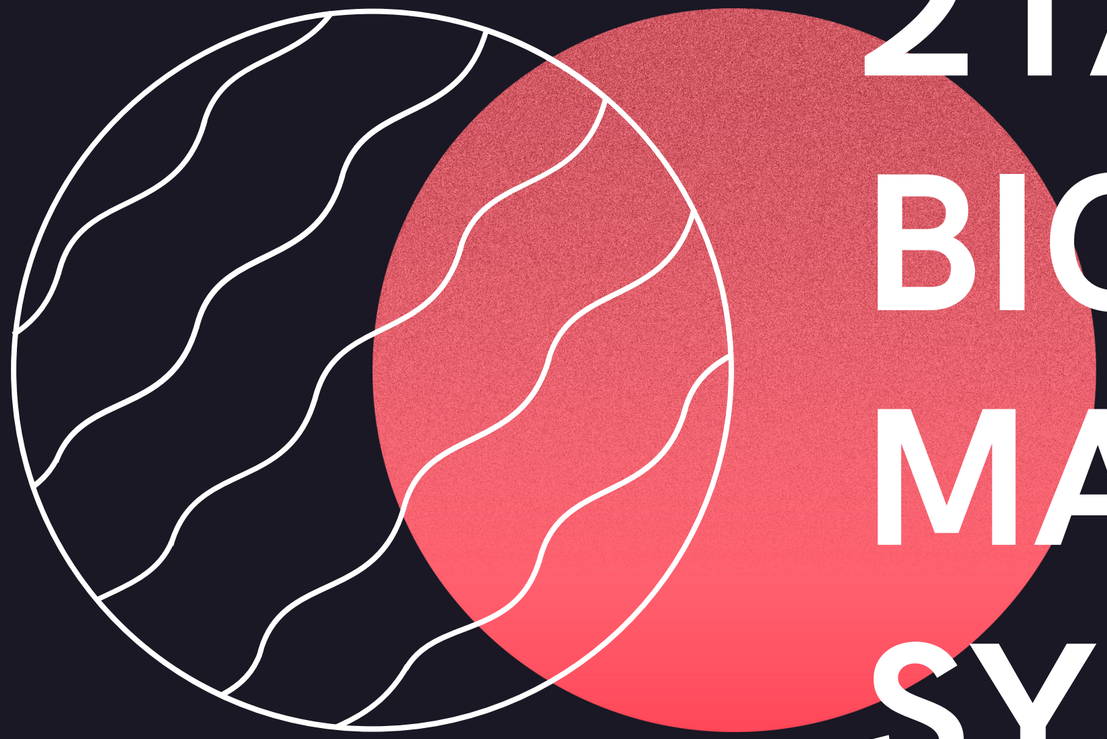


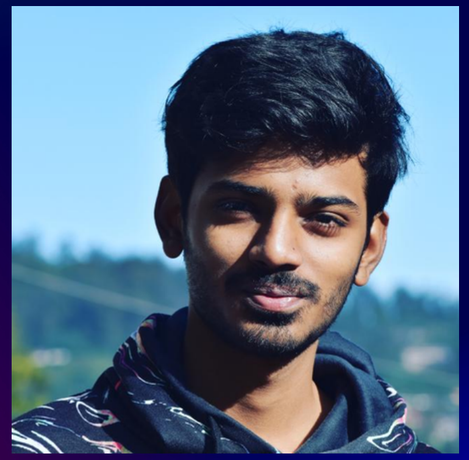
21AIE304 BIG DATA and MANAGEMENT SYSTEMS

TEAM-01

MEMBERS

MANISH NADELLA

CB.EN.U4AIE20037



NIKHIL PALETI

CB.EN.U4AIE20046

PRANAV UNNIKRISHNAN

CB.EN.U4AIE20053

Real Time Sales Management

- Stream data (shopping invoices) from client to database
- Gain insights on the data (EDA)
- Use Recommendation system to generate recommendations to customers

Flow of Presentation...

Part-I: Real Time Data Streaming

Part-II: Data Analysis

Part-III: ML application on data

HIVEMQ

NVIDIA NETWORKING

APACHE Spark™

mongoDB.

