

Big Data Tools 2

GROUP PROJECT

Professor: Steven HOORNAERT

**Group 1**

SEO Yerim

TIRUMALE LAKSHMANA RAO Kiran

ZHENG Bin

IESEG School of Management, Masters in Big Data Analytics for Business, Lille, France.

**C:\Users\Joseph\Desktop\ieseg1.jpg**

Table of Contents

[**1. The Yelp Project** 3](#_Toc70102516)

[**1.1 Introduction** 3](#_Toc70102517)

[**1.2 Project Setup** 3](#_Toc70102518)

[**2. The Yelp Business** 3](#_Toc70102519)

[**4. Project Implementation in PySpark** 6](#_Toc70102520)

[**4.1 Reading in the Data** 7](#_Toc70102521)

[**4.2 Dropping the Unnecessary Columns** 7](#_Toc70102522)

[**4.3 Joining all 6 Tables** 8](#_Toc70102523)

[**4.4 Pre-processing the base table** 9](#_Toc70102524)

[**4.5 Modelling** 11](#_Toc70102525)

[**5. Conclusion and Recommendation** 12](#_Toc70102526)

[**6. Reference** 12](#_Toc70102527)

# **1. The Yelp Project**

## **1.1 Introduction**

2020 was abnormal. ‘We cannot go back to the old world, before global pandemic declaration of the COVID-19 Pandemic’ is what majority of medical professionals are saying. Due to the unexpected Covid-19, lives of people have changed dramatically. The way people eat, get dressed and socialize is totally different from what it used to be. More than anything, the Covid-19 pandemic hit the leisure industry the hardest in particularly the restaurants, bars, and food-chain industries. The social media style customer review website, Yelp, is where customers can share their experiences with businesses including restaurants.

Hired by Yelp, we were provided with the following 6 datasets[2] [3] to build a prediction model to forecast which businesses will start doing delivery/takeout after the first lockdown in the North American region.

The business informed datasets (5 datasets) are from March 2020, and the Covid dataset is from June 2020, after the first wave of covid. The goal is to identify and predict what factors lead some businesses to start doing **delivery or takeout** for the first time after the first lockdown.

Precise prediction, which can be regarded as an advance in digital transformation, will lead higher profits to our company, Yelp. Beyond the explicit and short-term prediction of providing list of restaurants that is about to start doing delivery or takeout, we, as a data team of Yelp, want to know what impact the COVID has brought into our lives. With the prediction, we will be able to provide the deep insights to both service-providers and service-users

## **1.2 Project Setup**

For this project, we used PySpark to leverage Spark’s in-memory, distributed processing engine that allows processing data efficiently in a distributed fashion, and much faster than traditional Hadoop MapReduce.

For the project pipeline, we followed the following 5 steps.

1. Reading in the Data
2. Dropping Unnecessary Columns
3. Joining all 6 datasets
4. Pre-processing the basetable
5. Modelling

Technical details of the project follow later in this report.

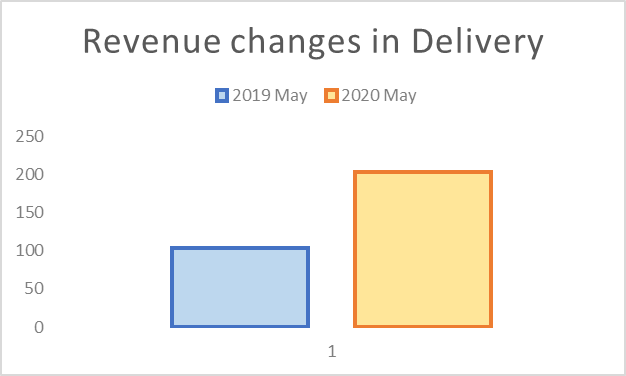
# **2. The Yelp Business**

COVID19 have changed the consumers’ behaviors. People tends to stick to their habits. Seeing the fact that people barely change their habits, we are expecting that even after the vaccine, there still will be the influence of COVID on people’s behaviors. Therefore, it is important to figure out what have been changed during the pandemic era since it will affect our near future as well.

Business from different industries suffered from COVID financially. People were forbidden or sometimes reluctant to go out and enjoy their lives as before. In consequences, almost all the business sectors such as flights, sports, clothing, cosmetics, restaurants and bars had no choice but to endure during the pandemic. The pandemic has stayed longer than we expected. Therefore, people cannot just wait for the pandemic to be passed. According to the Banque de France, GDP decreased 5.8% and 10% each in first and second quarters of 2020. Due to the lockdown caused by COVID, French people, for example, went through toughest economic crisis ever since the Second World War.

