UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

M.Sc. Computer Science with Spl. in Data Science – Semester III

Data Engineering

JOURNAL

2022-2023

Seat No. 30283

मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)

UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the work entered in this journal was done in the University Department of Computer Science laboratory by Mr./~~Ms.~~ **Sumon Singh** Seat No. **30283** for the course of M.Sc. Computer Science with Spl. in Data Science - Semester III (CBCS) (Revised) during the academic year 2022-2023 in a satisfactory manner.

_____ _____

Subject In-charge **Head of Department**

_____

External Examiner

Index

Practical : 1 (CouchDB)

Tasks :

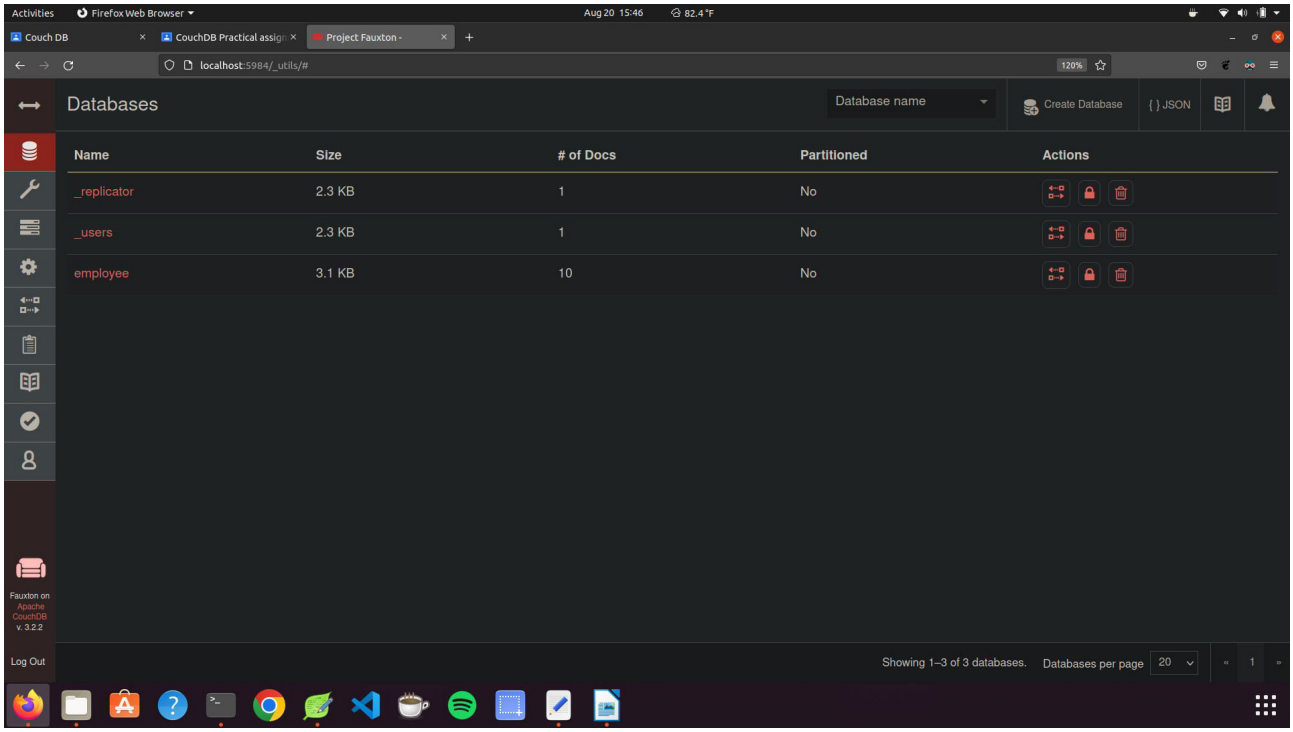
Perform the following operations in Couchdb:

- **Open Futon Web interface**
- **Create a database called employee using curl utility**
- **Show the databases.**
- **Insert the following documents in the database**
- **Select the employees who belong to the city Delhi.**
- **Select the name , designation and salary of the employees where the salary is between 200000 and 400000.**
- **Select the employee whose name is "Celina" and belong to "Pune" city.**
- **Update the city "Maharashtra" to "Nagpur"**
- **Delete the record of Celina who is a JavaScript Developer**

1. Open Futon Web interface :

Insruction :

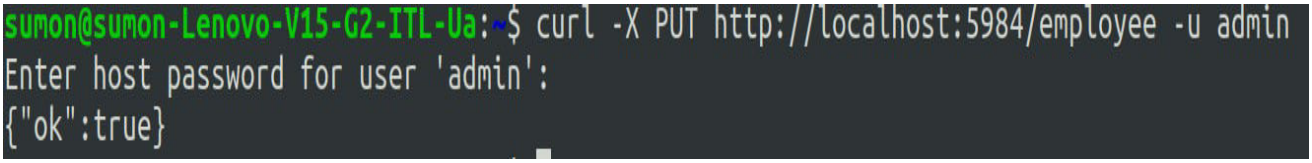
Go to http://localhost:5984/_utils/



2. Create a database called employee using curl utility

Instruction :

curl -X PUT http://localhost:5984/employee -u admin



3. Show the databases

Instruction :

curl -X GET http://localhost:5984/_all_dbs -u admin

```
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X GET http://localhost:5984/_all_dbs -u admin
Enter host password for user 'admin':
["_replicator","_users","employee"]
```

4. Insert the following documents in the database

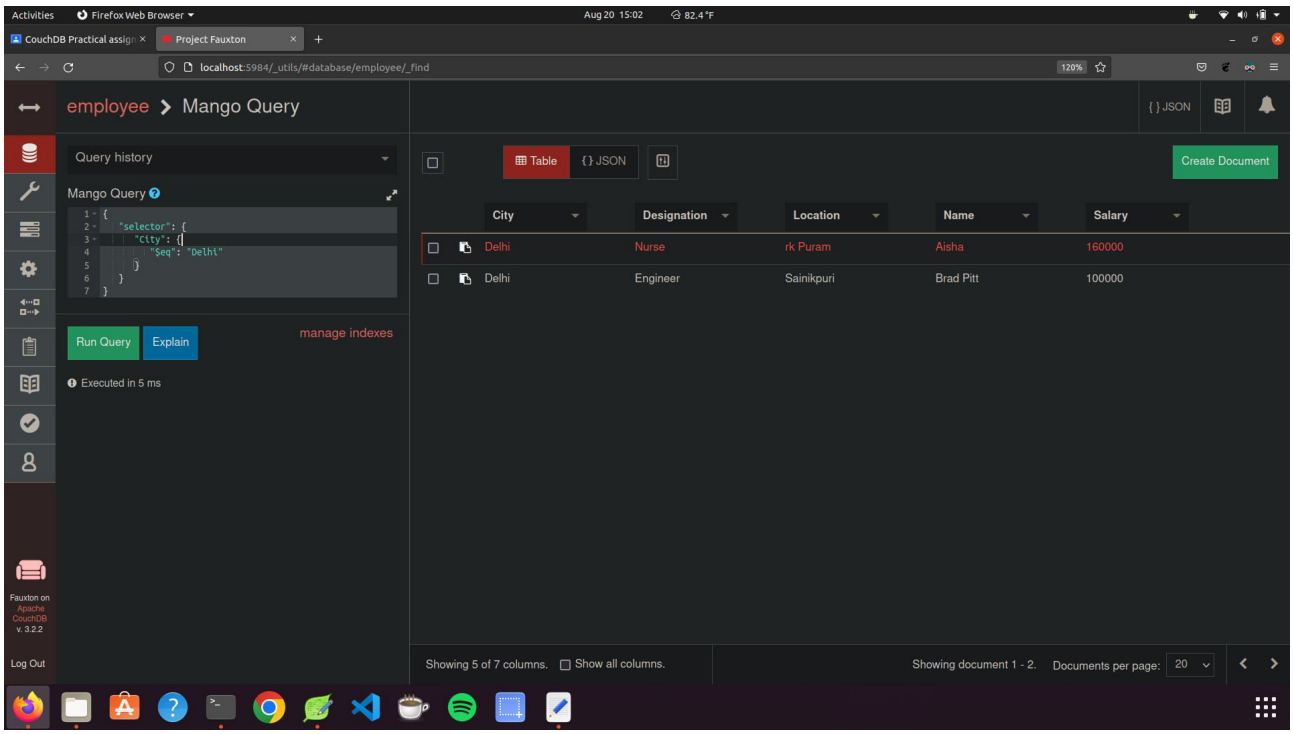
```
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/1 -d '{"Name":"Aisha","Designation":"Nurse","Sal
ary":160000,"Location":"rk Puram","City":"Delhi"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"1","rev":"1-4ffc951f6d2e29cfcd4df1ac8309ca1b"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/2 -d '{"Name":"Angelina","Designation":"CEO","Sa
lary":1000000,"Location":"JM Road","City":"Aundh Pune"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"2","rev":"1-a170707452390d06b44ca5795f719b2a"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/3 -d '{"Name":"Ashwini","Designation":"Junior En
gineer","Salary":860000,"Location":"Santa Cruz","City":"Mumbai"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"3","rev":"1-754d04456c2464d119ccbdf3fb5d7987"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/4 -d '{"Name":"Brad Henry","Designation":"Accoun
tant","Salary":370000,"Location":"Chaurasi lane","City":"Kolkata"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"4","rev":"1-7122d8c592f9d35c4c8ce44f52e995a7"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/5 -d '{"Name":"Brad Pitt","Designation":"Enginee
r","Salary":100000,"Location":"Sainikpuri","City":"Delhi"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"5","rev":"1-91ec8c7b371499d940f42d83b3aee5e1"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/6 -d '{"Name":"Celina","Designation":"Javascript
 Developer","Salary":430060,"Location":"Crr Lake Side Ville","City":"Tellapur"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"6","rev":"1-81e4e7d2f36535c7e284c1189b3d5593"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/7 -d '{"Name":"Celina","Designation":"Senior Dev
eloper","Salary":200000,"Location":"23, Chadni Chowk","City":"Pune"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"7","rev":"1-e9282e4186e360326108610f8cb8e707"}
```

```
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/8 -d '{"Name":"Deepa","Designation":"Team Leader
","Salary":200500,"Location":"Annanagar","City":"Chennai"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"8","rev":"1-f1c5dcbb3b7b2676d1d61b755596d6c8"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/9 -d '{"Name":"Gaurav","Designation":"Teacher","
Salary":100750,"Location":"Esquare,JM Road","City":"Pune"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"9","rev":"1-269a2a5fc9817d1ad741aeb44a038fdb"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/10 -d '{"Name":"Glenny","Designation":"Adminstra
tor","Salary":200500,"Location":"Nagpur","City":"Maharashtra"}' -u admin
Enter host password for user 'admin':
{"ok":true,"id":"10","rev":"1-568ab0d9b6280c337c1a30e776039bdb"}
```

5. Select the employees who belong to the city Delhi

Instruction :

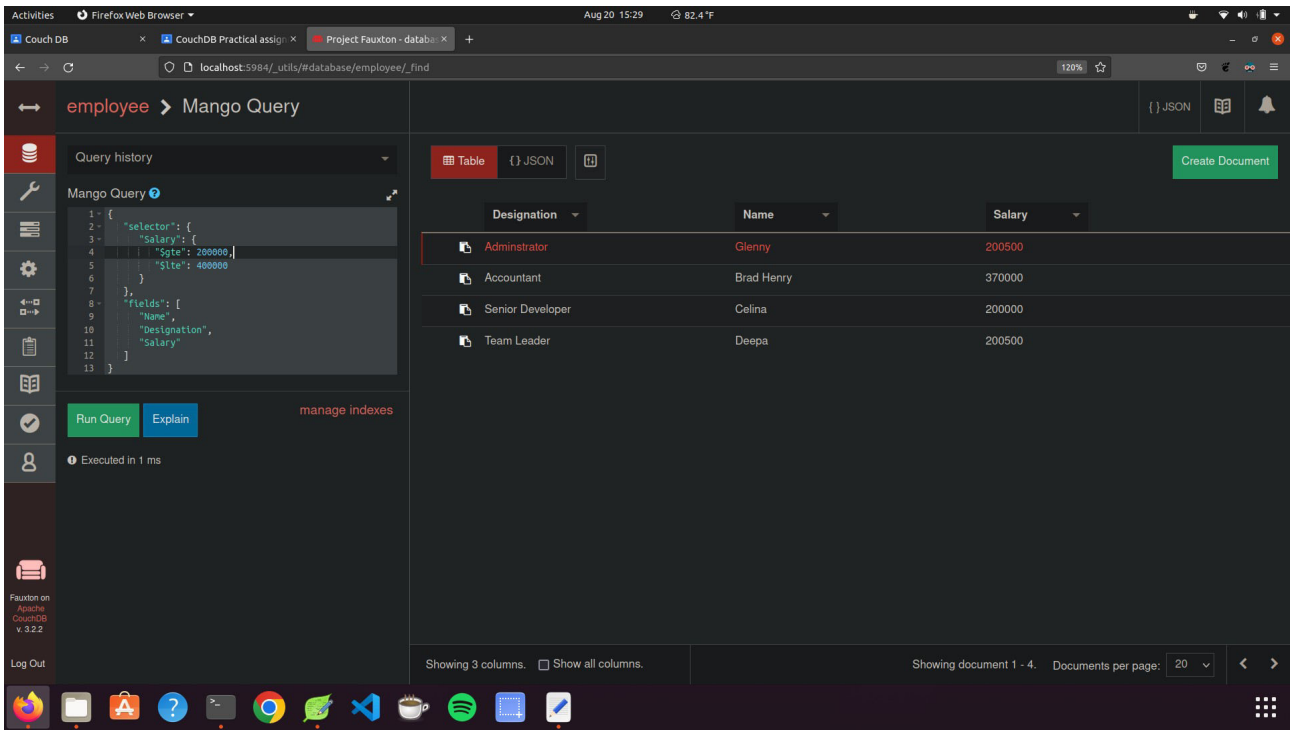
```
{
  "selector": {
    "City": {
      "$eq": "Delhi"
    }
  }
}
```