databricks

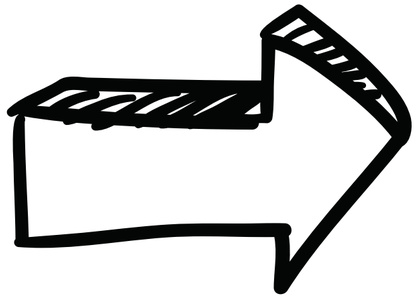
Streamlit

03

Data Streaming

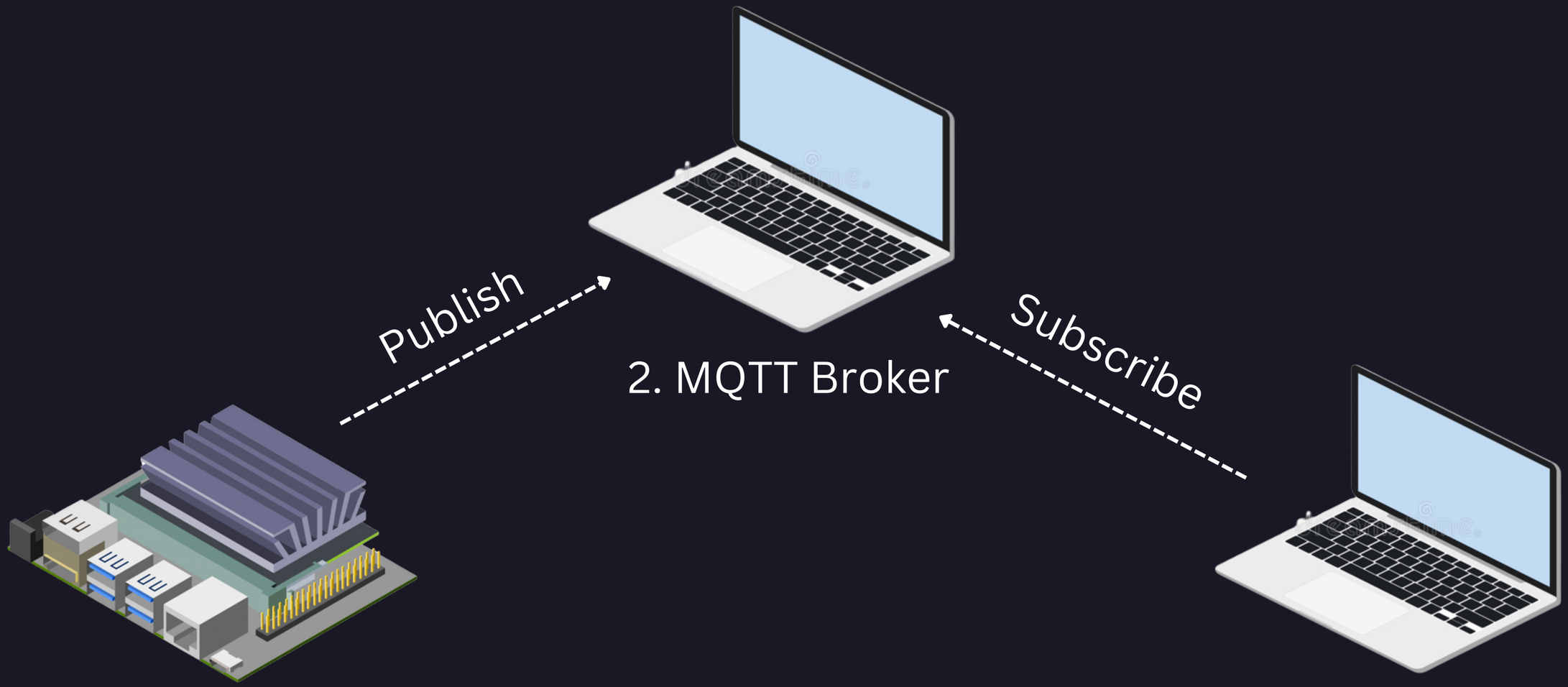
Main Idea

Customer checksout at a counter in store

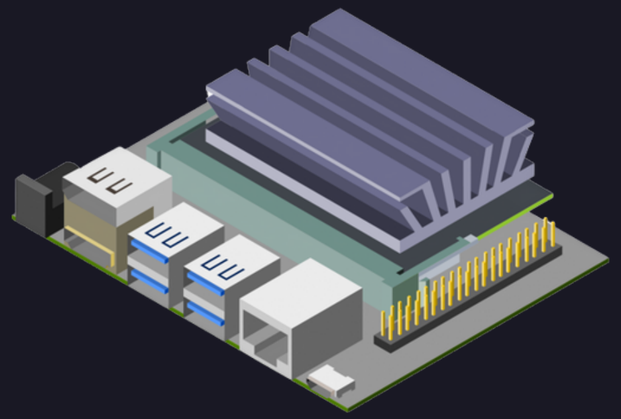


Stream the data(invoice) to a database server.

ARCHITECTURE



Client

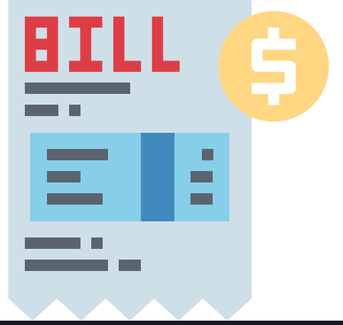


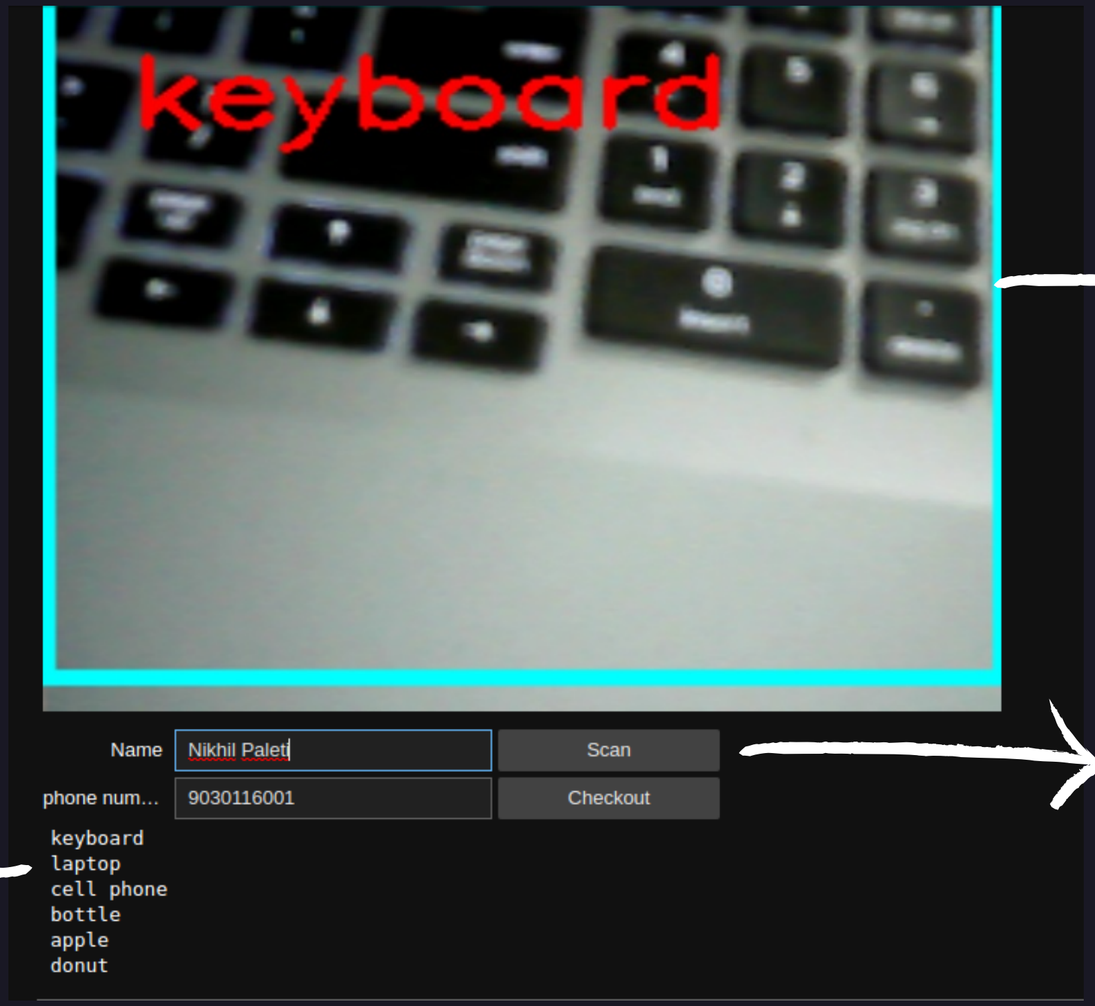
1. Get:
   - Customer Name
   - Phone Number
   - List of Products