Food-tech is a field that has remarkably grown in the world since the COVID outbreak. Food-tech is a word that refers to a new industry that combines food and technology, including food search, recommendation, delivery, and delivery of food ingredients. According to the statistics specialist Nielsen Scantrack, the growth rate of commerce in the food delivery industry in the first week of March, in the early days of COVID, reached 31.2% compared to the same period of the previous year.

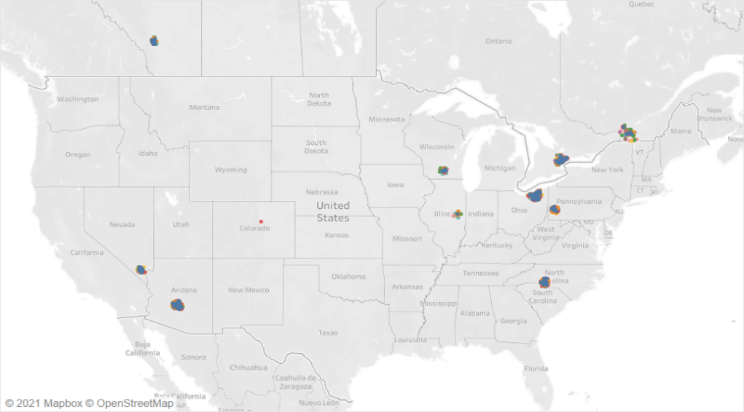
Customers decided to stay at home when feeding, drinking, and doing their gatherings during the pandemic. The revenues of delivery company have increased by 103 percent throughout the pandemic. Food-tech is making a breakthrough in the World food service industry, which has been stagnant due to COVID.



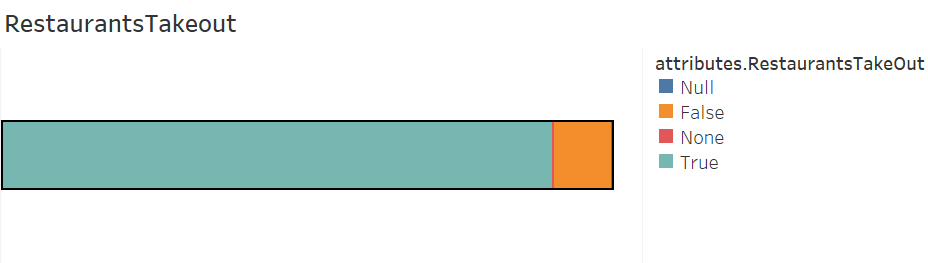
Nothing has been changed a lot from 2019 May to 2020 May. We can conclude that those big growth has been resulted from the Pandemic. There are some business actually bred the crisis as an opportunity.

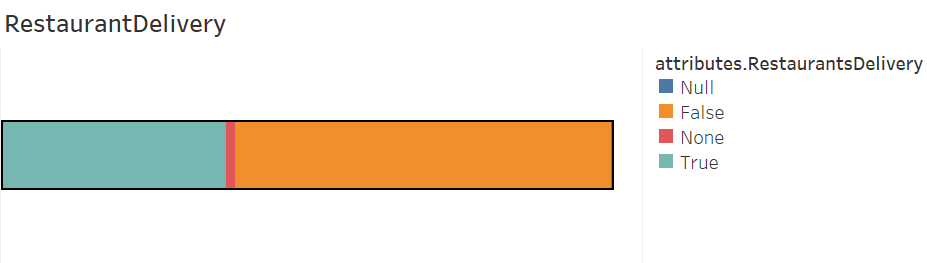
However, there are also lot of industries which faced a financial crisis during the pandemic. Food Service Vision, a consulting firm specializing in catering services, reported that the French catering industry suffered a loss of 8.8 billion euros at the beginning of last year due to COVID19. After the lockdown, restaurants and cafes were temporarily closed, and when to resume the business was unpredictable. In this situation, the only way restaurants could continue their business was delivery and take-out services. Restaurants gradually resumed operations through delivery platforms and delivery services, and by April of 2020, 49% of restaurants were able to resume operations. Now, it is time for the catering service company to change its crisis as an opportunity as well. This is the reason, we, as a data team of Yelp is providing an insight with the whole lot of data.

Our Data is mainly taking the place in the US as shown in the map below. So, we need to closely look at the special habits of people from US, especially the eastern area. As a data team, we are speaking with the data we analyzed.