6. Select the name , designation and salary of the employees where the salary is between 200000 and 400000

Instruction :

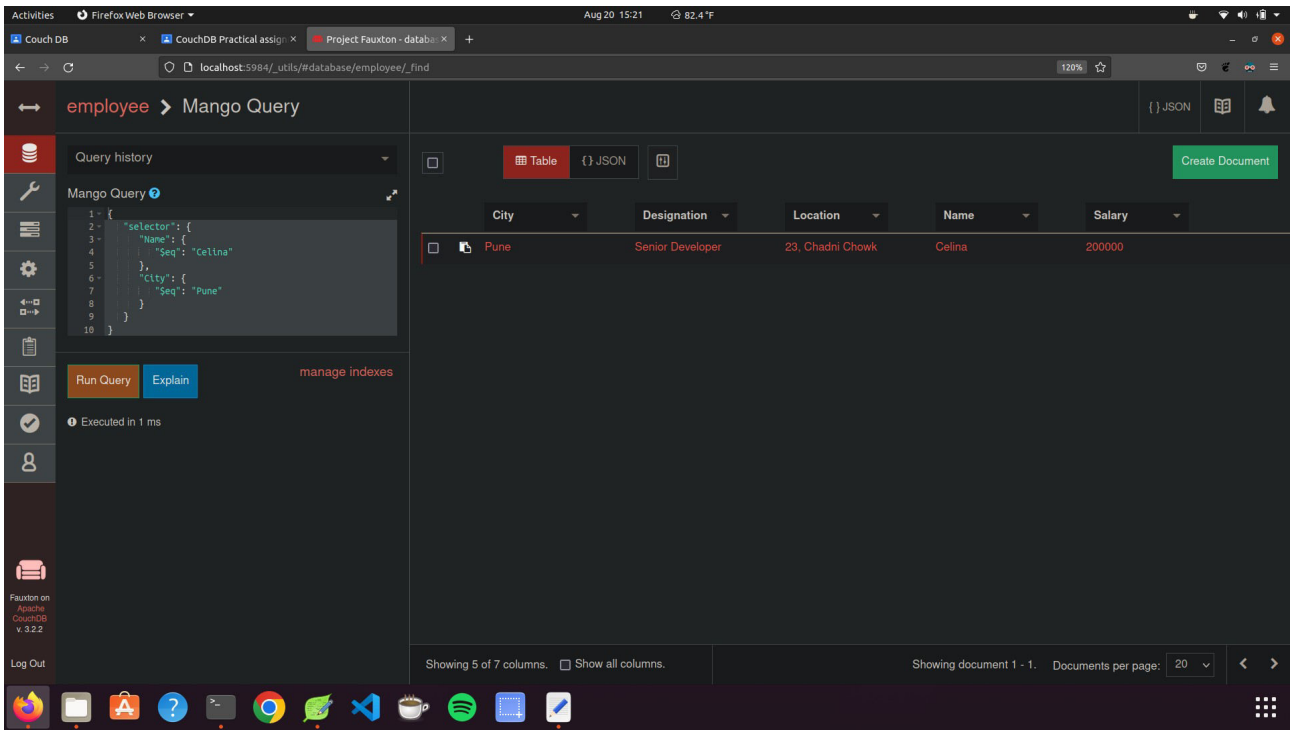
```
{
  "selector": {
    "Salary": {
      "$gte": 200000,
      "$lte": 400000
    }
  },
  "fields": [
    "Name",
    "Designation",
    "Salary"
  ]
}
```



7. Select the employee whose name is "Celina" and belong to "Pune" city

Instruction :

```
{
  "selector": {
    "Name": {
      "$eq": "Celina"
    },
    "City": {
      "$eq": "Pune"
    }
  }
}
```



8. Update the city "Maharashtra" to "Nagpur"

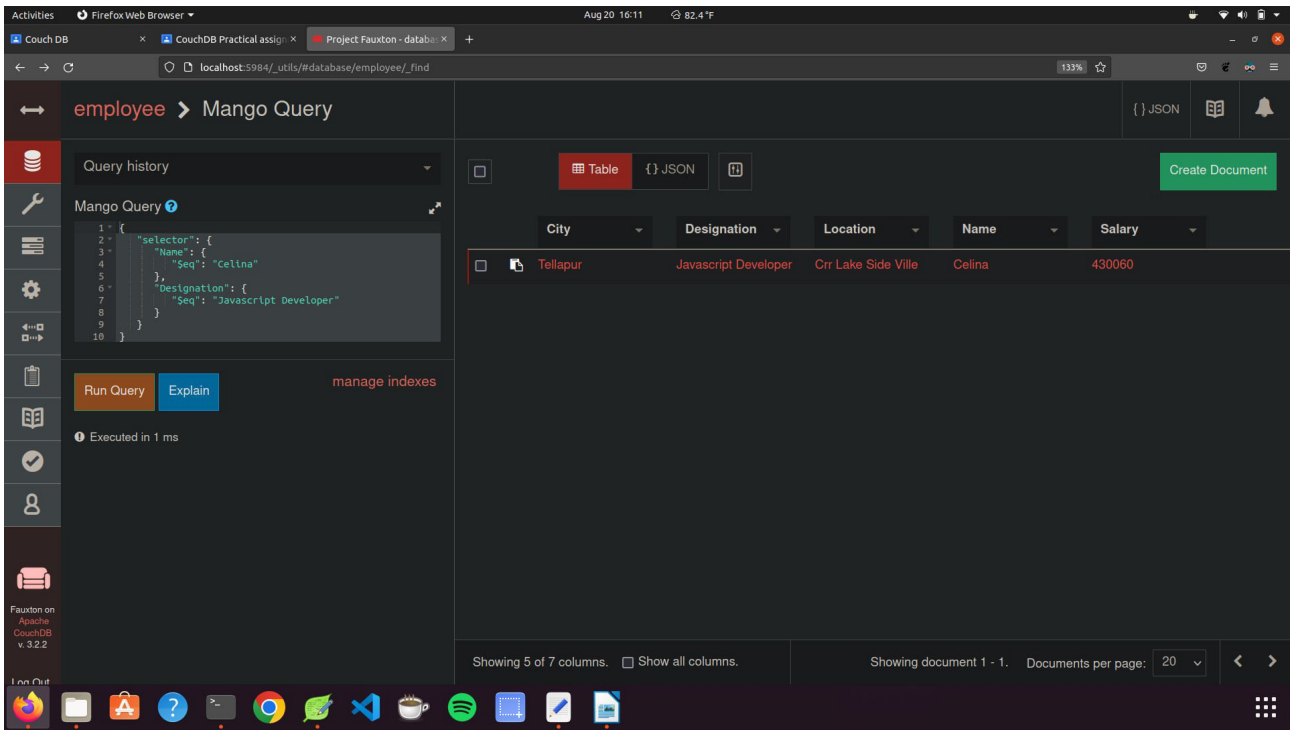
Instruction :

curl -X PUT http://localhost:5984/employee/10 -u admin -d '{"_rev":"4-574b4e4564a5035e392fe4ba8543b8d0","City":"Nagpur"}'

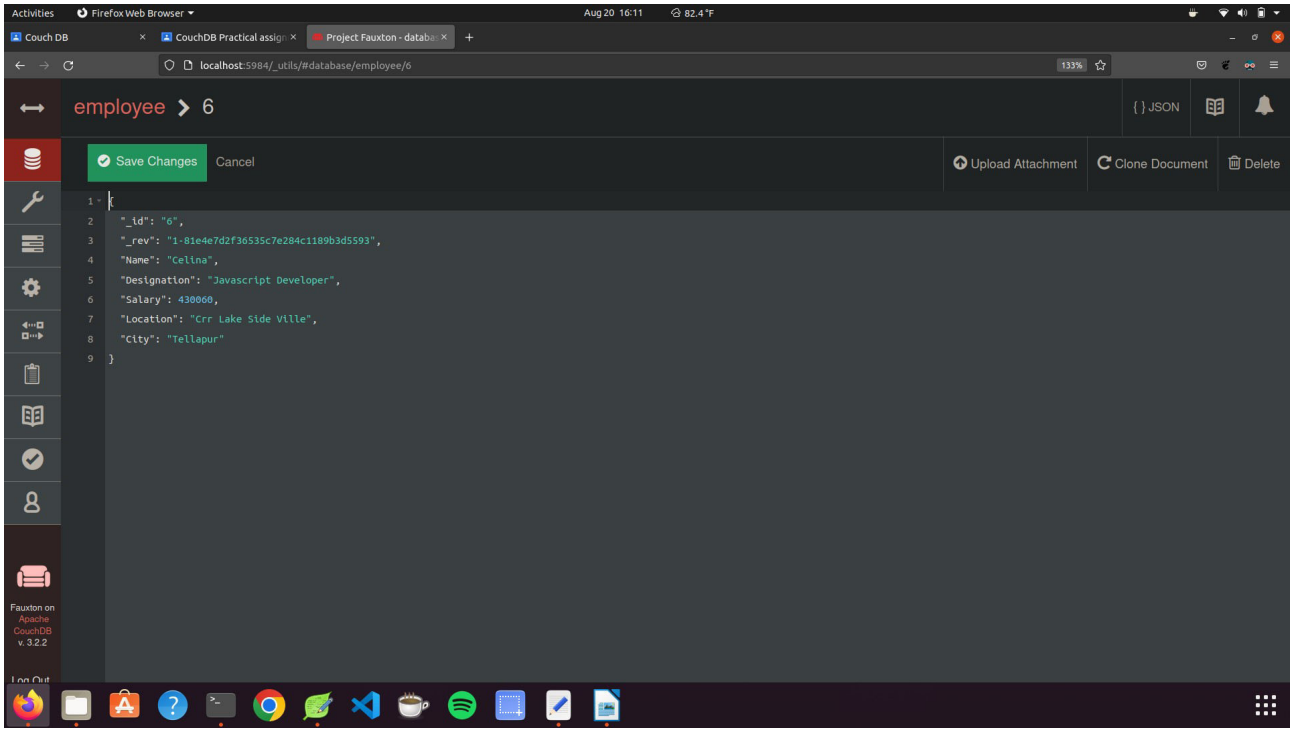
9. Delete the record of Celina who is a JavaScript Developer

Step 1: Find the required document

```
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X GET http://localhost:5984/employee/10 -u admin
Enter host password for user 'admin':
{"_id":"10","_rev":"4-574b4e4564a5035e392fe4ba8543b8d0","Name":"Glenny","Designation":"Adminstrator","Salary":200500,"Location"
:"Nagpur","City":"Maharashtra"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X PUT http://localhost:5984/employee/10 -u admin -d '{"_rev":"4-574b4e4564a5035e392fe
4ba8543b8d0","City":"Nagpur"}'
Enter host password for user 'admin':
{"ok":true,"id":"10","rev":"5-50b0cde02880a1adafc25d3f47a5cee6"}
```