2. Connect With MQTT Broker

3. Publish the invoice to a topic

CREATING INVOICE



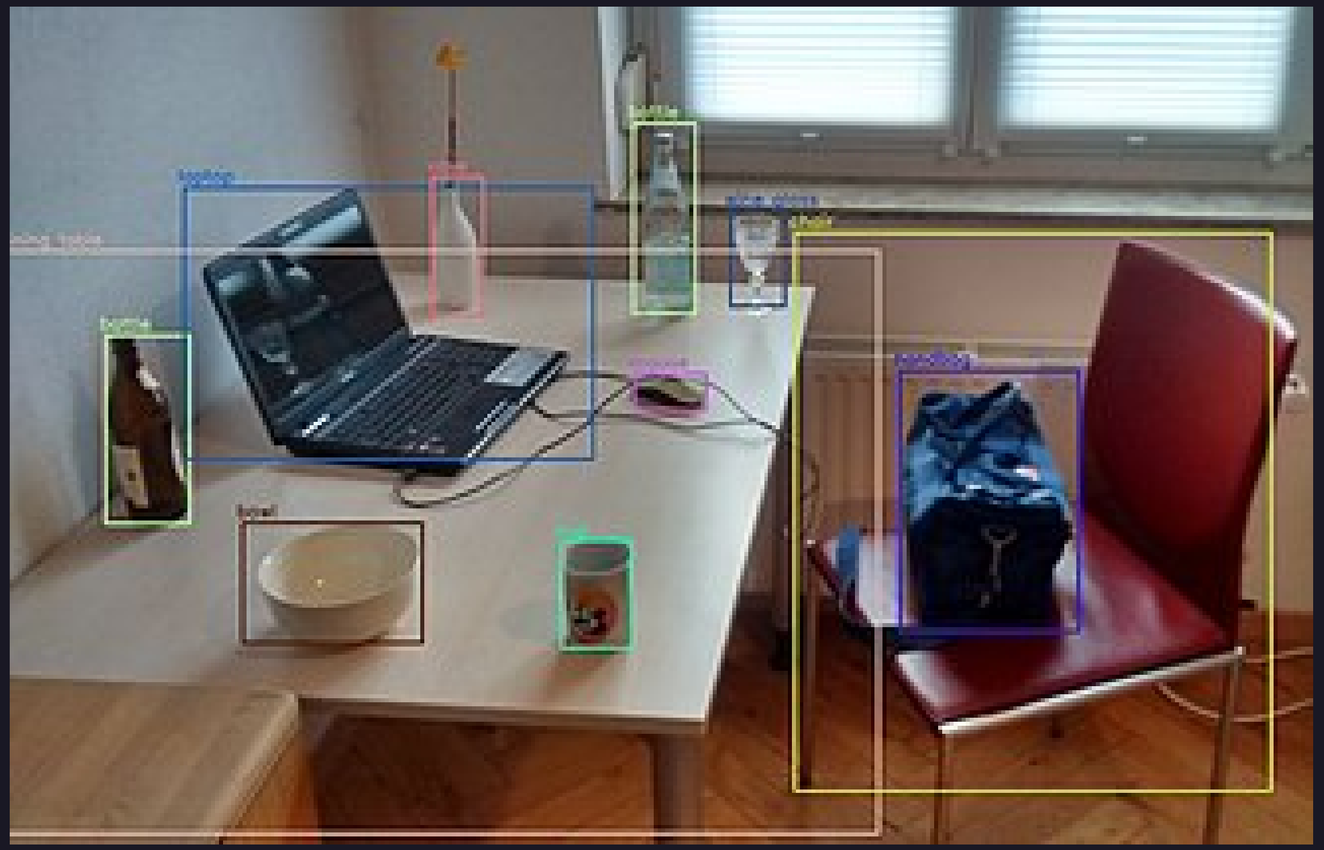


Camera View

Scan and checkout

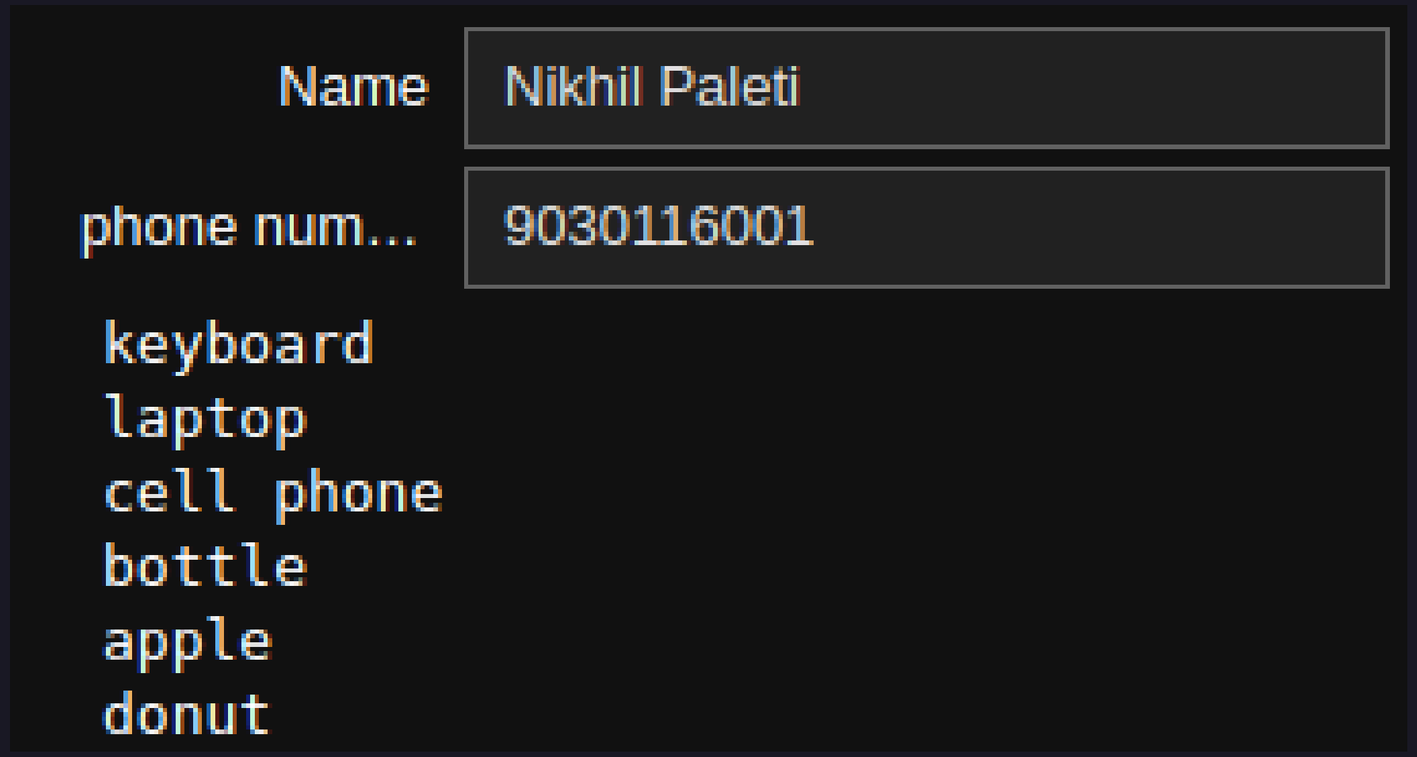
Invoice data

OBJECT DETECTION



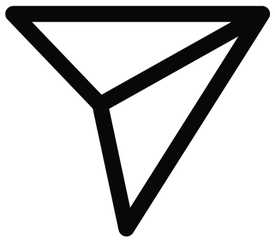
Network used: MobileNetSSD

SENDING MESSAGE



'Nikhil Paleti 9030116001 keyboard,laptop,cellphone,bottle,apple,donut'

