Data Profiling and Transformation

R Notebook

Loading Raw file in R markdown

```
melbourne_data <- data.frame(read.csv("Raw_good_columns.csv", header = TRUE))

# Check for NAs in Raw file
sapply(melbourne_data, function(x) sum(is.na(x)))
```

```
##                      id                    name
##                       0                       0
##                 summary                   space
##                       0                       0
##             description       neighborhood_overview
##                       0                       0
##                   notes                 transit
##                       0                       0
##                  access             interaction
##                       0                       0
##             house_rules                 host_id
##                       0                       0
##               host_name               host_since
##                       0                       0
##           host_location         host_response_time
##                       0                       0
##       host_response_rate          host_is_superhost
##                       0                       0
##       host_verifications      host_identity_verified
##                       0                       0
##                  street            neighborhood
##                       0                       0
##                    city                  suburb
##                       0                       0
##                   state                 zipcode
##                       0                       0
##          smart_location                 country
##                       0                       0
##                latitude               longitude
##                       0                       0
##        is_location_exact           property_type
##                       0                       0
##               room_type              accommodates
##                       0                       0
##               bathrooms                bedrooms
##                      17                       5
##                    beds              amenities
```

```
##                   34                    0
##                price             weekly_price
##                    0                20371
##          monthly_price         security_deposit
##                21004                 7494
##          cleaning_fee         guests_included
##                 5646                    0
##          extra_people         minimum_nights
##                    0                    0
##          maximum_nights         has_availability
##                    0                    0
##       availability_30         availability_60
##                    0                    0
##       availability_90        availability_365
##                    0                    0
##      number_of_reviews              first_review
##                    0                    0
##           last_review      review_scores_rating
##                    0                 5706
##  review_scores_cleanliness     review_scores_checkin
##                 5713                 5734
## review_scores_communication     review_scores_location
##                 5718                 5733
##      review_scores_value          requires_license
##                 5735                    0
##                license         instant_bookable
##                    0                    0
##      cancellation_policy  require_guest_profile_picture
##                    0                    0
## require_guest_phone_verification  calculated_host_listings_count
##                    0                    0
##       reviews_per_month
##                 5242
```

Imputing mean values to number columns to remove NAs and maintain data consistency

```
melbourne_data$bathrooms[is.na(melbourne_data$bathrooms)] <- mean(melbourne_data$bathrooms
, na.rm = T)
melbourne_data$bedrooms[is.na(melbourne_data$bedrooms)] <- mean(melbourne_data$bedrooms, n
a.rm = T)
melbourne_data$beds[is.na(melbourne_data$beds)] <- mean(melbourne_data$beds, na.rm = T)
melbourne_data$weekly_price[is.na(melbourne_data$weekly_price)] <- mean(melbourne_data$weekl
y_price, na.rm = T)
melbourne_data$monthly_price[is.na(melbourne_data$monthly_price)] <- mean(melbourne_data$mo
nthly_price, na.rm = T)
melbourne_data$security_deposit[is.na(melbourne_data$security_deposit)] <- mean(melbourne_data$
security_deposit, na.rm = T)
melbourne_data$cleaning_fee[is.na(melbourne_data$cleaning_fee)] <- mean(melbourne_data$cleanin
```

```
g_fee, na.rm = T)
melbourne_data$review_scores_rating[is.na(melbourne_data$review_scores_rating)] <- mean(melbourne_data$review_scores_rating, na.rm = T)
melbourne_data$review_scores_cleanliness[is.na(melbourne_data$review_scores_cleanliness)] <- mean(melbourne_data$review_scores_cleanliness, na.rm = T)

melbourne_data$review_scores_checkin[is.na(melbourne_data$review_scores_checkin)] <- mean(melbourne_data$review_scores_checkin, na.rm = T)


melbourne_data$review_scores_communication[is.na(melbourne_data$review_scores_communication)] <- mean(melbourne_data$review_scores_communication, na.rm = T)


melbourne_data$review_scores_location[is.na(melbourne_data$review_scores_location)] <- mean(melbourne_data$review_scores_location, na.rm = T)


melbourne_data$review_scores_value[is.na(melbourne_data$review_scores_value)] <- mean(melbourne_data$review_scores_value, na.rm = T)

melbourne_data$reviews_per_month[is.na(melbourne_data$reviews_per_month)] <- mean(melbourne_data$reviews_per_month, na.rm = T)
```

Export .csv file for further cleaning

```
write.csv(melbourne_data, "melbourne_clean.csv", col.names = T)

## Warning in write.csv(melbourne_data, "melbourne_clean.csv", col.names = T):
## attempt to set 'col.names' ignored
```

Python Notebook

1. Dropping the columns not required for the analysis

```
to_drop = ['last_scraped', 'scrape_id'
           ,'listing_url', 'last_scraped'
           ,'picture_url', 'host_about'
           ,'host_thumbnail_url', 'host_picture_url'
           ,'host_has_profile_pic', 'host_url'
           ,'host_neighborhood', 'country_code'
           ,'bed_type', 'calendar_updated'
           ,'calendar_last_scraped', 'review_scores_accuracy']

mel.drop(to_drop, inplace=True, axis=1)
```

```
mel.head()
```

	id	name	summary	space	description	neighborhood_overview	notes	transit	access
0	9835	Beautiful Room & House	NaN	House: Clean, New, Modern, Quite, Safe. 10Km f...	House: Clean, New, Modern, Quite, Safe. 10Km f...	Very safe! Family oriented. Older age group.	NaN	YES ! The bus (305,309) is exactly two blocks ...	Kitchen, backyard, upstairs lounge. We'd like ...