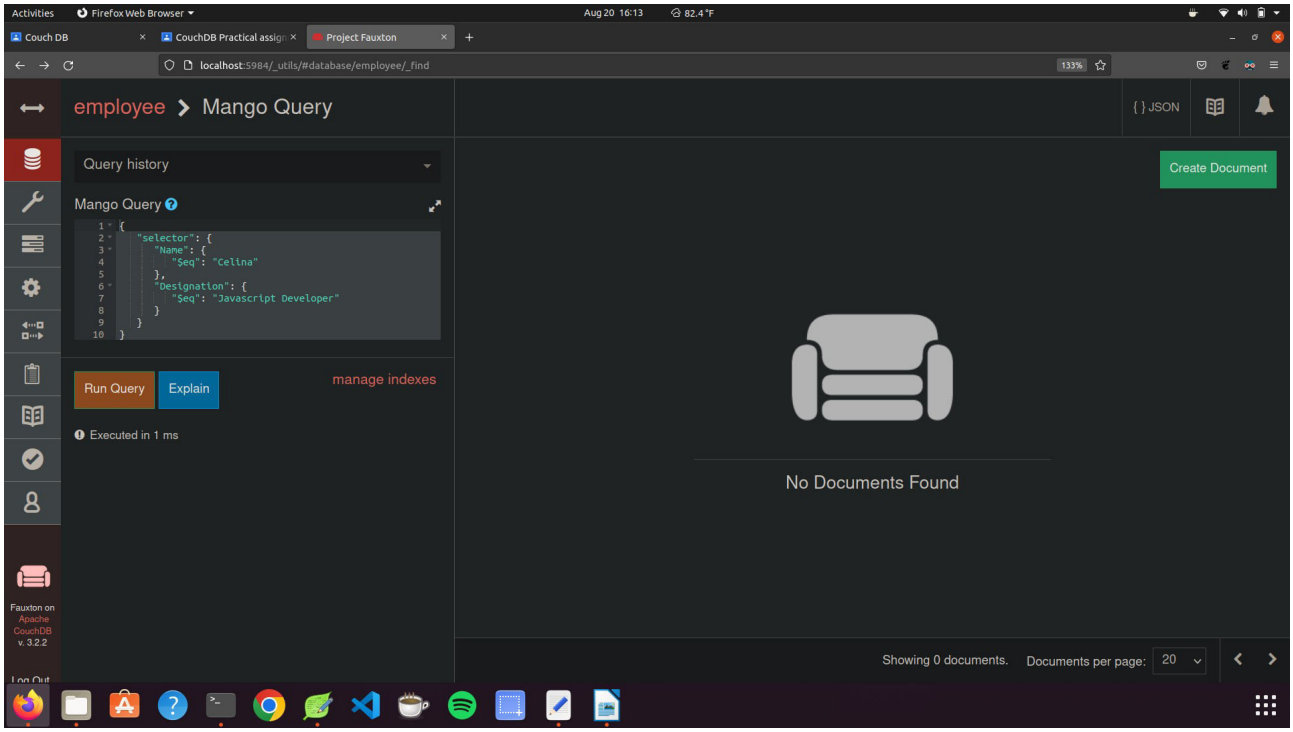
Step 2: Get the document's id and rev



Step 3: Delete the document

```
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X GET http://localhost:5984/employee/6 -u admin
Enter host password for user 'admin':
{"_id":"6","_rev":"1-81e4e7d2f36535c7e284c1189b3d5593","Name":"Celina","Designation":"Javascript Developer","Salary":430060,"Lo
cation":"Crr Lake Side Ville","City":"Tellapur"}
sumon@sumon-Lenovo-V15-G2-ITL-Ua:~$ curl -X DELETE http://localhost:5984/employee/6?rev=1-81e4e7d2f36535c7e284c1189b3d5593 -u a
dmin
Enter host password for user 'admin':
{"ok":true,"id":"6","rev":"2-abaeba00f4eba9a6b9fb03970e2f3e28"}
```

Step 4: Check if it's deleted or not



Practical : 2 (MongoDB)

Tasks :

Perform the following operations in MongoDB command line or GUI. Take screenshots. Write the question followed by the query and output as screenshot. Upload pdf with your name and roll number clearly written. Create the collection named pizza followed by your rollno.

- **Create the database restaurant in MongoDB**
- **Create a collection called pizza_rollno**
- **Insert documents**
- **Display all the records of the Pizza collection**
- **Print the pizza collection where size of the pizza is small**
- **Print the columns "name", "Size" and "price" of the pizza**
- **Print the records of the pizza where the price is less than 20.**
- **Print the name and size of the pizza where the pizza is a cheese pizza.**
- **Use map reduce operation to group the pizza by name and print the total quantity.**
- **Use map reduce operation to count the number of pizzas by name.**
- **Use map reduce to group the pizzas by name and print the sum of twice(double) the price of the pizzas.**
- **Print the information about pizzas where price is between 20 and 40.**
- **use aggregation pipeline to print the total quantity ordered according to name.**
- **use aggregation pipeline to print the name and price of the pepperoni pizza.**
- **Sort the pizzas by descending price. use aggregation pipeline.**
- **Print the sum of total quantity ordered between 01-01-2021 to 01-03-2021**
- **To print the minimum price of the pizza as per size.**
- **Print the name , size, price and discount which is calculated as , if quantity ordered is greater than 20 then 5/- discount else 2/- discount. (use aggregation pipeline)**
- **Print the pizza name, size and new_price which is calculated as per size , if size is small new_price is 10, if size is medium ,new_price is 20 and if size is large new_size is 40.**

1. Create the database restaurant in MongoDB

Query:

use restaurant;

2. Create a collection called pizza_rollno

Query:

db.createCollection('pizza_rollno');

```
test> use restaurant;
switched to db restaurant
restaurant> db.createCollection('pizza_rollno');
{ ok: 1 }
```

3. Insert documents

Query:

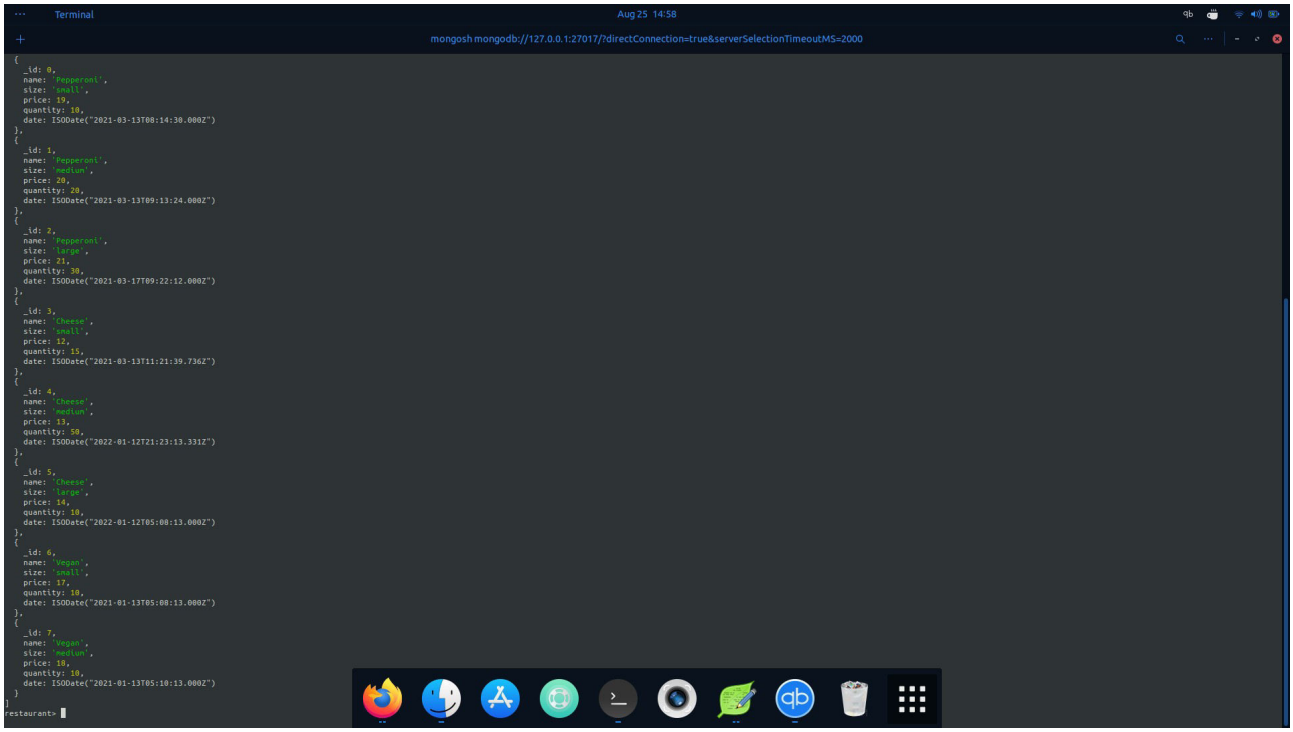
db.pizza_rollno.insertMany([{ _id: 0, name: "Pepperoni", size: "small", price: 19, quantity: 10, date: ISODate("2021-03-13T08:14:30Z") }, { _id: 1, name: "Pepperoni", size: "medium", price: 20, quantity: 20, date : ISODate("2021-03-13T09:13:24Z") }, { _id: 2, name: "Pepperoni", size: "large", price: 21, quantity: 30, date : ISODate("2021-03-17T09:22:12Z") }, { _id: 3, name: "Cheese", size: "small", price: 12, quantity: 15, date : ISODate("2021-03-13T11:21:39.736Z") }, { _id: 4, name: "Cheese", size: "medium", price: 13, quantity:50, date : ISODate("2022-01-12T21:23:13.331Z") }, { _id: 5, name: "Cheese", size: "large", price: 14, quantity: 10, date : ISODate("2022-01-12T05:08:13Z") }, { _id: 6, name: "Vegan", size: "small", price: 17, quantity: 10, date : ISODate("2021-01-13T05:08:13Z") }, { _id: 7, name: "Vegan", size: "medium", price: 18, quantity: 10, date : ISODate("2021-01-13T05:10:13Z") }]);

```
restaurant> db.pizza_rollno.insertMany([{ _id: 0, name: "Pepperoni", size: "small", price: 19, quantity: 10, date: ISODate( "2021-03-13T08:14:30Z" ) }, { _id:
 1, name: "Pepperoni", size: "medium", price: 20, quantity: 20, date : ISODate( "2021-03-13T09:13:24Z" ) }, { _id: 2, name: "Pepperoni", size: "large", price:
 21, quantity: 30, date : ISODate( "2021-03-17T09:22:12Z" ) }, { _id: 3, name: "Cheese", size: "small", price: 12, quantity: 15, date : ISODate( "2021-03-13T1
1:21:39.736Z" ) }, { _id: 4, name: "Cheese", size: "medium", price: 13, quantity:50, date : ISODate( "2022-01-12T21:23:13.331Z" ) }, { _id: 5, name: "Cheese",
 size: "large", price: 14, quantity: 10, date : ISODate( "2022-01-12T05:08:13Z" ) }, { _id: 6, name: "Vegan", size: "small", price: 17, quantity: 10, date : I
SODate( "2021-01-13T05:08:13Z" ) }, { _id: 7, name: "Vegan", size: "medium", price: 18, quantity: 10, date : ISODate( "2021-01-13T05:10:13Z" ) } ])
{
  acknowledged: true,
  insertedIds: { '0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5, '6': 6, '7': 7 }
}
```

4. Display all the records of the Pizza collection

Query:

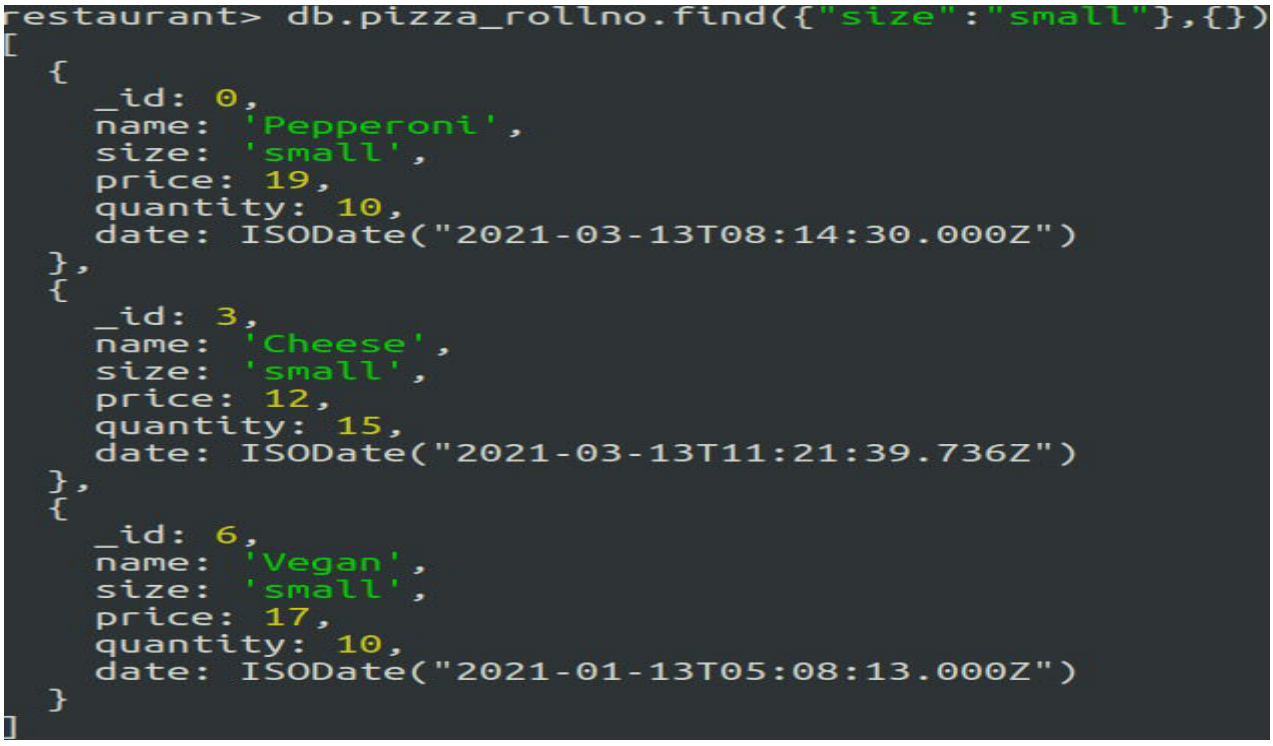
db.pizza_rollno.find({},{});



5. Print the pizza collection where size of the pizza is small

Query:

db.pizza_rollno.find({"size":"small"},{});



6. Print the columns "name", "Size" and "price" of the pizza

Query:

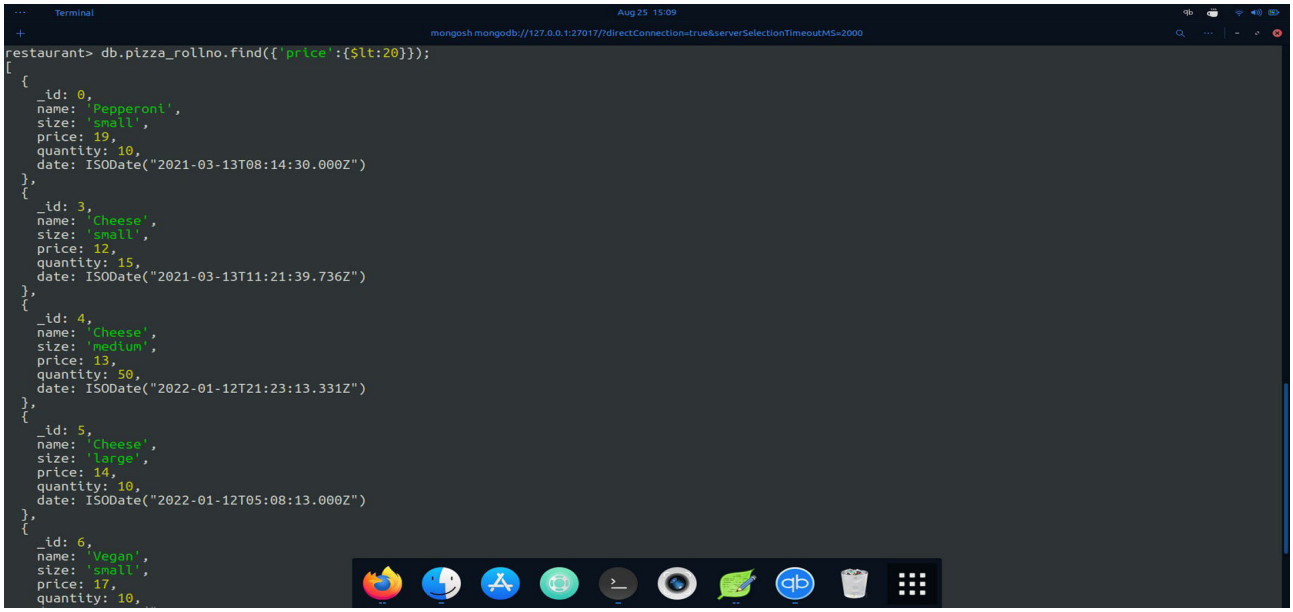
db.pizza_rollno.find({},{'name':1,'size':1,'price':1});

```
restaurant> db.pizza_rollno.find({},{'name':1,'size':1,'price':1});
[
  { _id: 0, name: 'Pepperoni', size: 'small', price: 19 },
  { _id: 1, name: 'Pepperoni', size: 'medium', price: 20 },
  { _id: 2, name: 'Pepperoni', size: 'large', price: 21 },
  { _id: 3, name: 'Cheese', size: 'small', price: 12 },
  { _id: 4, name: 'Cheese', size: 'medium', price: 13 },
  { _id: 5, name: 'Cheese', size: 'large', price: 14 },
  { _id: 6, name: 'Vegan', size: 'small', price: 17 },
  { _id: 7, name: 'Vegan', size: 'medium', price: 18 }
]
```

7. Print the records of the pizza where the price is less than 20.

Query:

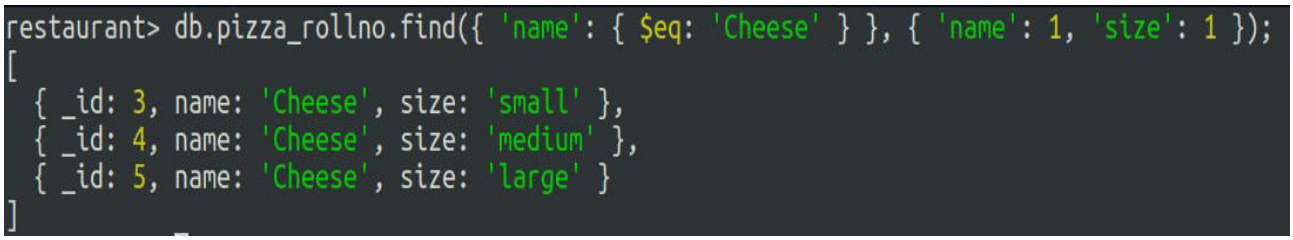
db.pizza_rollno.find({'price':{$lt:20}});



8. Print the name and size of the pizza where the pizza is a cheese pizza.

Query:

db.pizza_rollno.find({ 'name': { $eq: 'Cheese' } }, { 'name': 1, 'size': 1 });



9. Use map reduce operation to group the pizza by name and print the total quantity.

Query:

```
var mapper=function(){emit(this.name,this.quantity)};
var reducer=function(n,q){return Array.sum(q)};
db.pizza_rollno.mapReduce(mapper,reducer,{out:"result"});
db.result.find({},{});
```

```
restaurant> var mapper=function(){emit(this.name,this.quantity)};

restaurant> var reducer=function(n,q){return Array.sum(q)};

restaurant> db.pizza_rollno.mapReduce(mapper,reducer,{out:"result"});
{ result: 'result', ok: 1 }
restaurant> db.result.find({},{});
[
  { _id: 'Cheese', value: 75 },
  { _id: 'Vegan', value: 20 },
  { _id: 'Pepperoni', value: 60 }
]
```

10. Use map reduce operation to count the number of pizzas by name.

Query:

```
var mapper=function(){emit(this.name,1)};
var reducer=function(n,c){return Array.sum(c)};
db.pizza_rollno.mapReduce(mapper,reducer,{out:"pizza_count"});
db.pizza_count.find({},{});
```

```
restaurant> var mapper=function(){emit(this.name,1)};

restaurant> var reducer=function(n,c){return Array.sum(c)};

restaurant> db.pizza_rollno.mapReduce(mapper,reducer,{out:"pizza_count"});
{ result: 'pizza_count', ok: 1 }
restaurant> db.pizza_count.find({},{});
[
  { _id: 'Vegan', value: 2 },
  { _id: 'Pepperoni', value: 3 },
  { _id: 'Cheese', value: 3 }
]
```

11. Use map reduce to group the pizzas by name and print the sum of twice(double) the price of the pizzas.

Query:

```
var mapper=function(){emit(this.name,2*this.price)};
var reducer=function(n,p){return Array.sum(p)};
db.pizza_rollno.mapReduce(mapper,reducer,{out:"double_pizza"});
db.double_pizza.find({},{});
```

```
restaurant> var mapper=function(){emit(this.name,2*this.price)};

restaurant> var reducer=function(n,p){return Array.sum(p)};

restaurant> db.pizza_rollno.mapReduce(mapper,reducer,{out:"double_pizza"});
{ result: 'double_pizza', ok: 1 }
restaurant> db.double_pizza.find({},{});
[
  { _id: 'Cheese', value: 78 },
  { _id: 'Vegan', value: 70 },
  { _id: 'Pepperoni', value: 120 }
]
```

12. Print the information about pizzas where price is between 20 and 40.

Query:

db.pizza_rollno.find({$and:[{'price':{$gte:20}},{'price':{$lte:40}}]});

```
restaurant> db.pizza_rollno.find({$and:[{'price':{$gte:20}},{'price':{$lte:40}}]});
[
  {
    _id: 1,
    name: 'Pepperoni',
    size: 'medium',
    price: 20,
    quantity: 20,
    date: ISODate("2021-03-13T09:13:24.000Z")
  },
  {
    _id: 2,
    name: 'Pepperoni',
    size: 'large',
    price: 21,
    quantity: 30,
    date: ISODate("2021-03-17T09:22:12.000Z")
  }
]
```

13. use aggregation pipeline to print the total quantity ordered according to name.

Query:

db.pizza_rollno.aggregate({$group:{_id:"$name",total_qty:{$sum:"$quantity"}}},
{$out:"grp_qty"});
db.grp_qty.find({});

```
restaurant> db.pizza_rollno.aggregate({$group:{_id:"$name",total_qty:{$sum:"$quantity"}}},{$out:"grp_qty"});

restaurant> db.grp_qty.find({});
[
  { _id: 'Cheese', total_qty: 75 },
  { _id: 'Vegan', total_qty: 20 },
  { _id: 'Pepperoni', total_qty: 60 }
]
```

14. use aggregation pipeline to print the name and price of the pepperoni pizza.

Query:

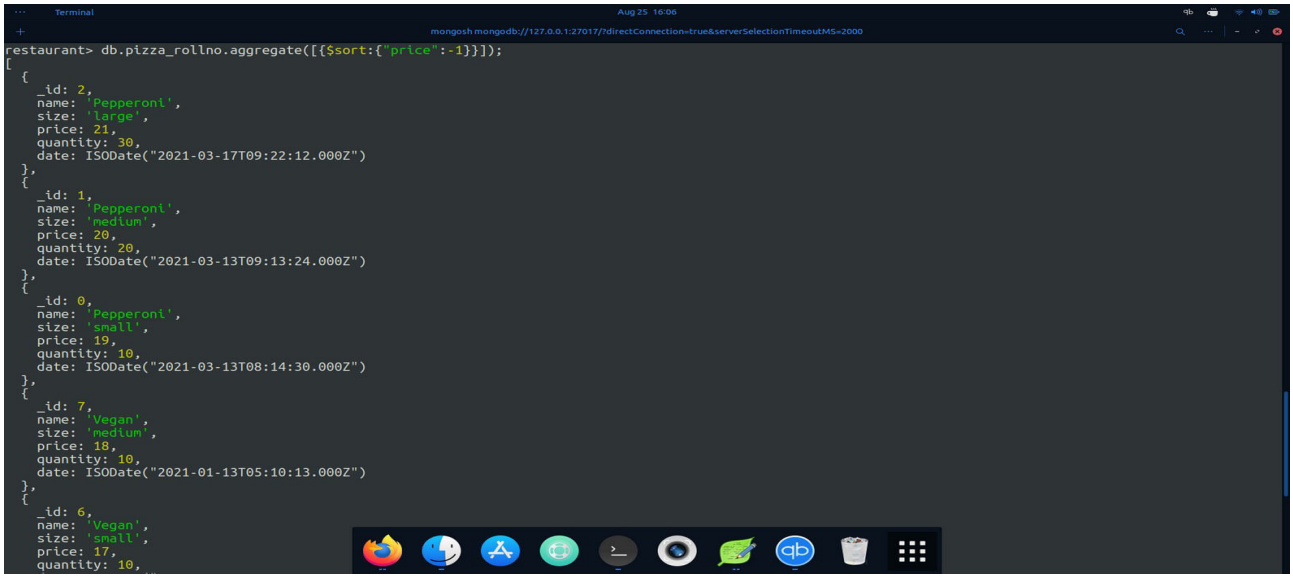
db.pizza_rollno.aggregate({$match:{"name":"Pepperoni"}},{$project:{"name":1,"price":1}});

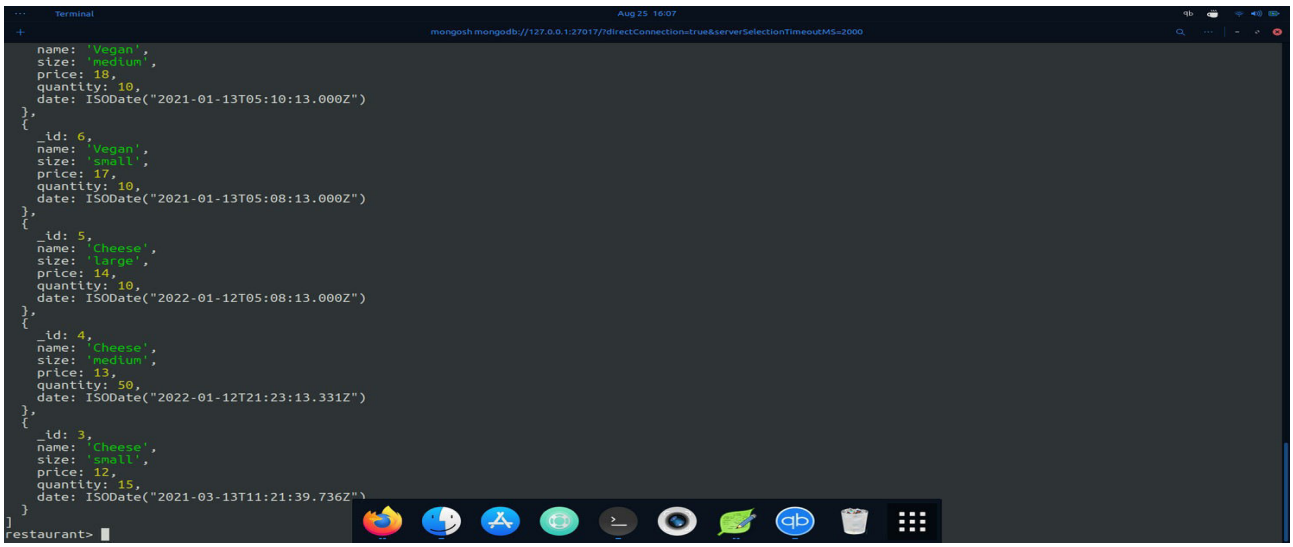
```
restaurant> db.pizza_rollno.aggregate({$match:{"name":"Pepperoni"}},{$project:{"name":1,"price":1}});
[
  { _id: 0, name: 'Pepperoni', price: 19 },
  { _id: 1, name: 'Pepperoni', price: 20 },
  { _id: 2, name: 'Pepperoni', price: 21 }
]
```