SENDING MESSAGE



send data to broker

```
def send_data(final_string):
    global products_list, out
    client = mqtt.Client()
    client.connect("192.168.136.253", 1883, 60)
    print("connected")
    client.publish("test", final_string)
    client.disconnect()
```

Data from MQTT Broker

1. Actively Listen to incoming data from client

2. Upon receiving data, Pusblish it to database server

Server

1.Read data from MQTT
and publish it to a kafka topic

2. Use Spark Streaming to read
data from Kafka topic

3. Use spark write to publish
data to MongoDB Atlas

1.READ DATA FROM MQTT

Read from MQTT

```
client = mqtt.Client(client_id=None)

on_message = lambda client, userdata, msg: sendmessage2kafka(msg.payload.decode())
client.on_message = on_message

client.connect("192.168.136.253", 1883, 60)
client.loop_forever()
```

PUBLISH IT TO A KAFKA TOPIC

Publish it to Kafkatopic

```
producer = KafkaProducer(
bootstrap_servers='localhost:9092',
value_serializer=lambda x: dumps(x).encode('utf-8')
)

producer.send(kafka_topic, messages)
```

2. USE SPARK STREAMING TO READ DATA FROM KAFKA TOPIC

read data from kafka topic

```
spark = SparkSession.builder.appName("AISupermarket").master("local[*]") \
    .config("spark.mongodb.input.uri",
("mongodb+srv://admin:admin@cluster0.75ml3wq.mongodb.net/AISupermarket.Invoice")) \
    .config("spark.mongodb.output.uri",
("mongodb+srv://admin:admin@cluster0.75ml3wq.mongodb.net/AISupermarket.Invoice")) \
    .getOrCreate()

spark.sparkContext.setLogLevel("ERROR")
df = spark.readStream.format("kafka").option("kafka.bootstrap.servers",
"localhost:9092").option("subscribe", "invoice_kaf").load()

df1 = df.selectExpr("cast(value as string) as kafka_output")

df1.writeStream.format("console").option("truncate","false").outputMode("Append").start()
```

3. USE SPARK WRITE TO PUBLISH DATA TO MONGODB ATLAS

Write it to mongoDB

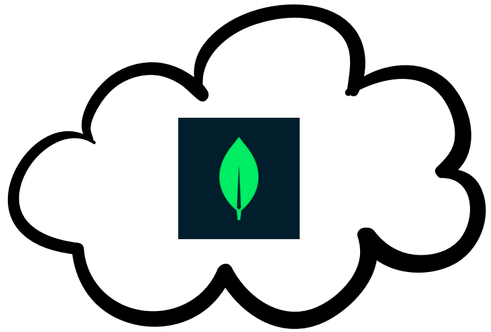
```
def mongoInsert(dataframe, epoch_id):
    global curr_invoice,item_ailes
    try:
        if dataframe.count() > 0:
            str = dataframe.collect()[0].__getitem__("kafka_output").replace("\"", "").split()

            item_list = str[3].split(",")
            quantity,subtotal,total = get_invoice(item_list)
            aile_no = []
            for i in item_list:
                aile_no.append(item_ailes[i])
            cur_DT = datetime.now()
            #curr_invoice = 0
            dataframe = spark.createDataFrame([
            Row(
                InvoiceNo = curr_invoice,
                FirstName=str[0],
                LastName=str[1],
                PhoneNumber = str[2] ,
                Date = cur_DT.strftime("%d/%m/%Y"),
                Time = cur_DT.strftime("%H:%M:%S"),
                DayOfWeek = cur_DT.strftime("%A"),
                Items=quantity,
                Aisle_visited = aile_no,
                SubTotal=subtotal,
                Total=total)
            ])

            curr_invoice += 1
            print("Inserting into MongoDB")
            dataframe.show()
            dataframe.write.format("mongo").mode("append").save()
    except Exception as e:
        print(e)


df1.writeStream.foreachBatch(mongoInsert).start().awaitTermination()
```


DEMO

Data Analysis (EDA)

Main Idea

Fetch all the data from MongoDB Atlas to Spark and gain insights



FETCHING DATA TO SPARK FROM MONGODB

Read data to spark from Mongo

```
var df = spark.read.mongo(ReadConfig(Map("uri" ->
"mongodb+srv://admin:admin@cluster0.75ml3wq.mongodb.net/?retryWrites=true&w=majority", "database"
-> "AISupermarket", "collection" -> "Invoice")))
```

DATA FETCHED

Aisle_visited	Date	DayOfWeek	FirstName	InvoiceNo	Items	LastName	PhoneNumber	SubTotal	Time	Total
[Furniture, Utens...	2018-01-01	Monday	Mary	1	{null, null, null...	Siegel	6880863634	{null, null, null...	01:40:00	5099.98
[Furniture, Kitch...	2018-01-01	Monday	Gail	2	{null, null, null...	Bilbo	6246205185	{null, null, null...	21:59:00	26349.97
[Kitchen]	2018-01-01	Monday	William	3	{null, null, null...	Lyle	8106543257	{null, null, null...	01:48:00	22299.98
[Furniture, Stati...	2018-01-01	Monday	James	4	{null, null, 1, n...	Anthony	8724845208	{null, null, 3, n...	14:27:00	10042.96
[Kitchen, Furnitu...	2018-01-01	Monday	Maureen	5	{1, 1, null, 1, 1...	Tillman	7616416146	{2, 999.99, null,...	00:00:00	139281.76
[Clothing, Grocer...	2018-01-01	Monday	Jason	6	{2, null, null, n...	Benton	8804388364	{4, null, null, n...	00:45:00	30603.960000000003
[Furniture, Groce...	2018-01-01	Monday	Ray	7	{null, null, null...	White	7646850741	{null, null, null...	01:13:00	249.99
[Furniture, Kitch...	2018-01-01	Monday	Greg	8	{null, null, null...	Gil	6086142115	{null, null, null...	16:53:00	13889.93
[Stationary, Furn...	2018-01-01	Monday	Bradley	9	{null, null, null...	Bonner	8853384685	{null, null, null...	03:46:00	6804.919999999998
[Stationary, Clot...	2018-01-01	Monday	John	10	{1, null, 1, null...	France	8572207505	{2, null, 3, null...	11:18:00	84454.92000000001
[Clothing, Statio...	2018-01-01	Monday	Elizabeth	11	{null, null, null...	Phillips	7173101252	{null, null, null...	00:11:00	2599.9799999999996
[Stationary, Kitc...	2018-01-01	Monday	Emory	12	{null, null, null...	Stewart	7852150541	{null, null, null...	06:56:00	23884.949999999997
[Furniture, Kitch...	2018-01-01	Monday	Concepcion	13	{null, null, null...	Alvarez	6236447168	{null, null, null...	15:39:00	12999.99
[Stationary, Kitc...	2018-01-01	Monday	Jacqueline	14	{null, null, null...	Diaz	7388244060	{null, null, null...	19:37:00	28919.930000000004
[Grocery, Utensil...	2018-01-01	Monday	Malissa	15	{1, null, null, n...	Goulet	6413018736	{2, null, null, n...	20:00:00	9901.949999999999
[Utensils, Kitche...	2018-01-01	Monday	Linda	16	{null, null, null...	Wood	8805374002	{null, null, null...	23:39:00	19049.930000000008
[Grocery, Kitchen]	2018-01-01	Monday	Michael	17	{null, null, null...	Spells	7070428676	{null, null, null...	15:49:00	6539.99