2. Replacing Blank spaces with NA

```
import numpy as np
mel.replace('', np.nan, inplace=True)
mel
```

	id	name	summary	space	description	neighborhood_overview	notes	
0	9835	Beautiful Room & House	NaN	House: Clean, New, Modern, Quite, Safe. 10Km f...	House: Clean, New, Modern, Quite, Safe. 10Km f...	Very safe! Family oriented. Older age group.	NaN	Y bus (is ex
1	10803	Room in Cool Deco Apartment in Brunswick	A large air conditioned room with queen spring...	The apartment is Deco/Edwardian in style and h...	A large air conditioned room with queen spring...	This hip area is a crossroads between two grea...	NaN	opti trar
2	12936	St Kilda 1BR APT+BEACHSIDE+VIEWS+PARKING+WIFI+AC	RIGHT IN THE HEART OF ST KILDA! It doesn't get...	FREE WiFi FREE in-building remote controlled g...	RIGHT IN THE HEART OF ST KILDA! It doesn't get...	A stay at our apartment means you can enjoy so...	First floor apartment with both lift and stair...	apa locat
3	15246	Large private room-close to city	Comfortable, relaxed house, a home away	The atmosphere is relaxed and easy going.	Comfortable, relaxed house, a home away	This is a great neighbourhood – it is quiet, v...	A simple self service breakfast is	tra

3. Changing the column names

```
mel.rename(columns={'neighbourhood': 'city', 'city': 'suburb','name': 'listing_name'}, inplace=True)
mel.head()
```

6]:

	id	listing_name	summary	space	description	neighborhood_overview	notes	transit	access
0	9835	Beautiful Room & House	NaN	House: Clean, New, Modern, Quite, Safe. 10Km f...	House: Clean, New, Modern, Quite, Safe. 10Km f...	Very safe! Family oriented. Older age group.	NaN	YES ! The bus (305,309) is exactly two blocks ...	Kitchen, backyard, upstairs lounge. We'd like ...
1	10803	Room in Cool Deco Apartment in Brunswick	A large air conditioned room with queen spring...	The apartment is Deco/Edwardian in style and h...	A large air conditioned room with queen spring...	This hip area is a crossroads between two grea...	NaN	Easy transport options - the tram is right out...	Wifi. Bathroom and kitchen is shared but I mos...
2	12936	St Kilda 1BR APT+BEACHSIDE+VIEWS+PARKING+WIFI+AC	RIGHT IN THE HEART OF ST KILDA! It doesn't get...	FREE WiFi FREE in-building remote controlled g...	RIGHT IN THE HEART OF ST KILDA! It doesn't get...	A stay at our apartment means you can enjoy so...	First floor apartment with both lift and stair...	Our apartment is located within walking distan...	Guests have exclusive and private access to th...
3	15246	Large private room-close to city	Comfortable, relaxed house, a home away from ...	The atmosphere is relaxed and easy going. You ...	Comfortable, relaxed house, a home away from ...	This is a great neighbourhood – it is quiet, y...	A simple self service breakfast is available –...	Public transport is super convenient with a ch...	You are welcome to make yourself at home in th...
4	16760	Melbourne BnB near City & Sports	NaN	We offer comfortable accommodation in Inner Me...	We offer comfortable accommodation in Inner Me...	NaN	NaN	NaN	NaN

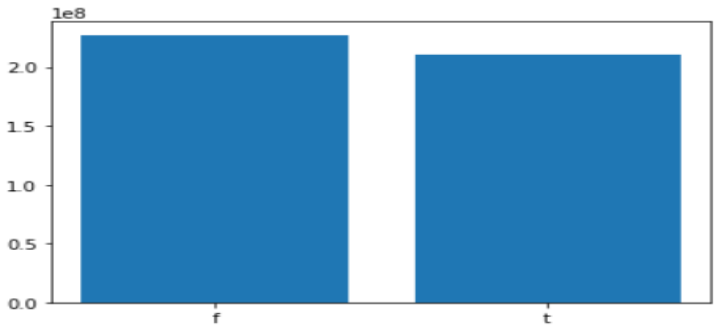
4. Exporting the data as an Excel file

```
In [21]: mel.to_excel(r'Desktop/Melbourne_cleanest.xlsx')
```