15. Sort the pizzas by descending price. use aggregation pipeline.

Query:

db.pizza_rollno.aggregate([{$sort:{"price":-1}}]);





16. Print the sum of total quantity ordered between 01-01-2021 to 01-03-2021

Query:

db.pizza_rollno.aggregate([{$match:{"date":{$gte:new ISODate("2021-01-01"),$lte:new ISODate("2021-03-30")}}},{$group:{_id:{$dateToString:{format:"%Y-%m-%d",date:"$date"}},"tot_qty":{$sum:"$quantity"}}}]);

```
restaurant> db.pizza_rollno.aggregate([{$match:{"date":{$gte:new ISODate("2021-01-01"),$lte:new ISODate("2021-03-30")}}},{$group:{_id:{$dateToString:{format:"
%Y-%m-%d",date:"$date"}},"tot_qty":{$sum:"$quantity"}}}]);
[
  { _id: '2021-03-13', tot_qty: 45 },
  { _id: '2021-03-17', tot_qty: 30 },
  { _id: '2021-01-13', tot_qty: 20 }
]
```

17. To print the minimum price of the pizza as per size.

Query:

db.pizza_rollno.aggregate([{$group:{_id:"$size","min_price":{$min:"$price"}}}]);

```
restaurant> db.pizza_rollno.aggregate([{$group:{_id:"$size","min_price":{$min:"$price"}}}]);
[
  { _id: 'large', min_price: 14 },
  { _id: 'small', min_price: 12 },
  { _id: 'medium', min_price: 13 }
]
```

18. Print the name , size, price and discount which is calculated as , if quantity ordered is greater than 20 then 5/- discount else 2/- discount. (use aggregation pipeline)

Query:

db.pizza_rollno.aggregate({$project:{"name":1,"size":1,"price":1,"discount":{$cond:{if:{$gte:["$quantity",20]},then:5,else:2}}}});

```
restaurant> db.pizza_rollno.aggregate({$project:{"name":1,"size":1,"price":1,"discount":{$cond:{if:{$gte:["$quantity",20]},then:5,else:2}}}});
[
  { _id: 0, name: 'Pepperoni', size: 'small', price: 19, discount: 2 },
  { _id: 1, name: 'Pepperoni', size: 'medium', price: 20, discount: 5 },
  { _id: 2, name: 'Pepperoni', size: 'large', price: 21, discount: 5 },
  { _id: 3, name: 'Cheese', size: 'small', price: 12, discount: 2 },
  { _id: 4, name: 'Cheese', size: 'medium', price: 13, discount: 5 },
  { _id: 5, name: 'Cheese', size: 'large', price: 14, discount: 2 },
  { _id: 6, name: 'Vegan', size: 'small', price: 17, discount: 2 },
  { _id: 7, name: 'Vegan', size: 'medium', price: 18, discount: 2 }
]
```

19. Print the pizza name, size and new_price which is calculated as per size , if size is small new_price is 10, if size is medium ,new_price is 20 and if size is large new_price is 40.

Query:

db.pizza_rollno.aggregate({$project:{"name":1,"size":1,"new_price":{$cond:{if:{$eq:["$size","small"]},then:10,else:{$cond:{if:{$eq:["$size","medium"]},then:20,else:40}}}}}});

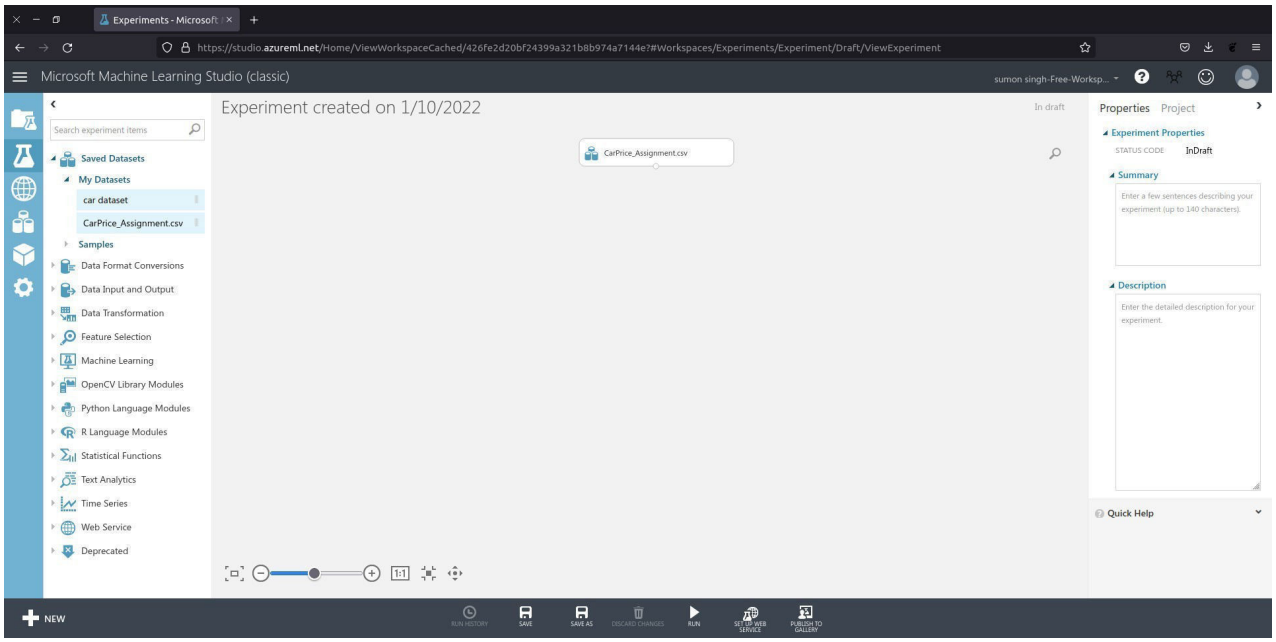
```
restaurant> db.pizza_rollno.aggregate({$project:{"name":1,"size":1,"new_price":{$cond:{if:{$eq:["$size","small"]},then:10,else:{$cond:{if:{$eq:["$size","mediu
m"]},then:20,else:40}}}}}});
[
  { _id: 0, name: 'Pepperoni', size: 'small', new_price: 10 },
  { _id: 1, name: 'Pepperoni', size: 'medium', new_price: 20 },
  { _id: 2, name: 'Pepperoni', size: 'large', new_price: 40 },
  { _id: 3, name: 'Cheese', size: 'small', new_price: 10 },
  { _id: 4, name: 'Cheese', size: 'medium', new_price: 20 },
  { _id: 5, name: 'Cheese', size: 'large', new_price: 40 },
  { _id: 6, name: 'Vegan', size: 'small', new_price: 10 },
  { _id: 7, name: 'Vegan', size: 'medium', new_price: 20 }
]
```

Practical : 3 (AutoML)

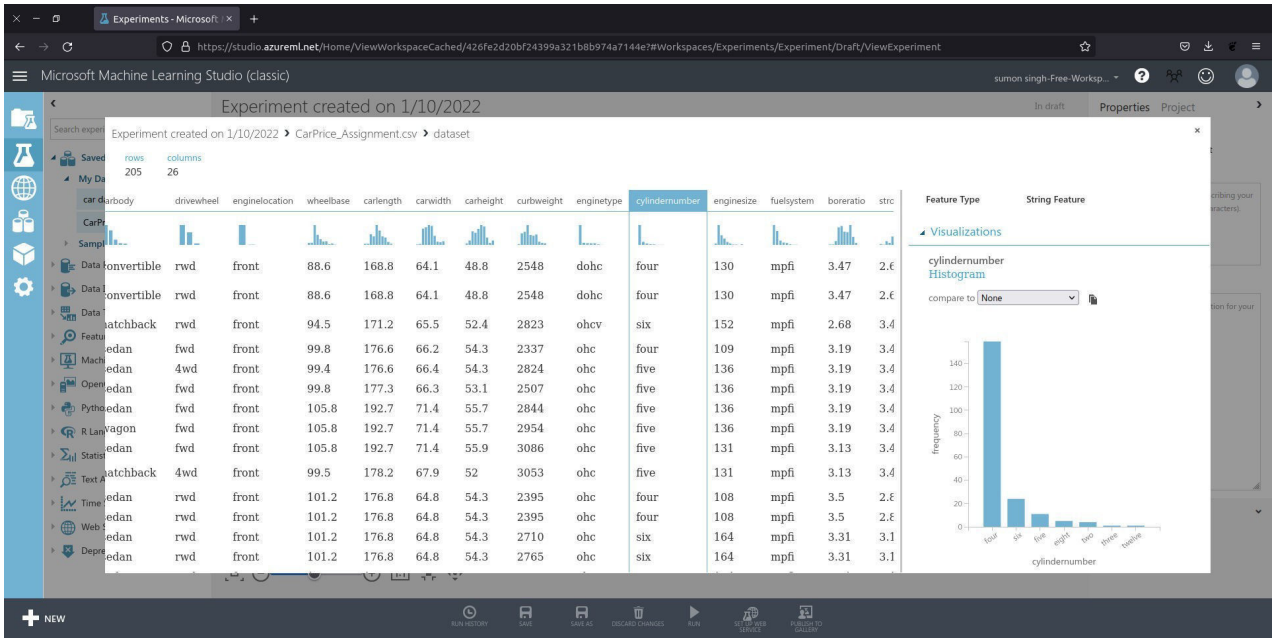
Tasks :

