

## Capstone Project Weekly Progress Report

<b>Project Title</b>	MEALBUDDY
<b>Group Name</b>	GROUP_G
<b>Student names/Student IDs</b>	ELVIN IYPE MATHEW C0769974 ELDA VARGHESE C0769741 TOM JOSEPH C0760915 JEENA HELEN FRANCIS C0764493 CHINJU BABY C0769912
<b>Reporting Week</b>	26 JULY 2020 - 01 AUGUST 2020
<b>Faculty Supervisor</b>	WILLIAM POURMAJIDI

### 1. Tasks Outlined in Previous Weekly Progress Report

- Final Report work start
- Machine Learning based recommendation functionality
- Studied about AWS service - Amazon SageMaker
- Updated Architecture diagram
- Presentation work start
- Testing scenarios

### 2. Progress Made in Reporting Week

- Tried restaurants recommendation system with different ML algorithms
- First trained and tested Random forest Algorithm in Jupyter Notebook
- In Random Forest, since we are getting the model score as 1, i.e 100%.
- But in real case situations it's not possible to get 100% result.

```
In [41]: model.score(X_test, y_test)
```

```
Out[41]: 1.0
```

- so we tried with xgboost algorithm
- Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning (ML) models quickly. SageMaker removes the heavy lifting from each step of the machine learning process to make it easier to develop high quality models.

#### What AWS SageMaker does?

- Select and prepare training data
- Choose and optimize ML model
- Setup and manage environment for training
- Train and tune your model (trial & error)
- Deploy your ML model to production
- Hyperparameter optimization is possible with AWS SageMaker- For a given set of input features (the hyperparameters), hyperparameter tuning optimizes a model for the metric that you choose.
- Scale and manage production environment

The screenshot shows the AWS SageMaker console and the JupyterLab interface. The console displays a list of notebook instances, and the JupyterLab interface shows the file explorer and the JupyterLab logo.

**AWS SageMaker Console - Notebook instances**

Name	Instance	Creation time	Status	Actions
restaurantRecommendationML	m1t2.medium	Jul 30, 2020 17:35 UTC	InService	Open Jupyter   Open JupyterLab

**JupyterLab Interface**

book.us-east-1.sagemaker.aws/tree

jupyter

Files Running Clusters SageMaker Examples Conda

Select items to perform actions on them.

Upload New

Name	Last Modified	File size
ML_restaurants.ipynb	3 minutes ago	10.2 kB
Untitled.ipynb	11 minutes ago	72 B

```

resturantrecommendationml.notebook.us-east-1.sagemaker.aws/notebooks/ML_restaurants.ipynb
jupyter ML_restaurants Last Checkpoint: 2 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Help Not Trusted No Kernel

In [69]: import pandas as pd
import requests
import json
from urllib.parse import urljoin

In [70]: API_KEY = "vj2DjKvI1oBpqInyqQ0l4hqr37Y3E9WMyu1TP9cne1Fuq4tOW09HwXBn2KIPfj00ms5xVjF6BRVSmQmLwdBdLwc0QHcyxfNrR790kjF1G2m_glhyy7I"
YELP_SEARCH_API = "https://api.yelp.com/v3/businesses/search"
DEFAULT_TERM = "dinner"
DEFAULT_LOCATION = "Toronto"

In [102]: def searchRestaurants(cuisine,offset):
    payload = {
        'term': cuisine + " restaurants",
        'location': DEFAULT_LOCATION,
        'sort_by': 'rating',
        'offset': offset,
        'limit': 50,
    }
    auth_key = ' '.join(['Bearer', API_KEY])
    headers = {"Authorization": auth_key}
    api_call = requests.get(YELP_SEARCH_API, params=payload, headers=headers)

    result = api_call.json()
    biz = result['businesses']
    result_str = []
    if not biz:
        return result_str
    return biz

In [103]: #print(searchRestaurants('indian'))

In [109]: import os
import time

cuisines = ['italian', 'chinese', 'indian', 'american', 'mexican', 'spanish', 'greek', 'latin', 'persian']
path = os.getcwd()
restaurantlist = []
for cuisine in cuisines:
    try:
        print('\nDataframe for '+cuisine)
        offset = 0
        while offset < 1000:
            result = searchRestaurants(cuisine,offset)
            for rec in result:
                restaurantlist.append((rec.get('id'),cuisine,rec.get('review count'),rec.get('rating')))
    
```