5. Plotting Graphs in Python - Host_ID vs host_is_superhost

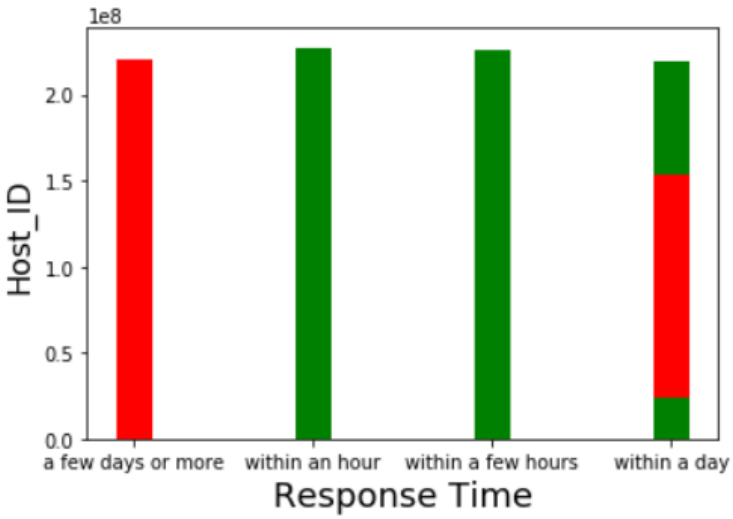
```
import matplotlib.pyplot as plt
```

```
y = melb.host_id
x = melb.host_is_superhost
plt.bar(x,y)
plt.show()
```



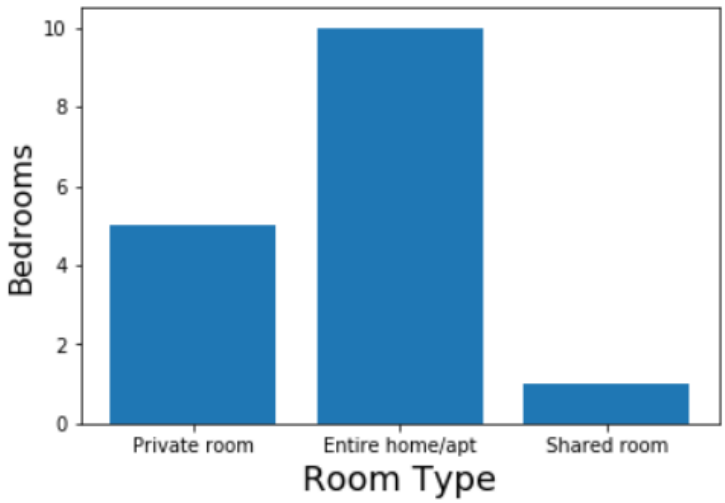
6. Difference Between Response Time

```
y = melb.host_id
x = melb.host_response_time
plt.xlabel('Response Time', fontsize=18)
plt.ylabel('Host_ID', fontsize=16)

plt.bar(x,y,width = 0.2, color = ['red', 'green'])
plt.show()
```



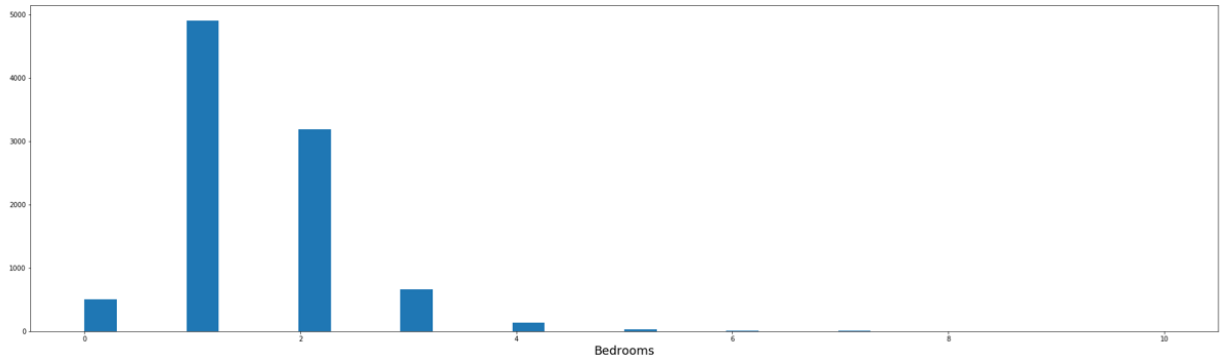
7. Types of room having bedrooms on average

```
y = melb.bedrooms
x = melb.room_type
plt.xlabel('Room Type', fontsize=18)
plt.ylabel('Bedrooms', fontsize=16)
plt.bar(x,y)
plt.show()
```



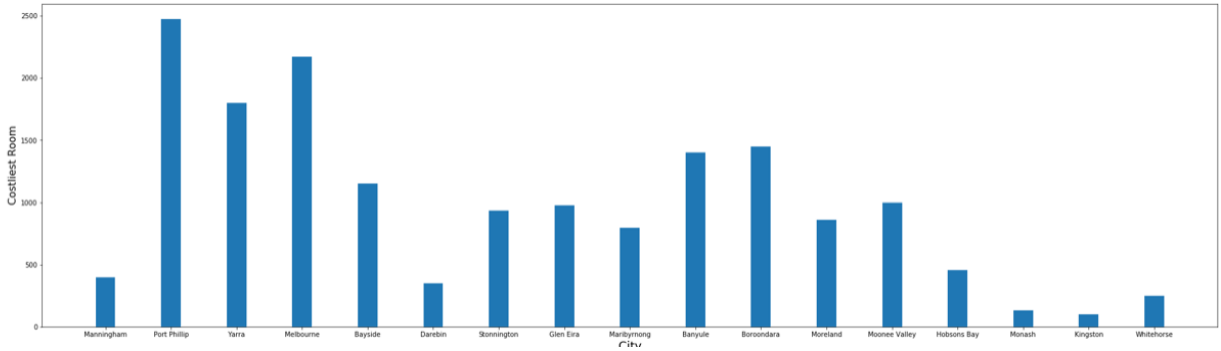
8. Airbnbs' having number of Bedrooms on average

```
y = melb.beds
x = melb.bedrooms
plt.xlabel('Bedrooms', fontsize=18)
#plt.ylabel('Beds', fontsize=16)
plt.hist(x,bins='auto', width=0.3)
plt.show()
```