Create a Automl project on car price dataset in Azure ml

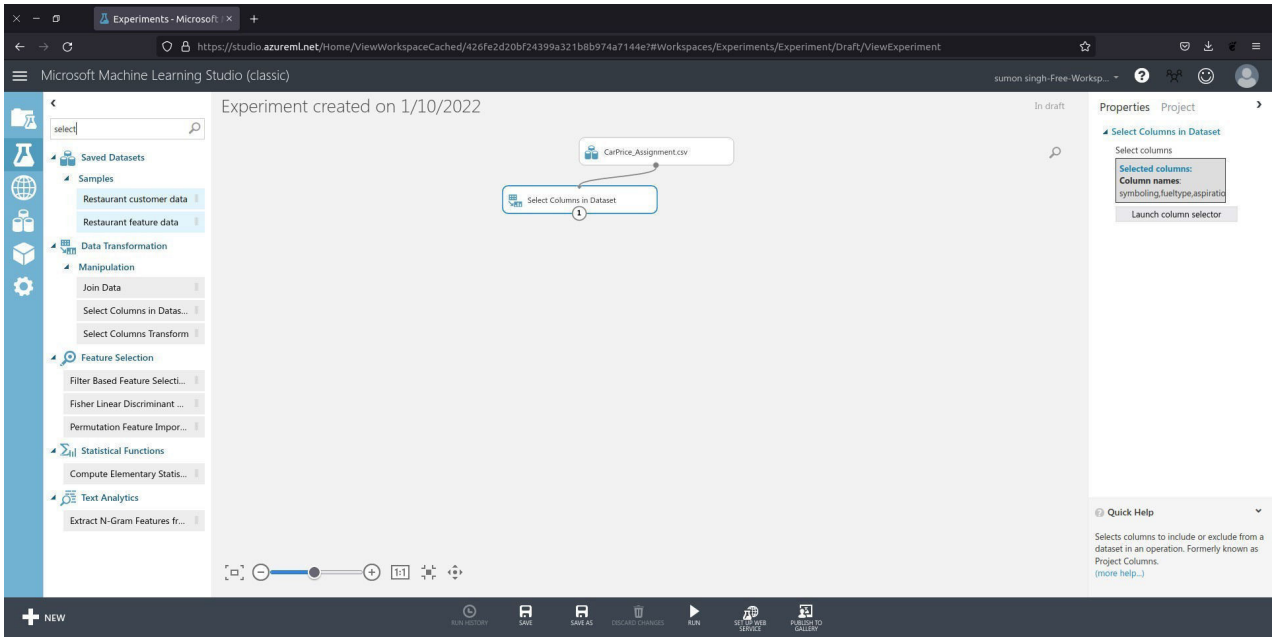
1. Import Dataset



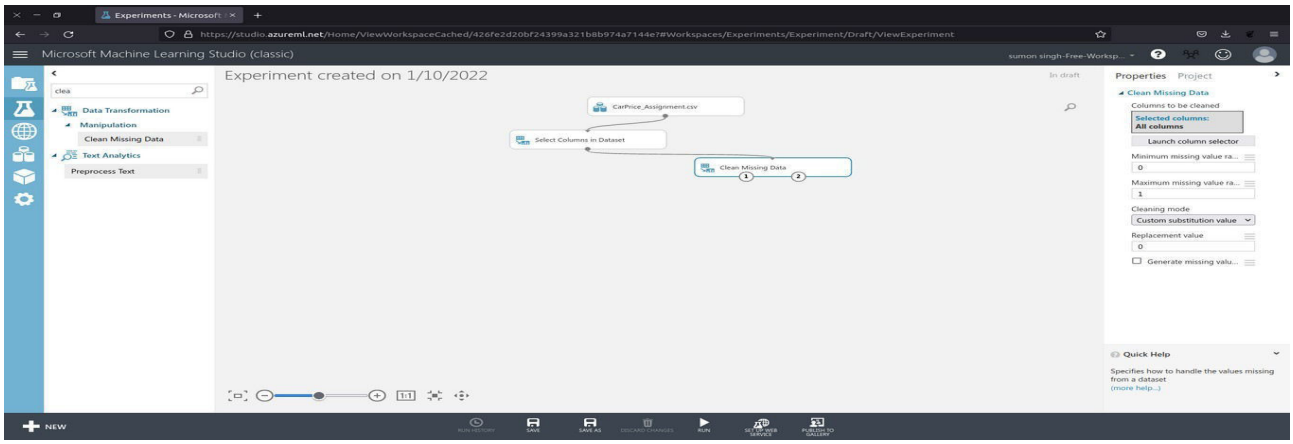
2. Visualization

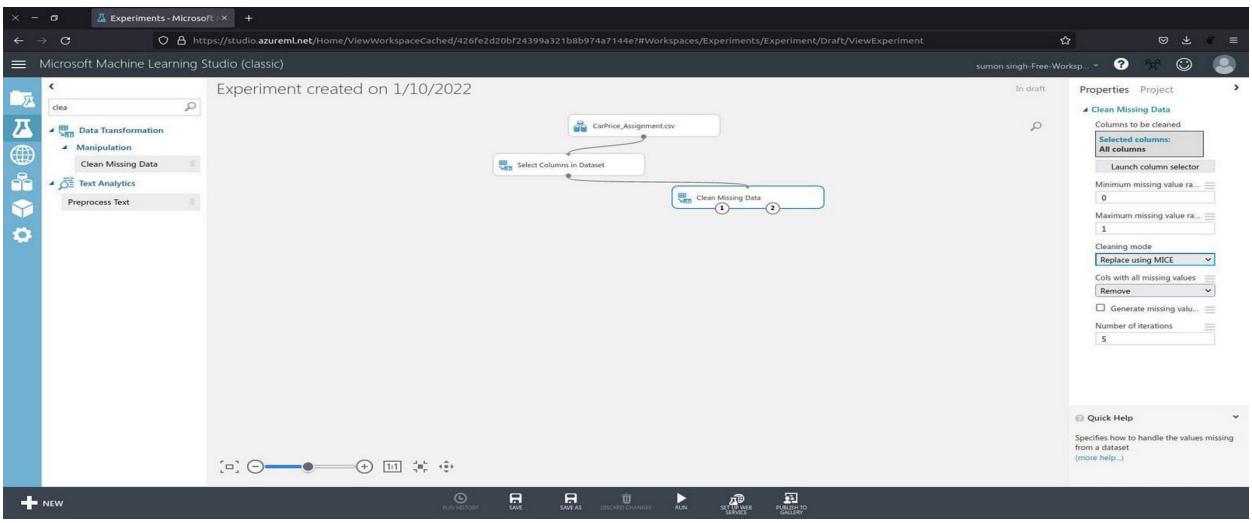


3. Select Columns

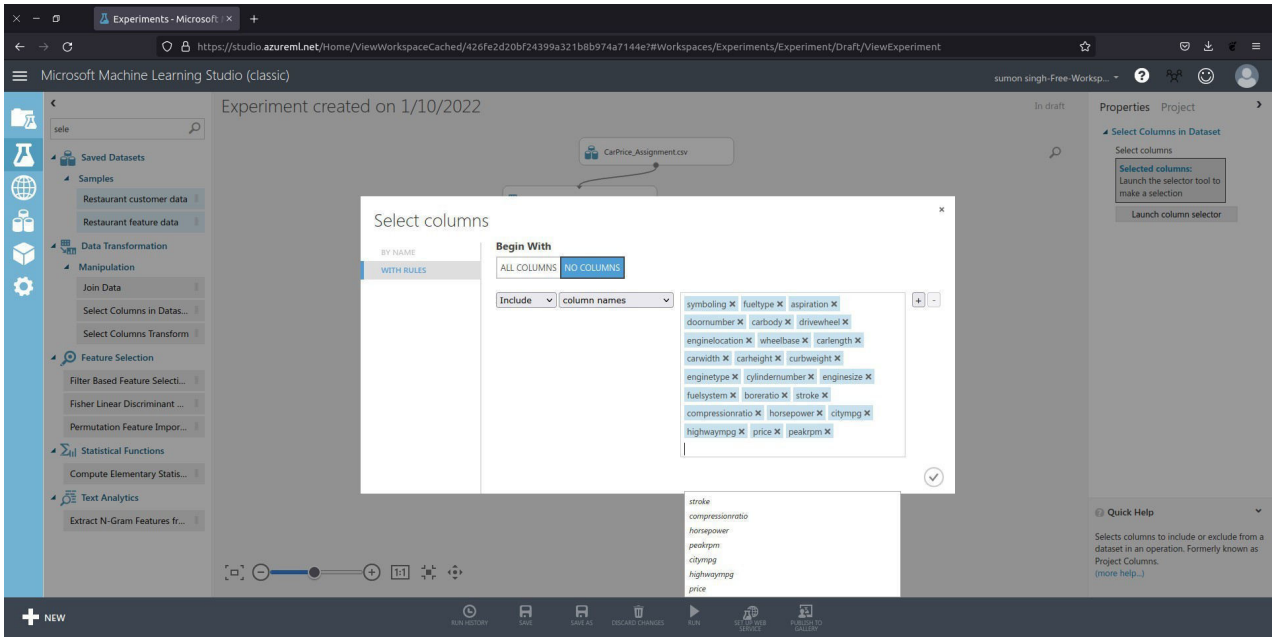


4. Clean Missing Data

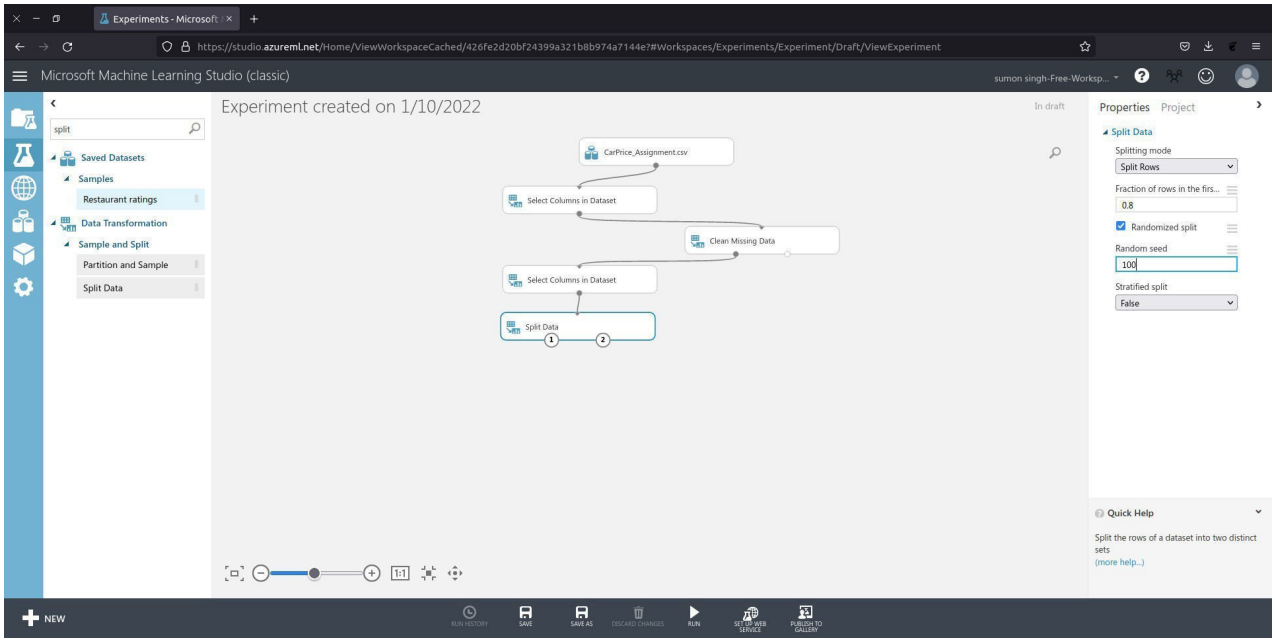




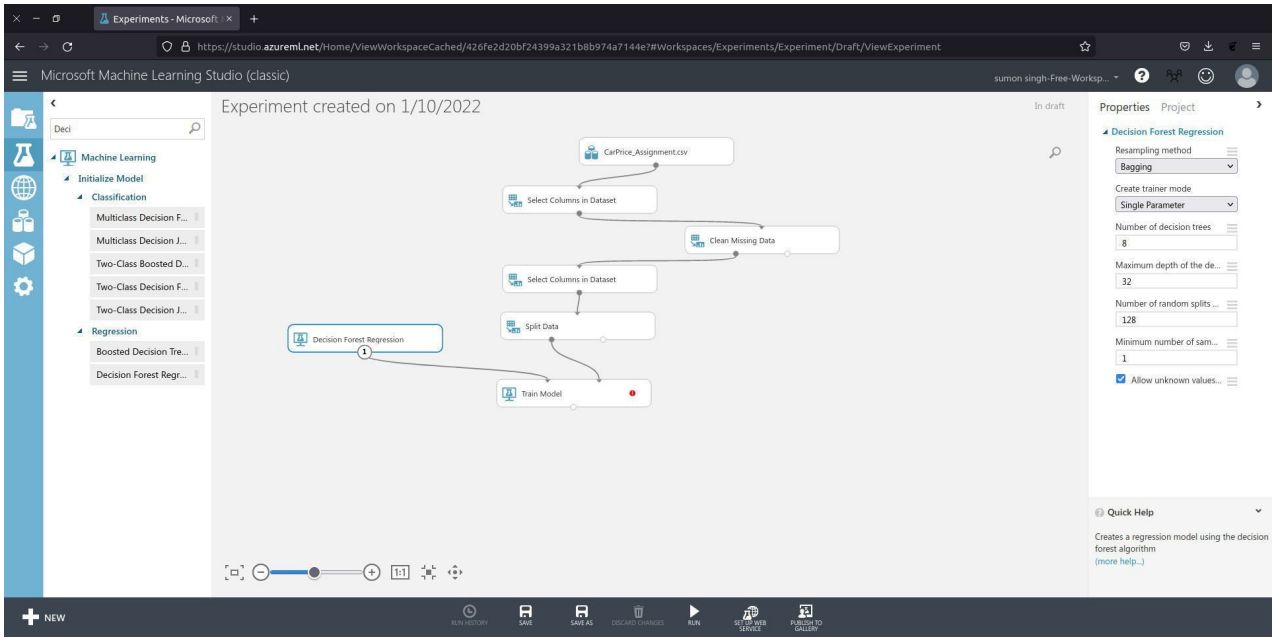
5. Select Columns



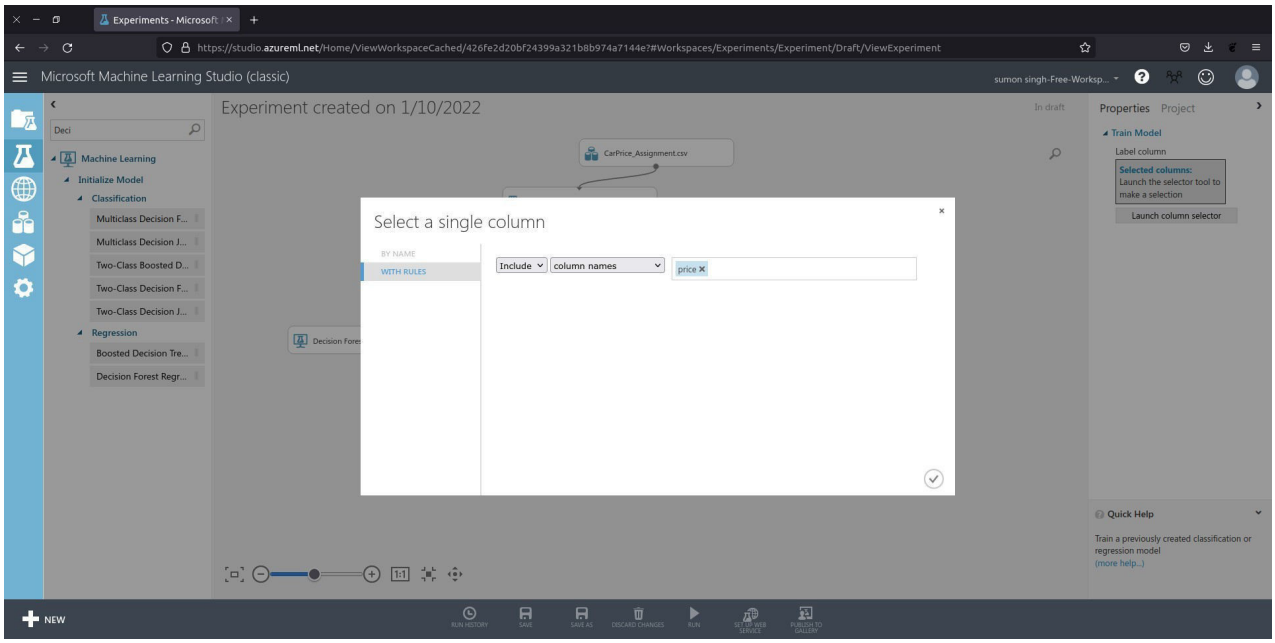
6. Split Dataset Into 80:20



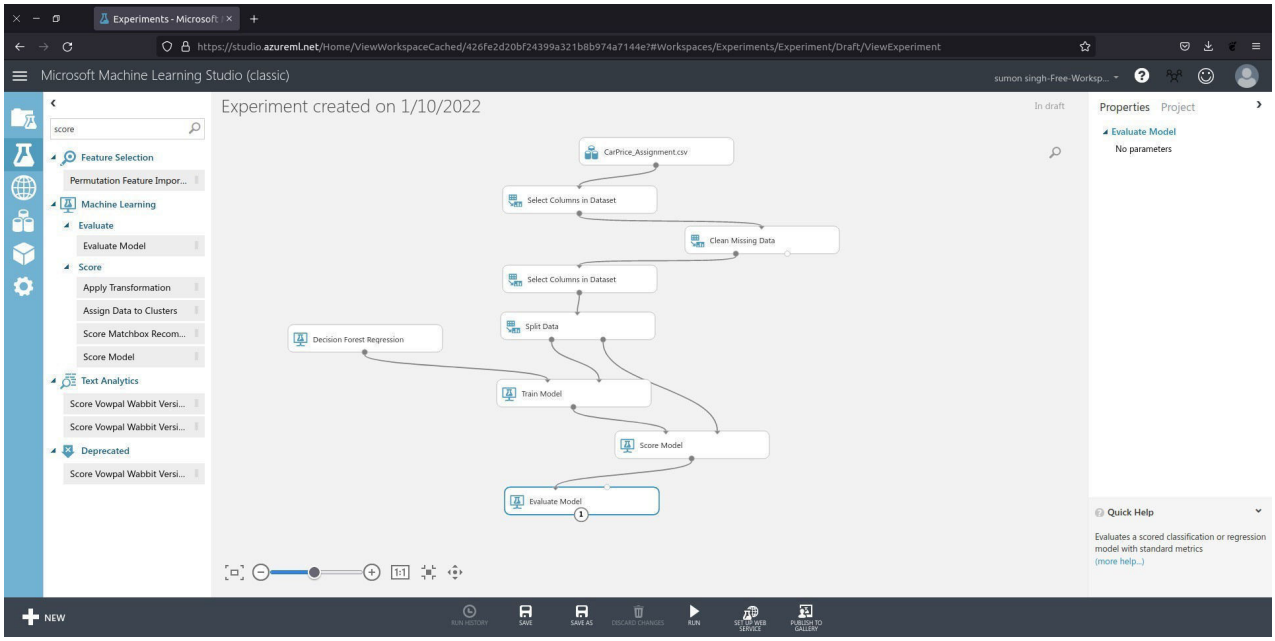
7. Select Decision Forest Tree