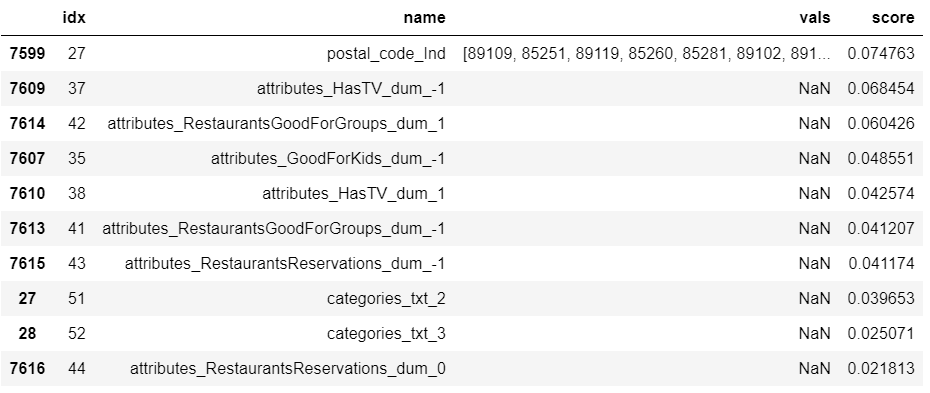


We can see that still majority of restaurants are not providing either delivery service while takeout service is provided by most of the restaurants.

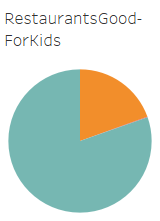
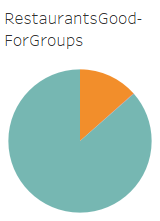
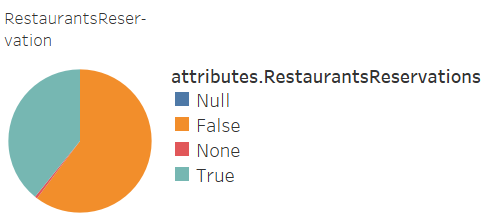
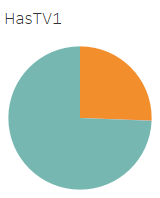




Below are the variables which affects a lot on our target variable, delivery or take out:



Postal code, availability of TV, reservation, and the concept of the restaurant affected whether if the restaurant is going to start delivery services.

We cannot know the consequences of the attributes, but still, we can guess that the restaurants with those aspects can be predicted to start their delivery soon or later. As we can see the chart above, lot of restaurants share the common traits with the restaurants that already doing the deliver services. This may imply that there are still more restaurants that may able or willing to start the delivery services.

Further informations

We predict that the growth will be centered on two axes: food-tech for hygienic purposes that minimizes human contact, and food-tech for healthy food manufacturing that transparently shares the food processing process with consumers. Considering the increasing use of applications such as 'Yuka', 'Yazio', and 'Foodvisor' that recommend healthy food and tell the origin of food, we are more confident in our predictions. Consumers have new expectations of reducing the number of tables, reducing the distribution stage, and contactless payments when experiencing the COVID-19 outbreak. Delivery and take-out services seem to be a necessity, not an option, as restaurants need to avoid the resulting loss if they reduce the number of tables.

Furthermore, we can expect that food delivering system may eventually results in companies optimized for transport, like ghost kitchens or nontraditional instruction spaces.

# **4. Project Implementation in PySpark**

As previously mentioned, for this project we leveraged the benefits of PySpark, to apply our understanding from the Big Data Tools 2 course into a real-scenario project. For the PySpark platform, we used databricks, and also a windows machine with PySpark configured in it.

Tip: A windows machine with PySpark configured in it, runs several times faster than Databricks!

Below were the 6 datasets we obtained from Yelp to know which businesses (specifically restaurant businesses) were to

1. **Business** - Contains business data including location data, attributes, and categories.
2. **Review** - Contains full review text data including the user\_id that wrote the review and the business\_id the review is written for.
3. **User** - User data including the user's friend mapping and all the metadata associated with the user.
4. **Checkin** - Checkins on a business.
5. **Tip** - Tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions.
6. **Covid** - whether a business was successful in opening up after the first covid wave or not and other covid related details of businesses

**The Project Pipeline:**

For the project pipeline, we followed the following 5 steps.

1. Reading in the Data
2. Dropping Unnecessary Columns
3. Joining all 6 datasets
4. Preprocessing the basetable
5. Modeling

The project pipeline was a cyclic process. Wherein, in order to improve the process, we went back to step 2, reworked on the previously taken decisions for each of the following step in order to improve the model’s performance.

## **4.1 Reading in the Data**

The first approach to any project, is understanding the data. Several days were spent into understanding each dataset, and each column of the dataset.