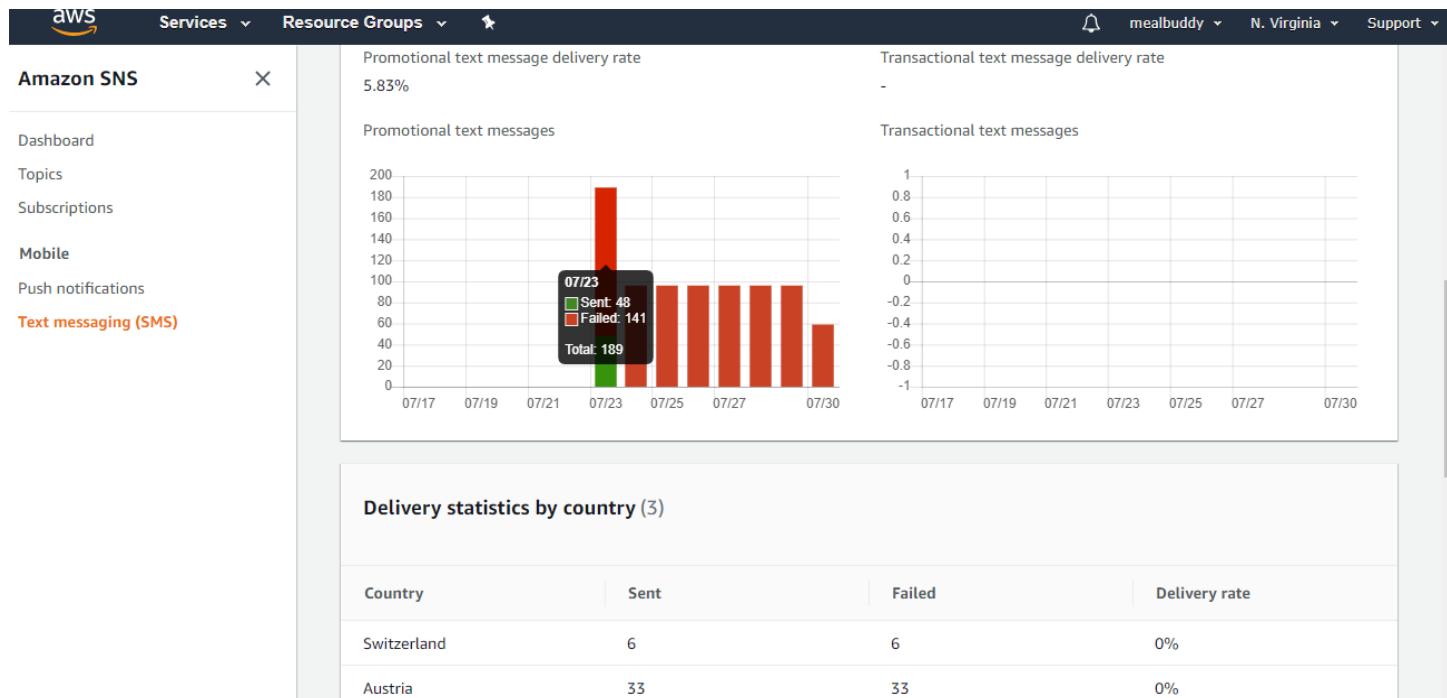
### 3. Difficulties Encountered in Reporting Week

- We were experiencing problems with sending messages through SNS. It was working during initial tests. We dug deeper into the problem and found that we have reached the monthly limit of 1 USD.

- b. (Optional) For **Account spend limit**, enter the amount (in USD) that you want to spend on SMS messages each calendar month.

**⚠ Important**

- By default, the spend quota is set to 1.00 USD. If you want to raise the service quota, [submit a request](#).
- If the amount set in the console exceeds your service quota, Amazon SNS stops publishing SMS messages.
- Because Amazon SNS is a distributed system, it stops sending SMS messages within minutes of the spend quota being exceeded. During this interval, if you continue to send SMS messages, you might incur costs that exceed your quota.



- So as recommended by AWS we raised the ticket to increase service limit

Case ID 7229417361 [Info](#)[Resolve case](#)**Case details**

## Subject

Limit Increase: SNS

## Case ID

7229417361

## Created

2020-07-30T13:53:45.457Z

## Case type

Service limits

## Opened by

mealbuddyservices@gmail.com

## Status

Unassigned

## Severity

General question

## Category

Service Limit Increase, SNS

## Additional contacts

-

**4. Tasks to Be Completed in Next Week**

- **Continue with final report**
- **Continue with Presentation**
- **Demo test run**
- **Test scenarios**