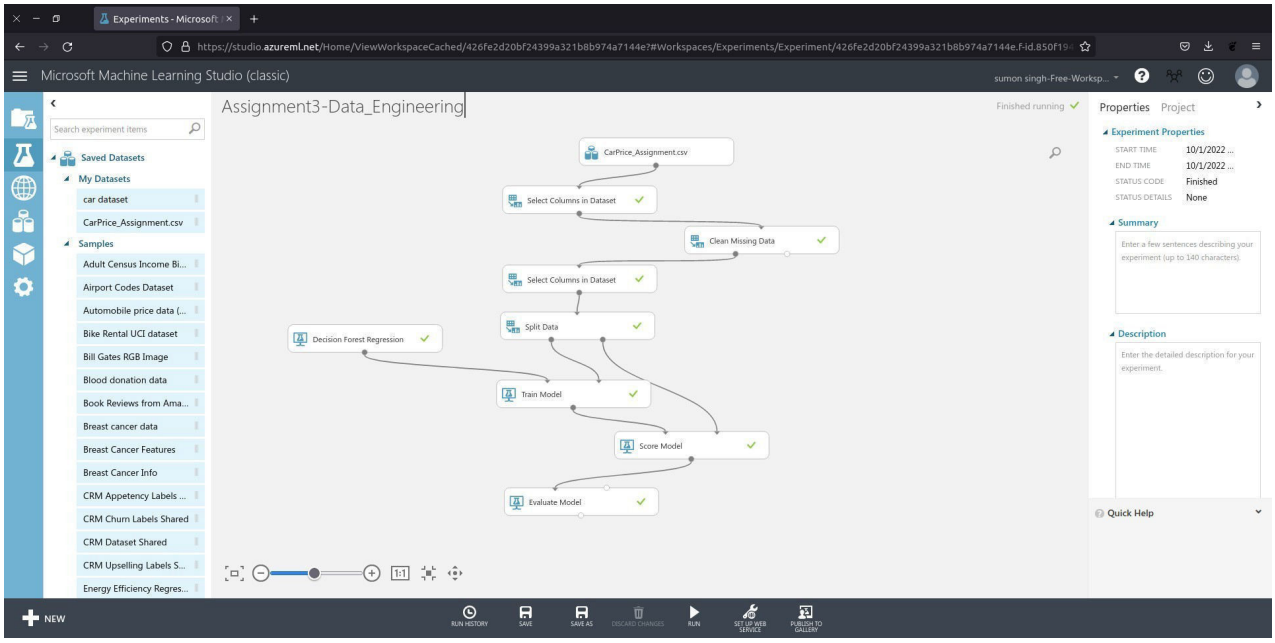
8. Select Target Column



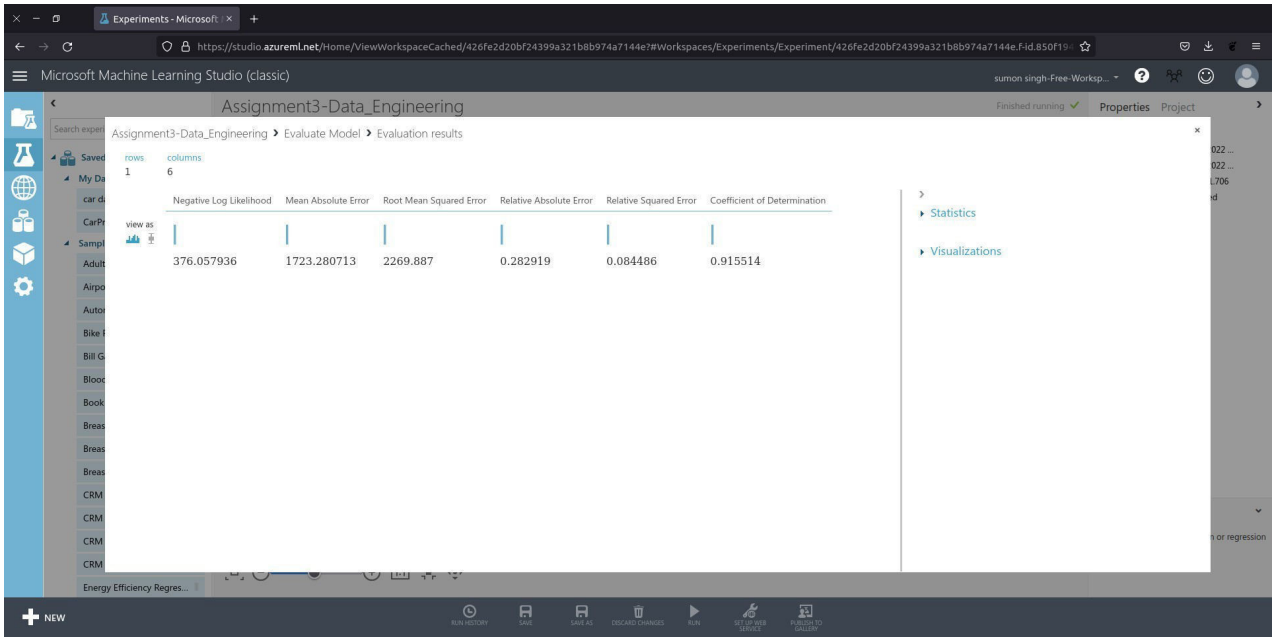
9. Evaluation



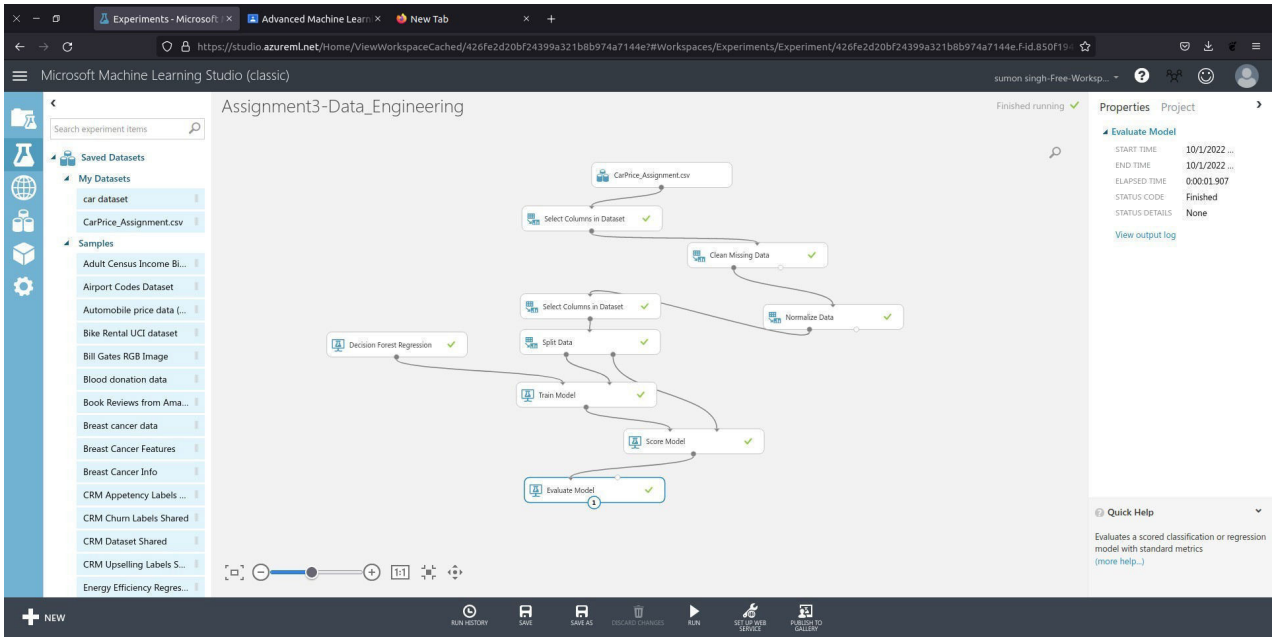
10. Train The Model



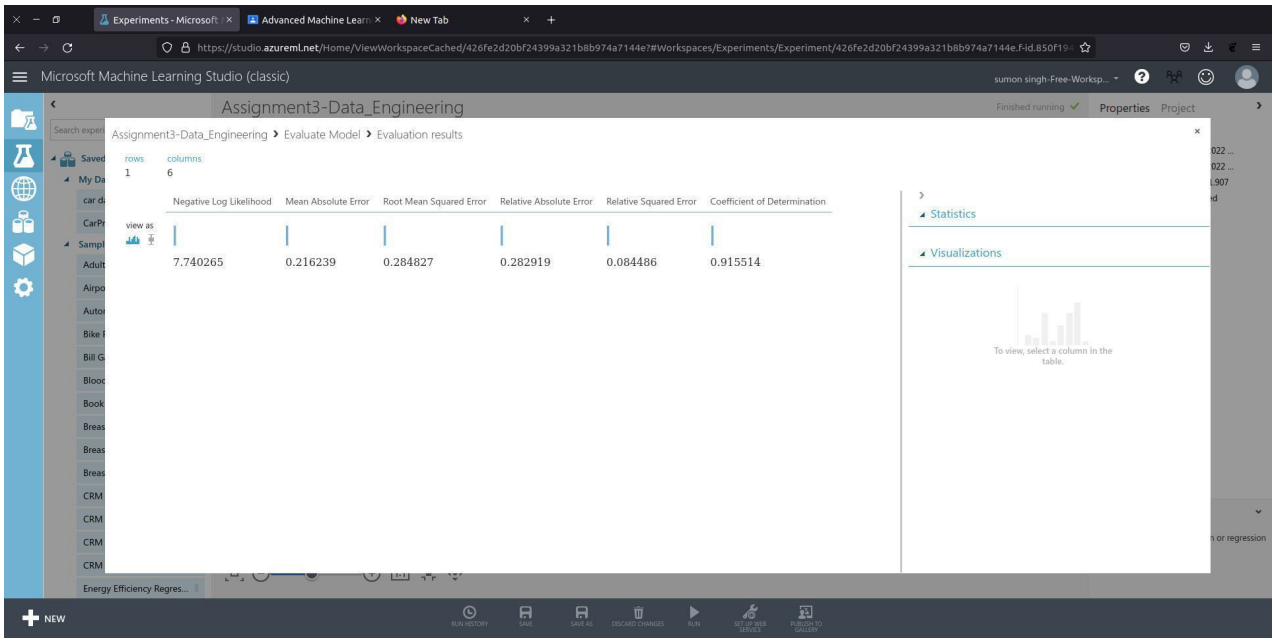
11. Result



12. Train Model On Normalized Data



13. Result For Normalized Data



Practical : 4

Aim : Data exloration using pyspark SQL

Task :

```
Using distributed data Spark RDD in the cloud.
Use Iris dataset to divide the data into chunks
and perform distributed data processing in the cloud
```