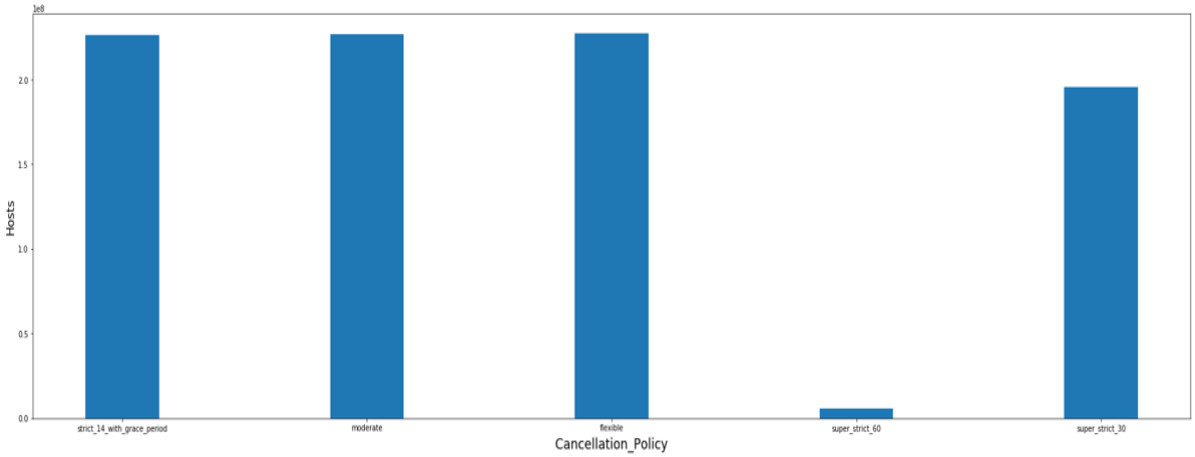
9. Costliest Room in each city

```
y = melb.price
x = melb.city
plt.xlabel('City', fontsize=18)
plt.ylabel('Costliest Room', fontsize=16)
plt.rcParams["figure.figsize"] = [32,9]
plt.bar(x,y, width = 0.3)
plt.show()
```



10. Cancellation Policy followed by Hosts.

```
y = melb.host_id
x = melb.cancellation_policy
plt.xlabel('Cancellation_Policy', fontsize=18)
plt.ylabel('Hosts', fontsize=16)
plt.rcParams["figure.figsize"] = [32,9]
plt.bar(x,y, width = 0.3)
plt.show()
```



Business Terms

- Listings: detailed listings data including full descriptions and average review score;
- Calendar: detailed calendar data for listings, including listing id and the price and availability for that day;
- Reviews, detailed review data for listings including unique id for each reviewer and detailed comments;
- Listings-Summary: summary information and metrics for listings (good for visualizations);
- Reviews-Summary: summary Review data and Listing ID (to facilitate time-based analytics and visualizations linked to a listing);
- Neighborhoods: neighborhood list for geo filter Sourced from city or open source GIS files
- Id: ID is a dataset identifier It is a globally unique value that identifies a particular metadata record
- Host_id: Host ID is a specific piece of information which uniquely identifies a host visiting the listing
- Host_name: It is the same as host id It uniquely identifies hostname
- Host_since: Tells from how long the host is active on Airbnb

- Host_response_time: The time period in which the host responds back to the customers
- Host_is_superhost: Checks whether the host is also a super host
- Host_response_rate: Time rate in which the host responds back
- Host_verifications: Verifies whether the host is real or fake
- Host_identity_verified: Determines the host whose id is verified
- Street: Shows the neighborhood where the street is
- Neighborhood: a district, especially one forming a community within a town or city
- City: a large town
- State: Tells in which state the listing belongs to
- Zipcode: postal address to assist the viewers that the listing belongs where
- Smart_location: Interactive maps and data for measuring location efficiency
- Country: Determines which country the listing belongs to
- Is_location_exact: Tells where the exact location of the listing is
- Property_type: Determines what kind of property the listing is(apartment, condo)
- Room_type: determines what type of the room it is
- Accommodates: Tells how many people can be allowed to stay in the apartment
- Bathrooms: Tells how many bathrooms the listing chosen by the customer has
- Bedrooms: Tells the customer how many bedrooms the listing has
- Beds: Tells how many beds the listing has
- Bed_type: Determines what kind of beds there are(Single, double)
- Price: Determines the price of the listing
- Security_deposit: Tells what the deposit is for renting the Airbnb
- Cleaning_fee: Shows what the fees is, for cleaning
- Guests_included: Tells the guest how many people are allowed to stay in the listing
- Minimum_nights: Tells the number of minimum nights the guest is allowed to stay
- Maximum_nights: Tells the number of maximum nights the guest can stay
- Has_availability: Determines if the listing is available or not
- Availability_30: Tells the properties that are available for 30 days or less
- Availability_90: Tells the properties that are available for 90 days or less

