 to 1, where -1 is the most dissimilar (180°), and 1 implying that they are exactly the same (0°). | | |

3 | Prediction Metric

To start, the data is separated into ‘training’ and validation sets using Spark’s randomSplit method, with a distribution of 99% and 1% respectively. This is practical as we are using a simple linear regression model – we will not be updating the weights once they have been calculated, thus we want to maximize the number of ‘training’ data points, which in this situation includes a comprehensive ingredients database.

4 | Results & Discussion

5 | Conclusion