TOP 10 CUSTOMERS WITH MOST VISITS TO THE STORE

df.groupBy("FirstName", "LastName", "PhoneNumber").count().sort(col("count").desc).show(10)

```
+---------+---------+-----------+-----+
|FirstName| LastName|PhoneNumber|count|
+---------+---------+-----------+-----+
|     Lori|  Ludwick| 6467780662|  129|
|Katherine|   Taylor| 7637824734|  120|
|    Melba|   Baskin| 6057336700|  119|
|      Ray|    White| 7646850741|  119|
|    Doris|      Day| 7415226222|  119|
|   Jerome|    Wargo| 8246518334|  118|
|     Lisa|  Murdock| 7332064376|  118|
|   Janice|  Hartman| 6322773144|  118|
|   Arthur|    Hogan| 8226615704|  118|
|    Peter|Delatorre| 8134546324|  118|
+---------+---------+-----------+-----+
```

TOTAL SALES ON A GIVEN DATE

```
var sales_in_day = df.groupBy("Date").count()
sales_in_day.filter(col("Date") === "2020-06-23").show()
```

```
+----------+-----+
|      Date|count|
+----------+-----+
|2020-06-23|   54|
+----------+-----+
```

MOST BUSY DAYS

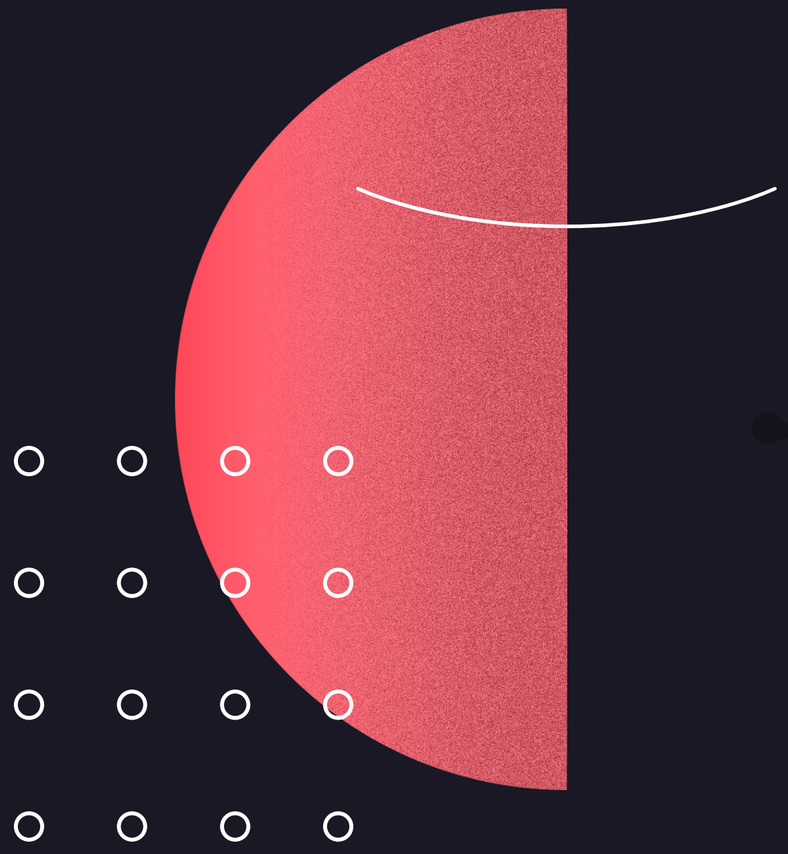
sales_in_day.sort(col("count").desc).show(5)

```
+----------+-----+
|      Date|count|
+----------+-----+
|2019-03-09|  117|
|2022-10-07|  115|
|2021-12-12|  114|
|2020-10-08|  109|
|2020-10-03|  106|
+----------+-----+
```

HOW MANY CUSTOMERS VISITED IN A MONTH?

```
df.groupBy(year(col("Date")).as("year") ,month(col("Date")).as("month") ).count().sort(col("year"), col("month")).show()
```

```
+----+-----+-----+
|year|month|count|
+----+-----+-----+
|2018|    1| 1604|
|2018|    2| 1480|
|2018|    3| 1549|
|2018|    4| 1350|
|2018|    5| 1602|
|2018|    6| 1691|
|2018|    7| 1601|
|2018|    8| 1635|
|2018|    9| 1308|
|2018|   10| 1475|
|2018|   11| 1506|
|2018|   12| 1601|
|2019|    1| 1464|
|2019|    2| 1345|
|2019|    3| 1784|
|2019|    4| 1642|
|2019|    5| 1656|
|2019|    6| 1618|
```