- Availability_365: Tells the properties that are available for 365 days or less
- Number_of_reviews: Determines how many reviews are posted for listing
- Review_scores_rating: The rating from 1-5 for hosts
- Review_scores_cleaniness: Tells the score on the scale of 1-5 for cleanliness of listing
- Review_scores_checkin: Determines the check-in is good/bad/average
- Instant_bookable: Tells if the listing is available for instant booking or not
- Cancellation_policy: The sum of money you must pay if you cancel a hotel reservation after the cancellation deadline
- Require_guest_profile_picture: Requires profile picture for some verification about the guest
- Require_guest_phone_verification: Requires phone no for verification of booked listing and sending details
- Reviews_per_month: Determines how many reviews the listing has received per month

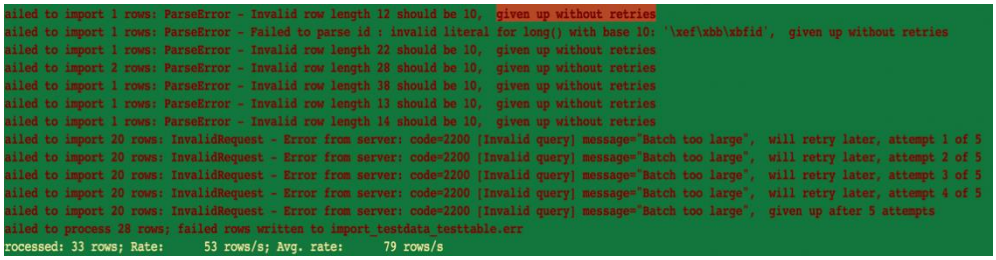
Risks/Issues

1. **Database Selection**:
    a. **Risk** – Selection of the database is the most important aspect in order to perform analysis on the data. So, if the database chosen is incorrect and not compatible with PowerBI, visualizes the data after the data is processed. This will delay the project and may also incur costs to the project adding to deadlines not meeting and losing client trust.
    b. Mitigation - Gather information for all NoSQL databases covered in the class and examine the pros and cons for all databases which are compatible with PowerBI

2. **Data Cleaning:**
    a. **Issue** – The Airbnb data consists of empty field values, redundant information, and columns which were not informational as per the objectives of the project. Incorrect information can lead to discrepancies in analysis result.
    b. Mitigation - Data profiling followed by cleaning and transformation of data using proper utility language.

3. **Installation of Database**
    a. **Issue** – We had two type of users, MAC and Windows. The steps for installation were different for each system.
    After installation on MAC, we got an error failed to get the connection when trying to reconnect to CassandraDB. After, researching found the issue was due to environment variables were commented in bash_profile.
    b. Mitigation - Gather and document the step-by-step procedure from multiple sources and complete it simultaneously in all the systems

4. **Loading of data**
    a. **Issue** – Initially the data was not clean as it had empty field and null values. Invalid row length and invalid literal errors, hence this data was not loaded into the database and it gave incorrect column values.



    b. Mitigation- Refer to the CQL queries data types and column constraints. Thereafter, load a sample data from another csv to ensure data is being loaded

5. **Selection of Utility language**
    a. **Issue**- Initially chose Python as the utility language to process the data cleaning but at later stage found it did not clean the data and the data still had empty field values.
    b. Mitigation – Firstly, examine the data set to determine the amount of cleaning required and based on the skillset of the team, determine the best tool/utility.

6. **Time Management**
    a. **Issue** – Every team member had different schedule of courses and tasks to be accomplished.
    b. Mitigation – Check the members availability and decide a common time frame, Scheduled meetings and booked locations for 2 weeks

7. **Improper Communication**
    a. **Risk** - Verbally communicating would sometimes lead to a few points being missed during the implementation. Hence, necessary to communicate and pass on the information as soon as you come across an important point.
    b. Mitigation - Created a slack channel and a WhatsApp group for effective communication

8. **Sharing of Information**
    a. **Issue** – When sharing information through email or other means, it leads to redundant information and it's difficult to get all information at one place.
    b. Mitigation - Created a shared google drive to enable effective sharing/collaboration

9. **Task Allocation**
    a. **Risk** – It is of utmost important to have the right resource with correct skillset to perform the right task. For example, a Data analyst should be allocated to a role based on their skillset, else mapping resource to different skillset from their role will lead to wastage of skills and time as well.
    b. Mitigation- Task allocation as per the skillset

10. **Discrepancy in results while analysis**
    a. **Risk** – If the output is not as per the results expected, it leads to incorrect results which may lead to misleading information.
    b. Mitigation - Manual validation at each step

11. **Missed Deadlines**
    a. **Issue** – Time is divided and allocated to each task. But there is always a possibility of missed deadlines unintentionally due to issues or errors taking time to be resolved, which creates a cascading effect leading to delays in next steps/tasks aligned, which may lead to missing the project delivery deadline.
    b. Mitigation- Allocate buffer time for each task, to be well ahead of the deadline.

