Business Intelligence assignment

1) When jio launched publicly in september 2016, it provided free internet to everyone acquiring a jio connection all over india. The tier 1 customers were not attracted to the offer on a large scale as the tier 2 and tier 3 customers were. It was a chance for the lower tier users to freely surf the internet without having to pay a penny.
   The video/content applications like Youtube and tiktok have a very user friendly platform which helps them to gain and attract more users easily. As on the other hand the online ordering applications have a little more complicated platform as compared to the former.
   The first point could be that the lower tier users are not as smart as the upper tier users and even today they seek the help of others while ordering goods online as it includes filling in the forms, typing a lot of things, a payment portal and many others things.
   Online payment has always been problematic to the lower tiers in india. In the applications like youtube and tik tok, an user does not have to pay for watching a content. Let's leave the tier thing for once and consider this example- my mother searches for an item she needs, and then asks me to order it for her always but she uses youtube all the day for watching some cooking videos, gardening videos.
   Second point could be that the video content applications provide entertainment, knowledge, etc. while the other does not have something of the similar kind in free of cost.
   One other point could be that video content applications use a large amount of data for internet surfing a few times. Before jio the data provided by other companies were limited but jio removed the data constrain by which the customers are more attracted towards free entertainment whereas for flipkart and amazon, they never used high data and were not affected on a large scale by the free internet provided by jio
   Applications like flipkart and amazon are not so successful in the rural area as they are in urban areas because it takes time to deliver items in the rural areas about a week. People in need of supplies rather buy it from the local shops and do not wait for a week to receive the order. Let's take an example- I live in a town named Jaigaon which is located in west bengal and it shares its border with bhutan. It takes about 7-10 days for an order to reach my place.Essential items has to be bought by a local supermarket.
   Applications like youtube and tik tok are made for knowledge and entertainment and this genre will always have a huge growth in its audience as compared to applications like flipkart and amazon. Even the kids of 4-5 years of age know how to open youtube applications and use it to watch cartoons.
   The free internet provided by jio helped these platforms to gather a large amount of audience.

2) Let's look into the data that was provided:
   The number of new installs grew at the same rate as it was growing.
   The number of first time buyers was almost doubled. This shows that free shipping for the first time buyers idea was a success
   By looking in the fourth column it can be said that the revenue generated has increased but the average billing per order has decreased.

The first time buyers who purchased for a second time has also increased.
The overall metrics shows that this was a success overall as the revenue earned by the company has increased in regards of the first time buyers but we have not been provided with the data of the existing customers by which we cannot say that whether some of the first time buyers are the already existing customers who have made a new account to avail the free shipping offer for the first time. As you can see the ratio of ftbs and ftbs who purchased for a second time was 2:1 but that decreased after July to 5:2 approximately which shows that more customers now only avail the free shipping order.
By offering “ 10000 products under 999” the average billing decreased as more people tend to buy cheaper products.
But by looking at the metrics it is a success for the company.

3)

A. Users will spend more time on the application watching the movies and less time searching for one. If the recommendations are good then user retention will increase and users will also use the recommendations to watch similar kinds of movies.
B. Goals of the feature should be to increase the watch time of the current users and to attract new users. Metrics for this feature should be movies of the same genre, movies from the same actors, movies from the same directors, etc.
C. Before launching this feature, the company should take a poll from the current users if they would want this feature in the application so that the users will get the knowledge of what this company is planning to bring up for the users. And after some time when the feature is ready to roll, an in-app popup and the carousel with the new recommendations with the home screen should be displayed. This feature could be evaluated by the increase in the watch time by users or by taking a quick feedback through rating by the customers.

Android Assignment

2.Let’s see the concurrent modification exception scenario with an example.

```
package com.journaldev.ConcurrentModificationException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

public class ConcurrentModificationExceptionExample {
    public static void main(String args[]) {
        List<String> myList = new ArrayList<String>();
        myList.add("1");
        myList.add("2");
        myList.add("3");
        myList.add("4");
        myList.add("5");
        Iterator<String> it = myList.iterator();
        while (it.hasNext()) {
            String value = it.next();
            System.out.println("List Value:" + value);
            if (value.equals("3"))
                myList.remove(value);
        }
        Map<String, String> myMap = new HashMap<String, String>();
        myMap.put("1", "1");
        myMap.put("2", "2");
        myMap.put("3", "3");
        Iterator<String> it1 = myMap.keySet().iterator();
        while (it1.hasNext()) {
            String key = it1.next();
            System.out.println("Map Value:" + myMap.get(key));
            if (key.equals("2")) {
                myMap.put("1", "4");
                // myMap.put("4", "4");
            }
        }
        }
}
```

Above program will throw java.util.ConcurrentModificationException when executed
The following methods can be used to avoid ConcurrentModificationException:

1. You can convert the list to an array and then iterate on the array. This approach works well for small or medium size list but if the list is large then it will affect the performance a lot.
2. You can lock the list while iterating by putting it in a synchronized block. This approach is not recommended because it will cease the benefits of multithreading.
3. If you are using JDK1.5 or higher then you can use **ConcurrentHashMap** and **CopyOnWriteArrayList** classes. This is the recommended approach to avoid concurrent modification exception.
4. You can use the iterator remove() function to remove the object from underlying collection object. But in this case, you can remove the same object and not any other object from the list.

   Let’s run an example using Concurrent Collection classes.

```
package com.journaldev.ConcurrentModificationException;

import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;
import java.util.concurrent.OnWriteArrayList;

public class AvoidConcurrentModificationException {

public static void main(String[] args) {

        List<String> myList = new OnWriteArrayList<String>();

        myList.add("1");
        myList.add("2");
        myList.add("3");
        myList.add("4");
        myList.add("5");

        Iterator<String> it = myList.iterator();
        while (it.hasNext()) {
            String value = it.next();
            System.out.println("List Value:" + value);
            if (value.equals("3")) {
                myList.remove("4");
                myList.add("6");
                myList.add("7");
            }
        }
        System.out.println("List Size:" + myList.size());

        Map<String, String> myMap = new ConcurrentHashMap<String, String>();
        myMap.put("1", "1");
        myMap.put("2", "2");
        myMap.put("3", "3");

        Iterator<String> it1 = myMap.keySet().iterator();
        while (it1.hasNext()) {
            String key = it1.next();
            System.out.println("Map Value:" + myMap.get(key));
            if (key.equals("1")) {
                myMap.remove("3");
                myMap.put("4", "4");
                myMap.put("5", "5");
            }
        }

        System.out.println("Map Size:" + myMap.size());
    }

}
```

The output of the above program is shown below. You can see that there is no ConcurrentModificationException being thrown by the program

```
List Value:1
List Value:2
List Value:3
List Value:4
List Value:5
List Size:6
Map Value:1
Map Value:2
Map Value:4
Map Value:5
Map Size:4
```

From the above example it's clear that:

1. Concurrent Collection classes can be modified safely, they will not throw ConcurrentModificationException.
2. In case of CopyOnWriteArrayList, iterator doesn't accommodate the changes in the list and works on the original list.
3. In case of ConcurrentHashMap, the behaviour is not always the same.
4. Use for loop to avoid java.util.ConcurrentModificationException

If you are working on single-threaded environment and want your code to take care of the extra added objects in the list then you can do so using for loop rather than an Iterator.

```
    for(int i = 0; i<myList.size(); i++){
            System.out.println(myList.get(i));
            if(myList.get(i).equals("3")){
            myList.remove(i);
            i--;
            myList.add("6");
    }
}
```