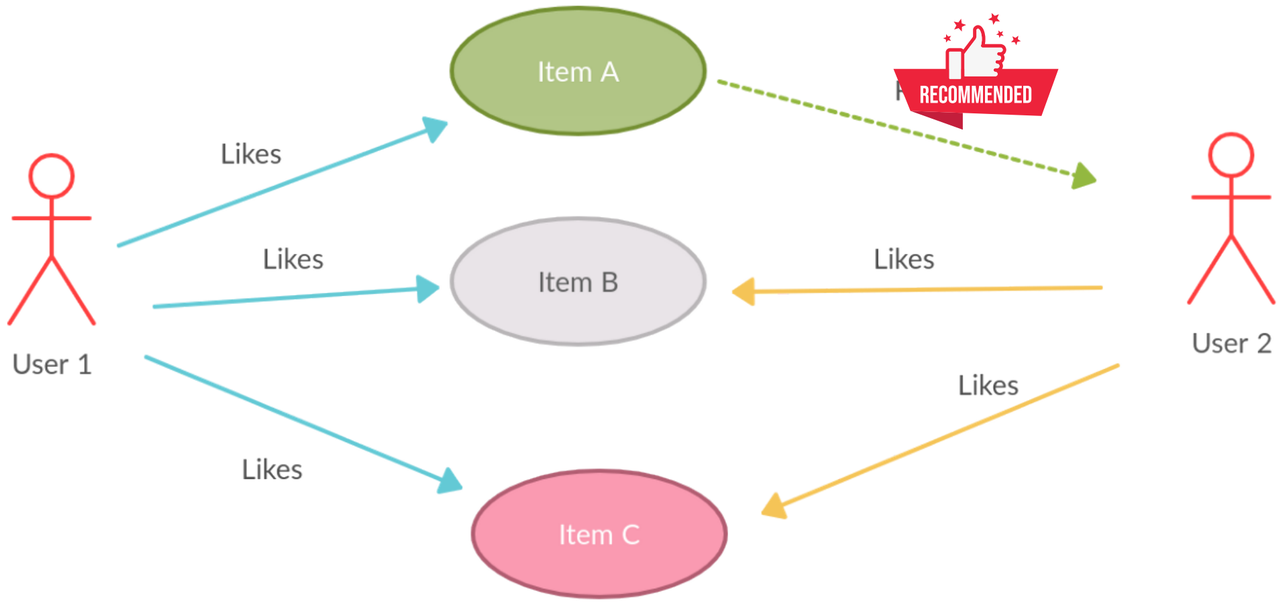


Recommendation System

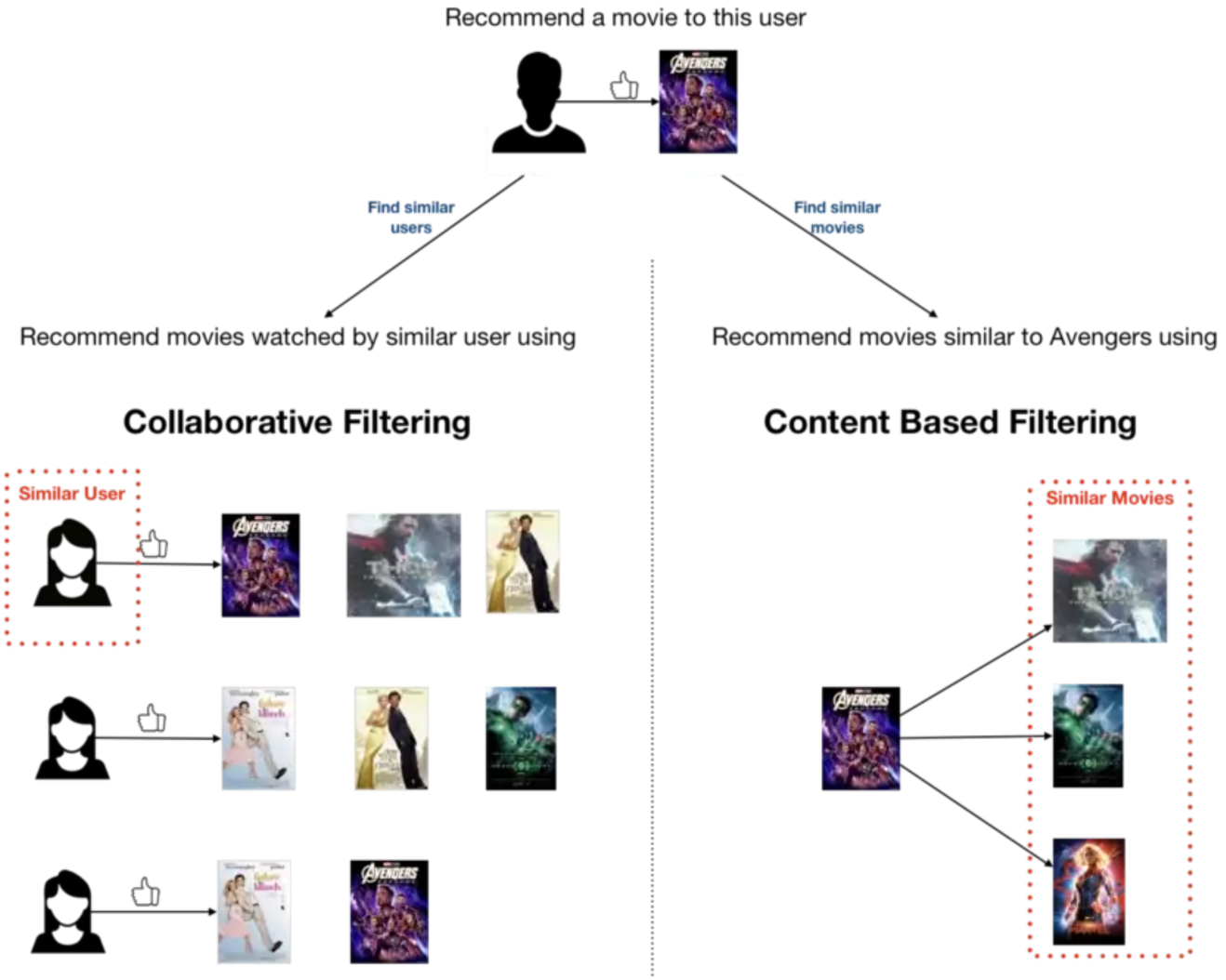
Data pre-processing
ALS
Predections

Main Idea

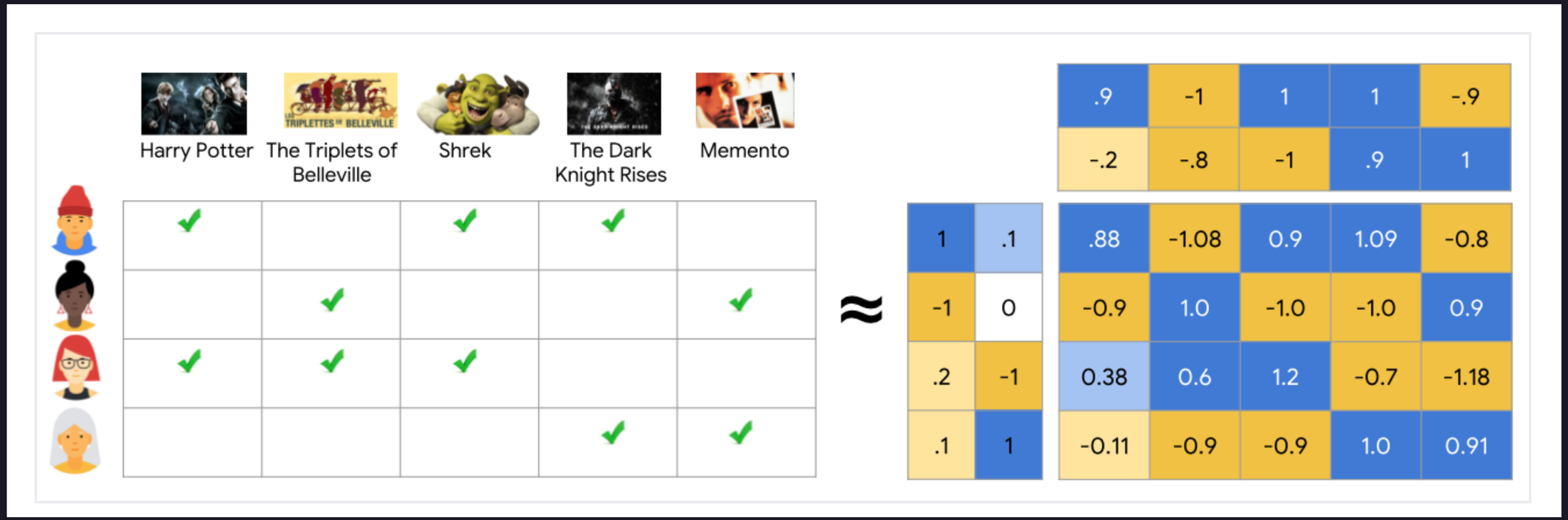
Train a Recommendation system (ALS)
to give product recommendations to users.