Database Comparison

We chose Cassandra DB, as it is a free and open source NoSQL DB with simple interface Query Language for accessing Cassandra. As per CAP (Consistency, Availability, and Partition Tolerance) theorem, Cassandra is an AP system providing high availability and partition tolerance.
Use C Data ODBC driver for Cassandra to visualize Cassandra data in Power BI Desktop, so it offers live interaction. When we issue complex queries from Power BI to Cassandra, the operations are directly pushed to Cassandra and it can then perform unsupported SQL operations like Joins using the embedded SQL engine.
Additionally, we do not have to perform complex queries on the data set or involve real time analytics as of now. Need to perform data extraction by columns using keys. Therefore, Cassandra is preferred over MongoDB and OrientDB as well.

MongoDB and Cassandra Comparison

MongoDB	Cassandra
It supports expressive object model with objects providing the nested features	Offers traditional structure of rows and columns
Supports one master model, thus when master goes down, it does not support writes	Supports multiple master model hence is always up and running with no time lag, thus provide ability of cluster to take writes
Does not provide write scalability due to its single master model	Provides write scalability with the multiple master model. So, the more servers you have in the cluster, the better it will scale
It does not support query language	Supports CQL query language
Not much easy to use when compared to Cassandra	Ease of use
Written in SQL	**Written in Java**

We would choose Cassandra since it is easy to set up and maintain, regardless of the data growth while.

References

1. https://scalegrid.io/blog/cassandra-vs-mongodb/
2. https://blog.panoply.io/cassandra-vs-mongodb
3. https://www.dataversity.net/choose-right-nosql-database-application/

Oriental DB and Cassandra Comparison

Cassandra	Orientdb
Cassandra is an open source, a column-oriented database designed to handle large amounts of data across many commodity servers.	OrientDB is a multi-model database, supporting graph, document, key/value, and object models.
It is a wide-column store based on ideas of BigTable and DynamoDB.	It is a multimodel DBMS.
SQL like SELECT, DML, DDL Statements are used in here.	SQL like query language, no joins are used in here.
No server-side scripts in Cassandra.	Server-side scripts are Java and Javascript.
Cassandra is more scalable and easier to setup.	OrientDB is less scalable compared to Cassandra.

Hence, yet again We would choose Cassandra since it is easy to set up and maintain, regardless of the data growth while.

References:

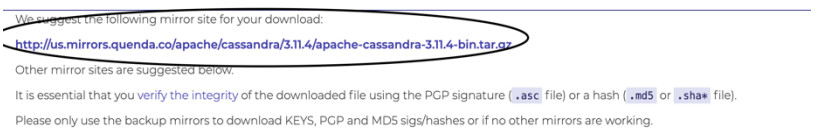
1. https://db-engines.com/en/system/Cassandra%3BOrientDB
2. https://www.g2.com/compare/cassandra-vs-orientdb
3. https://shuaiw.github.io/2017/08/18/choosing-a-nosql-db.html

Cassandra Installation

For MAC:

1. Download Cassandra 3.11.4 tar file from apache.org:

http://www.apache.org/dyn/closer.lua/cassandra/3.11.4/apache-cassandra-3.11.4-bin.tar.gz



2. Double click the downloaded file to unzip it.
3. Execute the following commands using the terminal to set Cassandra environment variables:

i. **cd Desktop/** (the unzip folder location for apache-cassandra-3.11.4-bin, in my case it is desktop)
ii. sudo mv apache-cassandra-3.11.4 /usr/local/cassandra
iii. Enter the password for your system
iv. **cd** (move to the user level)
v. **open .bash_profile** -- update the bash file with the below information for cassandra
export CASSANDRA_PATH=/usr/local/cassandra
export PATH=$PATH:$CASSANDRA_PATH/bin
vi. source .bash_profile

```
Prernas-MacBook-Pro:~ prerna$ cd Desktop/
Prernas-MacBook-Pro:Desktop prerna$ sudo mv apache-cassandra-3.11.4 /usr/local/cassandra
Password:
Prernas-MacBook-Pro:Desktop prerna$ cd
Prernas-MacBook-Pro:~ prerna$ open .bash_profile
Prernas-MacBook-Pro:~ prerna$ source .bash_profile
Prernas-MacBook-Pro:~ prerna$ which cassandra
/usr/local/cassandra/bin/cassandra
```

4. Open new terminal and start Cassandra using: **cassandra -f** then press enter
5. Then enter the command **cqlsh**: Test Cluster is connected
6. To confirm the connection: **DESCRIBE KEYSPACES**