In order to understand the dataset, for each column/feature we looked at:

* Shape / size of the dataset
* Number of unique business\_id or user\_id of each dataset
* Unique values of each column/feature, to better assist us with feature selection
* Missing values
* Unique ID of each column for joining the tables into a basetable

The detailed dataset information is attached in the below excel file:



**Note:** Although as initially planned, we decided to use the entire datasets provided by Yelp, but during the joining phase is when we realized that the covid dataset recently updated by Yelp, has no significance with the other business datasets. Hence, for the purpose of the project, the professor shared with us a sample of the dataset. This report is based on this sample dataset.

## **4.2 Dropping the Unnecessary Columns**

A large number of columns were dropped for the following reasons:

* Not required, as another similar feature present (business name vs business\_id)
* During aggregations, some features such as Text and Date cannot be aggregated, hence dropped
* Noticed that after including them in the initial analysis, during modelling they were not contributing to the model’s performance

**Business Dataset**

For the business dataset, after analyzing the dataset, we agreed on dropping the following columns:

* Address, Name – Dropped as we already have a business\_id identifying each business uniquely
* latitude, longitude – Instead, we took, the city, state and pincode

attributes\_Ambience, attributes\_BYOBCorkage, attributes\_BestNights, attributes\_BusinessParking, attributes\_DietaryRestrictions,

attributes\_GoodForMeal, attributes\_HairSpecializesIn, attributes\_Music, attributes\_NoiseLevel, attributes\_RestaurantsAttire, attributes\_Smoking, attributes\_WiFi – Dropped due to nested complex data

* hours\_Monday, hours\_Tuesday, hours\_Wednesday, hours\_Thursday, hours\_Friday, hours\_Saturday, hours\_Sunday – Were not contributing to the model’s performance
* attributes\_RestaurantsPriceRange2 – Dropped after running models, were not contributing to model’s performance

**Review Dataset**

For the review dataset we dropped the following features:

This table needed to be grouped by business\_id and we aggregated each column by taking the mean. The following columns could not be kept with the above process:

* date
* review\_id'
* text

**Tip Dataset**

Like for the review dataset, for the same reasons due to aggregations, the following two features were not required:

* text
* date

**User Dataset**

The following features were dropped from the user dataset:

* elite – Not contributing to the model’s performance
* friends – Not required for our purpose
* name – Selected the user\_id instead of the name

**Covid Dataset**

The following features were dropped from the covid dataset, as we noticed that they were not contributing to the model’s performance:

* highlights
* Temporary Closed Until

## **4.3 Joining all 6 Tables**

Once we dropped the above listed columns, we had slightly cleaner datasets. We followed the following steps to join all tables and obtain our single basetable

**Step 1:**

First, we joined the review and the user datasets into one dataset called review\_user, using the user\_id column which was common to both datasets. Hence the joined (inner join) dataset now had each review with its appropriate user details.

The review dataset which consisted of 500 000 rows with 19018 unique business\_id, hence the joined dataset too consisted of the same number of rows and unique business\_id. The joined review\_user dataset then was transformed so that each row consisted of one business\_id. For the transformation, we grouped the data by business\_id and took the mean of all columns. It was appropriate to take the mean, as we had columns such as number of stars, number of reviews etc.

**Step 2:**

Taking the business dataset as the main dataset, which consisted of 19018 unique business\_id, we could now join the above created dataset (review\_user), into a dataset called yelp\_datamart.

**Step 3:**

Prior to joining the tip dataset consisted of 124161 rows and 12578 unique business\_id. First, we transformed this dataset to get a single row per business\_id. This was done by grouping the data by business\_id and taking the sum of the compliment\_count column. We aggregated by sum, as we wanted the total number of compliments per business\_id.

This transformed tip dataset was then joined to the main yelp\_datamart dataset using a left join method, as the tip dataset consisted of only 12578 unique business\_id.

**Step 4:**

Similarly, the checkin dataset which consisted of 1990914 rows and 16244 unique business\_id, had to first be transformed into one row per business\_id. Hence, we the dataset by business\_id, and took the sum of all dates (which was pre-processed), to obtain the number of checkins per business\_id.

Then this transformed checkin dataset was joined using a left join to the above yelp\_datamart.

**Step 5:**

And finally, we had to join the main covid dataset, which contained our target column. This dataset consisted of duplicate business\_ids, hence we dropped the duplicate rows, and performed an inner join with the above created yelp\_datamart.

## **4.4 Pre-processing the base table**

Pre-processing is an integral part of the machine learning pipline, and one of the most critical part. We focused quite some time and tried various methods of preprocessing. Below are the final methods of pre-processing chosen that helped us achieve the best performing model.