RECOMMENDATION SYSTEM



ALTERNATING LEAST SQUARES (ALS)



PROCESSING DATA

DATA FETCHED

Aisle_visited	Date	DayOfWeek	FirstName	InvoiceNo	Items	LastName	PhoneNumber	SubTotal	Time	Total
[Furniture, Utens...	2018-01-01	Monday	Mary	1	{null, null, null...	Siegel	6880863634	{null, null, null...	01:40:00	5099.98
[Furniture, Kitch...	2018-01-01	Monday	Gail	2	{null, null, null...	Bilbo	6246205185	{null, null, null...	21:59:00	26349.97
[Kitchen]	2018-01-01	Monday	William	3	{null, null, null...	Lyle	8106543257	{null, null, null...	01:48:00	22299.98
[Furniture, Stati...	2018-01-01	Monday	James	4	{null, null, 1, n...	Anthony	8724845208	{null, null, 3, n...	14:27:00	10042.96
[Kitchen, Furnitu...	2018-01-01	Monday	Maureen	5	{1, 1, null, 1, 1...	Tillman	7616416146	{2, 999.99, null,...	00:00:00	139281.76
[Clothing, Grocer...	2018-01-01	Monday	Jason	6	{2, null, null, n...	Benton	8804388364	{4, null, null, n...	00:45:00	30603.960000000003
[Furniture, Groce...	2018-01-01	Monday	Ray	7	{null, null, null...	White	7646850741	{null, null, null...	01:13:00	249.99
[Furniture, Kitch...	2018-01-01	Monday	Greg	8	{null, null, null...	Gil	6086142115	{null, null, null...	16:53:00	13889.93
[Stationary, Furn...	2018-01-01	Monday	Bradley	9	{null, null, null...	Bonner	8853384685	{null, null, null...	03:46:00	6804.919999999998
[Stationary, Clot...	2018-01-01	Monday	John	10	{1, null, 1, null...	France	8572207505	{2, null, 3, null...	11:18:00	84454.92000000001
[Clothing, Statio...	2018-01-01	Monday	Elizabeth	11	{null, null, null...	Phillips	7173101252	{null, null, null...	00:11:00	2599.9799999999996
[Stationary, Kitc...	2018-01-01	Monday	Emory	12	{null, null, null...	Stewart	7852150541	{null, null, null...	06:56:00	23884.949999999997
[Furniture, Kitch...	2018-01-01	Monday	Concepcion	13	{null, null, null...	Alvarez	6236447168	{null, null, null...	15:39:00	12999.99
[Stationary, Kitc...	2018-01-01	Monday	Jacqueline	14	{null, null, null...	Diaz	7388244060	{null, null, null...	19:37:00	28919.930000000004
[Grocery, Utensil...	2018-01-01	Monday	Malissa	15	{1, null, null, n...	Goulet	6413018736	{2, null, null, n...	20:00:00	9901.949999999999
[Utensils, Kitche...	2018-01-01	Monday	Linda	16	{null, null, null...	Wood	8805374002	{null, null, null...	23:39:00	19049.930000000008
[Grocery, Kitchen]	2018-01-01	Monday	Michael	17	{null, null, null...	Spells	7070428676	{null, null, null...	15:49:00	6539.99

EXPECTED DATA FORMAT

User id	Item id	Rating
1	a	1
1	b	2
2	a	3
2	c	2
3	c	5

Data

items

```
item_df = item_df.groupBy(col("PhoneNumber" )).sum()
```

```
+-----------+-----+--------+------+---+-------+----+------+----+--------+----+------+---------+-----+-----+-----+---+----+----------+-----+----+----+---------+-------+---+-------
-+-----+------+---------+------+-----+------+----+-----+-----+----------+----------+------+--------+--------+----+----+-----+--------+---------+-------+----------+---+---------+
----+------+
|PhoneNumber|apple|backpack|banana|bed|blender|book|bottle|bowl|broccoli|cake|carrot|cellphone|chair|clock|couch|cup|desk|diningtable|donut|door|fork|hairdrier|handbag|hat|keyboar
d|knife|laptop|microwave|mirror|mouse|orange|oven|pizza|plate|pottedplant|refrigerator|remote|sandwich|scissors|shoe|sink|spoon|suitcase|teddybear|toaster|toothbrush| tv|umbrella|
vase|window|
+-----------+-----+--------+------+---+-------+----+------+----+--------+----+------+---------+-----+-----+-----+---+----+----------+-----+----+----+---------+-------+---+-------
-+-----+------+---------+------+-----+------+----+-----+-----+----------+----------+------+--------+--------+----+----+-----+--------+---------+-------+----------+---+---------+
----+------+
| 6741123020|    7|      15|    11|  7|     14|   7|     7|  11|      11|  10|    12|       10|    7|   12|   12|  8|  12|         8|   13|  10|   3|       18|     14| 10|      1
4|    5|    16|        7|     9|   14|     7|   8|    9|    7|         8|          15|    14|       9|       9|   8|   8|    3|      11|        6|     13|         9|  5|      15|
7|     7|
| 6848025004|    5|       8|     4|  7|     10|   8|     9|   6|       5|  11|    15|       12|    7|   10|   13| 11|   9|        13|   11|   7|   7|       13|      8|  8|
9|   10|     9|        8|     8|    9|     8|   9|   12|    7|        14|           9|    14|       5|      10|  11|  12|    8|       9|        7|      9|        10|  9|      13|
10|     8|
| 6758586356|   11|       6|     5| 12|      9|   6|    11|   9|      15|   7|     8|       11|   11|   16|   10|  9|  15|        10|   14|   8|  13|        5|     18|  3|      1
1|   23|    16|        8|     5|   19|    15|  10|   15|   10|        15|           9|    16|      11|      10|   7|  12|   17|      13|       12|     14|        17|  7|      16|
7|     9|
| 6267870466|    9|      13|     3|  6|     10|   5|     6|   9|       7|   9|     6|       11|    3|    9|    7|  8|   8|         7|    6|   7|  10|        5|      7|  8|
7|    7|     7|        6|     9|    3|    10|  14|    6|    6|         8|          15|     4|       9|       8|  15|  15|   12|       7|        7|     12|         9|  4|      18|
4|    12|
```