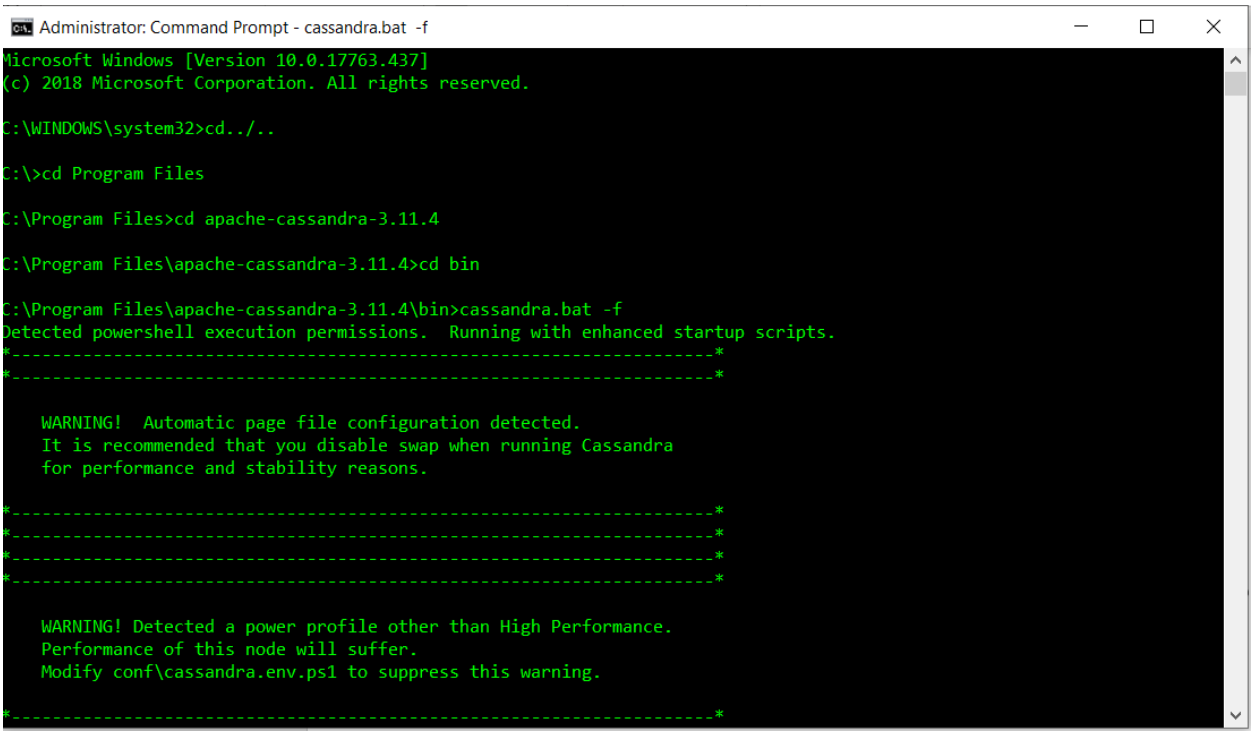
```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> describe keyspaces;

system_traces  system_schema  system_auth  system  system_distributed
```

For Windows:

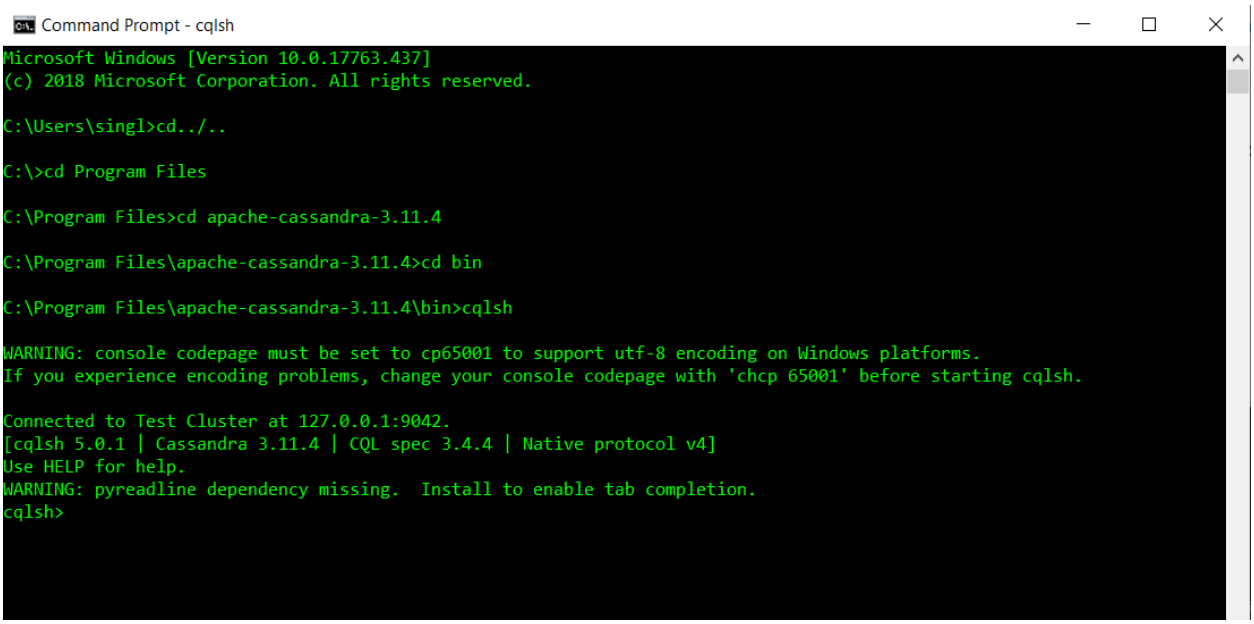
1. Download Cassandra 3.11.4 tar file from the link below

http://www.apache.org/dyn/closer.lua/cassandra/3.11.4/apache-cassandra-3.11.4-bin.tar.gz