* Dropping columns with over 80% missing values
* Initially we took 70% as the cut off for missing values.
* The model performed better with a cut-off of 80% missing values.
* For our base table with 19018 rows, an 80% cut-off meant any column having missing values of over 15214 were dropped.
* Using this method, we were dropped a total of 15 features
* Converting Boolean columns to integer values
* A total of 14 boolean column present in our dataset, had values of True, False, Missing
* True was replaced with 1
* False was replaced with 0
* Missing and Null values with -1
* Handling the Virtual Services Offered column
  + This column was a Boolean column but needed to be processed differently than the above. As this column contained the value False, and all other values indicating True
  + Hence, False was converted to 0
  + All other values converted to 1
* Converting the Target column delivery or takeout to Double format
  + PySpark requires that the target column be called label, and that it is in a double format.
  + Renamed the column to label
  + Changed the type of the column to double.
* Pre-processing the categorical columns.
* STEP 1: Our base table consisted of 4 following categorical columns:
  + - attributes\_Alcohol
    - city
    - postal\_code
    - state
* Checked for missing values for the above 4 categorical columns. Only attributes\_Alcohol contained missing values. Replaced missing values with -1.
  + STEP 2: In the previous step, we had cleaned the Boolean columns. The Boolean columns being categorical columns, were one-hot encoded. The model indeed performed better with the one-hot encoded columns.
* Pre-processing the text columns. The following two columns were processed using tokenizing and CountVectorizer method.
  + categories
  + Covid Banner

Note: We did not need to remove stop words in our case.

* Creating a backup of the final basetable.

We decided to export our final basetable into a parquet file, to avoid having to re-run the entire pipeline till pre-processing, every time we re-start our machine / databricks.

* + Renaming few columns from the basetable.
  + Exporting the basetable to a parquet file basetable\_final\_withOHE.parquet.

## **4.5 Modelling**

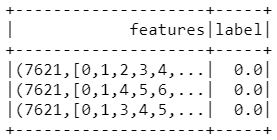
* Initially, when we ran our models, we received exceptionally high AUC scores of 0.99. Upon little analysis we found out that there were three columns that highly correlated with our target variable:
* attributes\_RestaurantsDelivery\_dum
* attributes\_RestaurantsTakeOut\_dum
* Grubhub\_enabled\_dum

The first two variables highly correlate with the target variable indeed. They indicate whether the business already had a delivery or takeout option prior to the Covid-19 Pandemic. If businesses already offered this service, they could easily continue doing so after the Pandemic. Hence these two variables didn’t were dropped.

The third variable, from the covid dataset was dropped to as it highly correlated with the target variable from the same dataset.

* Transforming the basetable for PySpark Modelling

PySpark requires that the basetable contains only two columns, features and label. To achieve this result, we used the RFormula method.



* Creating a Train and Test set
  + Create a train and test set with a 70% train, 30% test split.
  + Selecting only the features and the label columns for modelling:

**Modelling**

For the modelling, used several algorithms. The modelling pipeline is as follows:

* Defining the model
* Setting hyperparameters specific to the model
* Training the model on the train dataset
* Performing a cross validation (10-fold)
* Getting the model’s performance on the test dataset
* Computing the performance metrics
* Area under precision/recall curve
* area under Receiver Operating Characteristic curve
* Test Error = 1 – accuracy

Given that we already had target label, this project was a Supervised Learning project, focussed on Classification. Hence, since the beginning of the project, we focussed on modelling our basetable for the purpose of using the following classification algorithms:

* Random Forest
* Logistic Regression
* Decision Trees
* Gradient Boosting

# **5. Conclusion and Recommendation**

We measured the 4 model’s performances. The below table indicates the models’ performances:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Random Forest | Logistic Regression | Decision Trees | Gradient Boosting |
| Area under PR | 0.82 | 0.87 | 0.65 | 0.65 |
| Area under ROC | 0.84 | 0.92 | 0.84 | 0.85 |
| Test Error | 0.12 | 0.07 | 0.18 | 0.18 |
| Cross Validation | 10 | 10 | 10 | 10 |
| Hyperparameter Tuning | Y | Y | Y | N |

As we can see the Logistic Regression performed the best with the highest AUC and the smallest test error. Hence, we will choose Logistic Regression as the actual model for performing prediction for this project.

If we can have a longer model training time, we would prefer to further tune the parameters of each model, because it can get a better model.

# **6.** **Reference**

[1] Dr. Steven Hoornaert Course – Big Data Tools – Handouts (Slides and Notebooks)

[2] <https://www.yelp.com/dataset/download>

[3] <https://www.yelp.com/dataset/documentation/main>

[4] <https://sparkbyexamples.com/pyspark-tutorial/>