Stacking DF

PhoneNumber	item_name	Rating
6741123020	apple	7
6741123020	backpack	15
6741123020	banana	11
6741123020	bed	7
6741123020	blender	14
6741123020	book	7
6741123020	bottle	7
6741123020	bowl	11
6741123020	broccoli	11
6741123020	cake	10
6741123020	carrot	12
6741123020	cellphone	10
6741123020	chair	7
6741123020	clock	12
6741123020	couch	12
6741123020	cup	8
6741123020	desk	12
6741123020	diningtable	8

StringIndexer

id	category
0	a
1	b
2	c
3	a
4	a
5	c

id	category	categoryIndex
0	a	0.0
1	b	2.0
2	c	1.0
3	a	0.0
4	a	0.0
5	c	1.0

String indexer

```
new StringIndexer()
.setInputCol("category")
.setOutputCol("category_index")
.fit(df)
.transform(df)
```

indexed df

PhoneNumber	item_name	Rating	user_id	item_id
6741123020	apple	7	300.0	0.0
6741123020	backpack	15	300.0	1.0
6741123020	banana	11	300.0	2.0
6741123020	bed	7	300.0	3.0
6741123020	blender	14	300.0	4.0
6741123020	book	7	300.0	5.0
6741123020	bottle	7	300.0	6.0
6741123020	bowl	11	300.0	7.0
6741123020	broccoli	11	300.0	8.0
6741123020	cake	10	300.0	9.0
6741123020	carrot	12	300.0	10.0
6741123020	cellphone	10	300.0	11.0
6741123020	chair	7	300.0	12.0
6741123020	clock	12	300.0	13.0
6741123020	couch	12	300.0	14.0
6741123020	cup	8	300.0	15.0
6741123020	desk	12	300.0	16.0
6741123020	diningtable	8	300.0	17.0

BUILDING MODEL

Building model

```
val Array(train, test) = data.randomSplit(Array(0.8,0.2))

val als = new
ALS().setMaxIter(10).setRegParam(0.01).setUserCol("user_id").setItemCol("Item_id").setRatingCol("
Rating")

val model = als.fit(train)
model.setColdStartStrategy("drop")
val predections = model.transform(test)

val evaluator = new
RegressionEvaluator().setMetricName("rmse").setLabelCol("Rating").setPredictionCol("prediction")


val rmse = evaluator.evaluate(predections)
println(rmse)
```

making predictions

```
var Recs = model.recommendForAllUsers(5)
```

FirstName	LastName	PhoneNumber	recommended_item_1	recommended_item_2	recommended_item_3	recommended_item_4	recommended_item_5
James	Ellis	6052745548	clock	handbag	shoe	apple	umbrella
James	Hewitt	7234864018	door	blender	mirror	diningtable	desk
Carol	Guarino	6103587201	handbag	toaster	blender	shoe	oven
Janie	Houle	7526140165	mouse	laptop	remote	keyboard	bowl
Jean	Temple	6212566532	couch	desk	mirror	teddybear	backpack
Martha	Minton	8260114027	suitcase	bowl	vase	blender	toaster
Martha	Moody	6275716763	bowl	fork	donut	backpack	desk
Laura	Sylvester	6168733257	handbag	toaster	toothbrush	hairdrier	remote
James	Groce	6754277887	clock	scissors	vase	hairdrier	cake
Hugh	Harger	6270282118	oven	refrigerator	cellphone	blender	toaster
Randy	Whitten	7140160374	scissors	cup	teddybear	diningtable	plate
Paul	Williams	6236288812	plate	mouse	cellphone	remote	fork
Lillian	Berardi	7423434165	teddybear	vase	mirror	hairdrier	diningtable
Joan	Scott	7618161650	cake	broccoli	pizza	blender	mouse
Gregory	Hayden	7037886464	cake	orange	carrot	donut	blender
Karla	Flowers	7363383165	donut	desk	pottedplant	door	carrot
Jessie	Dejesus	7401130657	desk	handbag	cellphone	couch	shoe
Patricia	Perreira	8146131410	oven	apple	handbag	donut	carrot

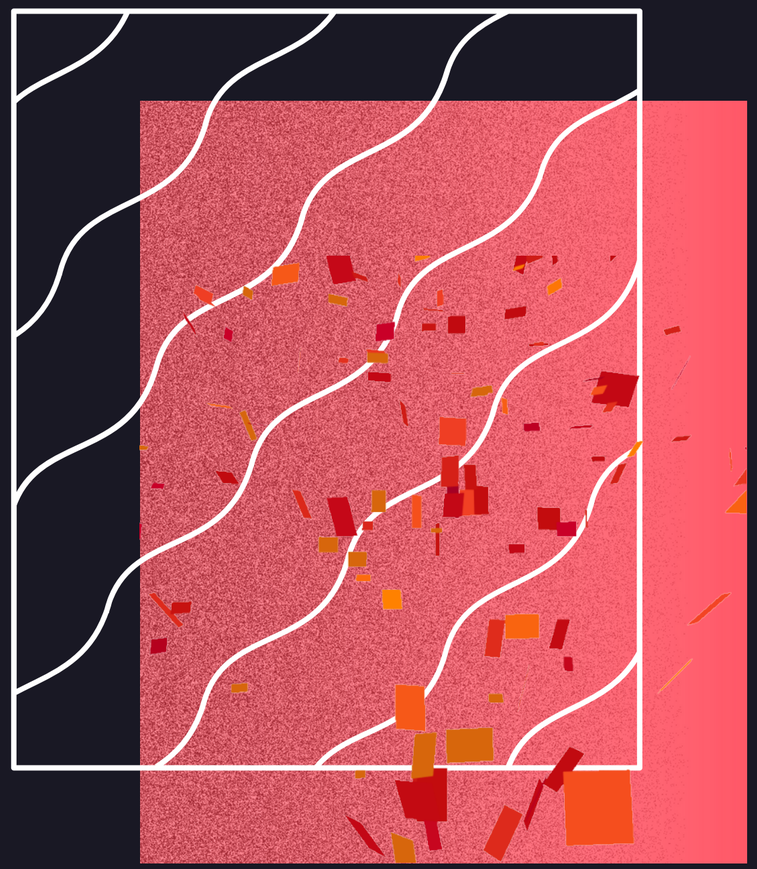
Pushing data to mongoDB

making predictions

```
val connect_string = "mongodb+srv://admin:admin@cluster0.75ml3wq.mongodb.net/test"
val database = "AISupermarket"
val collection = "Recommendation"

final_recommendations.write.format("mongo").option("spark.mongodb.output.uri",
connect_string).option("database",database).option("collection",collection).mode("append").save()
```

Recommendations

DEMO



Thank You