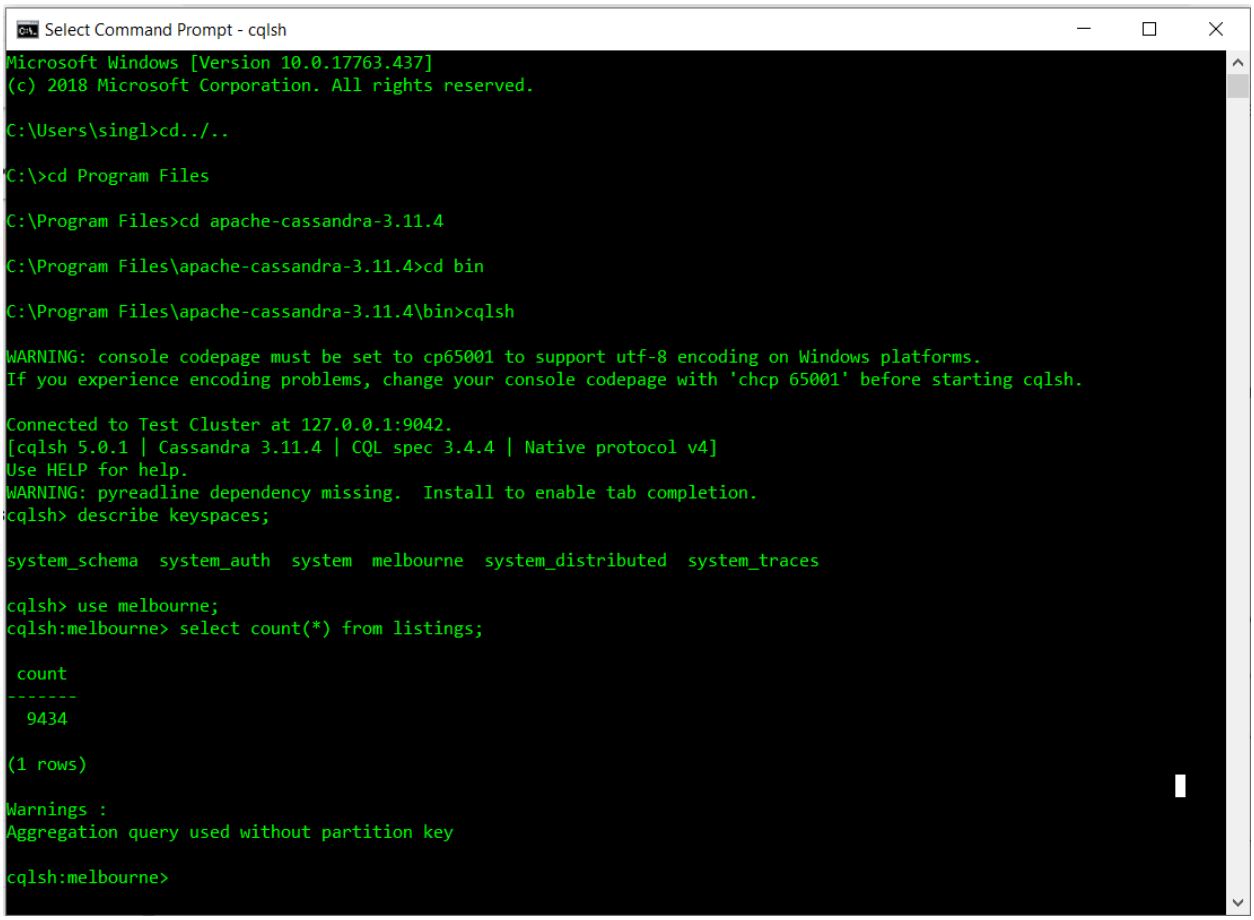
2. Extract the folder in the C drive
3. Once installed, open Command Prompt in administrator mode
4. Open the root directory to reach the bin folder of Cassandra and execute the .bat file to run Cassandra



5. Run another CMD to access the bin folder of Cassandra to execute the cqlsh command



6. Create keyspace and use it (Desc keyspace)
7. Create tables by running a script
8. Execute the query and validate using a Select query


Select Command Prompt - cqlsh
Microsoft Windows [Version 10.0.17763.437]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\singl>cd../..
C:\>cd Program Files
C:\Program Files>cd apache-cassandra-3.11.4
C:\Program Files\apache-cassandra-3.11.4>cd bin
C:\Program Files\apache-cassandra-3.11.4\bin>cqlsh
WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing. Install to enable tab completion.
cqlsh> describe keyspaces;
system_schema system_auth system melbourne system_distributed system_traces
cqlsh> use melbourne;
cqlsh:melbourne> select count(*) from listings;
count
9434
(1 rows)
Warnings :
Aggregation query used without partition key
cqlsh:melbourne>