Import Dataset

In [0]:

```
rdd_iris1 = sc.textFile('/FileStore/tables/Iris_New.csv')
```

Retrieve elements from the dataset

In [0]:

```
rdd_iris1.collect()
```

```
Out[15]: ['1,5.1,3.5,1.4,0.2,Iris-setosa',
 '2,4.9,3,1.4,0.2,Iris-setosa',
 '3,4.7,3.2,1.3,0.2,Iris-setosa',
 '4,4.6,3.1,1.5,0.2,Iris-setosa',
 '5,5,3.6,1.4,0.2,Iris-setosa',
 '6,5.4,3.9,1.7,0.4,Iris-setosa',
 '7,4.6,3.4,1.4,0.3,Iris-setosa',
 '8,5,3.4,1.5,0.2,Iris-setosa',
 '9,4.4,2.9,1.4,0.2,Iris-setosa',
 '10,4.9,3.1,1.5,0.1,Iris-setosa',
 '11,5.4,3.7,1.5,0.2,Iris-setosa',
 '12,4.8,3.4,1.6,0.2,Iris-setosa',
 '13,4.8,3,1.4,0.1,Iris-setosa',
 '14,4.3,3,1.1,0.1,Iris-setosa',
 '15,5.8,4,1.2,0.2,Iris-setosa',
 '16,5.7,4.4,1.5,0.4,Iris-setosa',
 '17,5.4,3.9,1.3,0.4,Iris-setosa',
 '18,5.1,3.5,1.4,0.3,Iris-setosa',
 '19,5.7,3.8,1.7,0.3,Iris-setosa',
```

Number of parition

In [0]:

```
rdd_iris1.getNumPartitions()
```

```
Out[16]: 2
```

Printing the length of specific patitions

In [0]:

```
len(rdd_iris1.glom().collect()[0])
```

```
Out[17]: 78
```

In [0]:

```
len(rdd_iris1.glom().collect()[1])
```

```
Out[18]: 72
```

Parallelizing data

In [0]:

```
rdd_iris2 = sc.parallelize(rdd_iris1.collect(),8)
```

In [0]:

```
rdd_iris2.getNumPartitions()
```

Out[20]: 8

In [0]:

```
for i in rdd_iris1.collect():
    print(i)
```

```
1,5.1,3.5,1.4,0.2,Iris-setosa
2,4.9,3,1.4,0.2,Iris-setosa
3,4.7,3.2,1.3,0.2,Iris-setosa
4,4.6,3.1,1.5,0.2,Iris-setosa
5,5,3.6,1.4,0.2,Iris-setosa
6,5.4,3.9,1.7,0.4,Iris-setosa
7,4.6,3.4,1.4,0.3,Iris-setosa
8,5,3.4,1.5,0.2,Iris-setosa
9,4.4,2.9,1.4,0.2,Iris-setosa
10,4.9,3.1,1.5,0.1,Iris-setosa
11,5.4,3.7,1.5,0.2,Iris-setosa
12,4.8,3.4,1.6,0.2,Iris-setosa
13,4.8,3,1.4,0.1,Iris-setosa
14,4.3,3,1.1,0.1,Iris-setosa
15,5.8,4,1.2,0.2,Iris-setosa
16,5.7,4.4,1.5,0.4,Iris-setosa
17,5.4,3.9,1.3,0.4,Iris-setosa
18,5.1,3.5,1.4,0.3,Iris-setosa
19,5.7,3.8,1.7,0.3,Iris-setosa
```

map transformation

In [0]:

```
rdd_iris3 = rdd_iris2.map(lambda x:x*2)
```

In [0]:

```
rdd_iris3.collect()
```

```
Out[23]: ['1,5.1,3.5,1.4,0.2,Iris-setosa1,5.1,3.5,1.4,0.2,Iris-setosa',
 '2,4.9,3,1.4,0.2,Iris-setosa2,4.9,3,1.4,0.2,Iris-setosa',
 '3,4.7,3.2,1.3,0.2,Iris-setosa3,4.7,3.2,1.3,0.2,Iris-setosa',
 '4,4.6,3.1,1.5,0.2,Iris-setosa4,4.6,3.1,1.5,0.2,Iris-setosa',
 '5,5,3.6,1.4,0.2,Iris-setosa5,5,3.6,1.4,0.2,Iris-setosa',
 '6,5.4,3.9,1.7,0.4,Iris-setosa6,5.4,3.9,1.7,0.4,Iris-setosa',
 '7,4.6,3.4,1.4,0.3,Iris-setosa7,4.6,3.4,1.4,0.3,Iris-setosa',
 '8,5,3.4,1.5,0.2,Iris-setosa8,5,3.4,1.5,0.2,Iris-setosa',
 '9,4.4,2.9,1.4,0.2,Iris-setosa9,4.4,2.9,1.4,0.2,Iris-setosa',
 '10,4.9,3.1,1.5,0.1,Iris-setosa10,4.9,3.1,1.5,0.1,Iris-setosa',
 '11,5.4,3.7,1.5,0.2,Iris-setosa11,5.4,3.7,1.5,0.2,Iris-setosa',
 '12,4.8,3.4,1.6,0.2,Iris-setosa12,4.8,3.4,1.6,0.2,Iris-setosa',
 '13,4.8,3,1.4,0.1,Iris-setosa13,4.8,3,1.4,0.1,Iris-setosa',
 '14,4.3,3,1.1,0.1,Iris-setosa14,4.3,3,1.1,0.1,Iris-setosa',
 '15,5.8,4,1.2,0.2,Iris-setosa15,5.8,4,1.2,0.2,Iris-setosa',
 '16,5.7,4.4,1.5,0.4,Iris-setosa16,5.7,4.4,1.5,0.4,Iris-setosa',
 '17,5.4,3.9,1.3,0.4,Iris-setosa17,5.4,3.9,1.3,0.4,Iris-setosa',
 '18,5.1,3.5,1.4,0.3,Iris-setosa18,5.1,3.5,1.4,0.3,Iris-setosa',
 '19,5.7,3.8,1.7,0.3,Iris-setosa19,5.7,3.8,1.7,0.3,Iris-setosa',
```

Flattening a text file

In [0]:

```
rdd_iris4 = rdd_iris2.flatMap(lambda x:x.split(","))
```

In [0]:

```
rdd_iris4.collect()
```

```
Out[28]: ['1',
 '5.1',
 '3.5',
 '1.4',
 '0.2',
 'Iris-setosa',
 '2',
 '4.9',
 '3',
 '1.4',
 '0.2',
 'Iris-setosa',
 '3',
 '4.7',
 '3.2',
 '1.3',
 '0.2',
 'Iris-setosa',
 '4',
```

In [0]:

```
rdd_iris4.getNumPartitions()
```

```
Out[29]: 8
```

In [0]:

```
rdd_iris4.glom().collect()[0]
```

```
Out[30]: ['1',
 '5.1',
 '3.5',
 '1.4',
 '0.2',
 'Iris-setosa',
 '2',
 '4.9',
 '3',
 '1.4',
 '0.2',
 'Iris-setosa',
 '3',
 '4.7',
 '3.2',
 '1.3',
 '0.2',
 'Iris-setosa',
 '4',
```

In [0]:

```
rdd_iris4.glom().collect()[1]
```

```
Out[31]: ['19',
 '5.7',
 '3.8',
 '1.7',
 '0.3',
 'Iris-setosa',
 '20',
 '5.1',
 '3.8',
 '1.5',
 '0.3',
 'Iris-setosa',
 '21',
 '5.4',
 '3.4',
 '1.7',
 '0.2',
 'Iris-setosa',
 '22',
```

In [0]:

```
rdd_iris5 = rdd_iris4.map(lambda x:x*2)
```

In [0]:

```
rdd_iris5.collect()
```

```
Out[33]: ['11',
 '5.15.1',
 '3.53.5',
 '1.41.4',
 '0.20.2',
 'Iris-setosaIris-setosa',
 '22',
 '4.94.9',
 '33',
 '1.41.4',
 '0.20.2',
 'Iris-setosaIris-setosa',
 '33',
 '4.74.7',
 '3.23.2',
 '1.31.3',
 '0.20.2',
 'Iris-setosaIris-setosa',
 '44',
```

Filtering

In [0]:

```
rdd_iris6 = rdd_iris1.flatMap(lambda x:x.split(",")).flter(lambda x:x=="Iris-virginica").map(lambda x:(x,1))
```

In [0]:

```
rdd_iris6.glom().collect()[1]
```

```
Out[40]: [('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1),
 ('Iris-virginica', 1)]
```

In [0]:

```
rdd_species = rdd_iris1.flatMap(lambda x:x.split(",")).
filter(lambda x:(x=="Iris-virginica" or
                 x=="Iris-setosa" or x=="Iris-versicolor"))
```

In [0]:

```
rdd_species.collect()
```

```
Out[42]: ['Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
 'Iris-setosa',
```

In [0]:

```
rdd_species1 = rdd_species.groupBy(lambda x:x).mapValues(lambda x:len(x))
```

In [0]:

```
rdd_species1.collect()
```

```
Out[44]: [('Iris-setosa', 50), ('Iris-virginica', 50), ('Iris-versicolor', 50)]
```

Practical : 5

Aim : Perform distributed data processing using Pyspark RDD ¶

Load csv file

In [0]:

```
df_iris=spark.read.format("csv").option("multiline",True).
option("header",True).option("inferschema",True).load("/FileStore/tables/Iris.csv")
```

In [0]:

```
df_iris.show()
```

```
+---+-------------+------------+-------------+------------+---------------+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|        Species|
+---+-------------+------------+-------------+------------+---------------+
|115|          5.8|         2.8|          5.1|         2.4| Iris-virginica|
|114|          5.7|         2.5|          5.0|         2.0| Iris-virginica|
|147|          6.3|         2.5|          5.0|         1.9| Iris-virginica|
|102|          5.8|         2.7|          5.1|         1.9| Iris-virginica|
|143|          5.8|         2.7|          5.1|         1.9| Iris-virginica|
|133|          6.4|         2.8|          5.6|         2.2| Iris-virginica|
|122|          5.6|         2.8|          4.9|         2.0| Iris-virginica|
|112|          6.4|         2.7|          5.3|         1.9| Iris-virginica|
|129|          6.4|         2.8|          5.6|         2.1| Iris-virginica|
|146|          6.7|         3.0|          5.2|         2.3| Iris-virginica|
|124|          6.3|         2.7|          4.9|         1.8| Iris-virginica|
|127|          6.2|         2.8|          4.8|         1.8| Iris-virginica|
|116|          6.4|         3.2|          5.3|         2.3| Iris-virginica|
|120|          6.0|         2.2|          5.0|         1.5| Iris-virginica|
|148|          6.5|         3.0|          5.2|         2.0| Iris-virginica|
|141|          6.7|         3.1|          5.6|         2.4| Iris-virginica|
| 84|          6.0|         2.7|          5.1|         1.6|Iris-versicolor|
|128|          6.1|         3.0|          4.9|         1.8| Iris-virginica|
|150|          5.9|         3.0|          5.1|         1.8| Iris-virginica|
|139|          6.0|         3.0|          4.8|         1.8| Iris-virginica|
+---+-------------+------------+-------------+------------+---------------+
only showing top 20 rows
```

In [0]:

```
df_iris.printSchema()
```

```
root
 |-- Id: integer (nullable = true)
 |-- SepalLengthCm: double (nullable = true)
 |-- SepalWidthCm: double (nullable = true)
 |-- PetalLengthCm: double (nullable = true)
 |-- PetalWidthCm: double (nullable = true)
 |-- Species: string (nullable = true)
```

In [0]:

```
df_iris.count()
```

```
Out[7]: 150
```

To print the summary of the columns

In [0]:

```
df_iris.summary().show()
```

```
+-------+----------------+------------------+-------------------+------------------+------------------+
--------------+
|summary|              Id|     SepalLengthCm|       SepalWidthCm|     PetalLengthCm|      PetalWidthCm|
Species|
+-------+----------------+------------------+-------------------+------------------+------------------+
--------------+
|  count|             150|               150|                150|               150|               150|
150|
|   mean|            75.5| 5.843333333333335|              3.054|3.7586666666666635| 1.198666666666665|
null|
| stddev|43.4453679924569|0.8280661279778638|0.43359431136217413|1.7644204199522626|0.7631607417008411|
null|
|    min|               1|               4.3|                2.0|               1.0|               0.1|
Iris-setosa|
|    25%|              38|               5.1|                2.8|               1.6|               0.3|
null|
|    50%|              75|               5.8|                3.0|               4.3|               1.3|
null|
|    75%|             113|               6.4|                3.3|               5.1|               1.8|
null|
|    max|             150|               7.9|                4.4|               6.9|               2.5|
Iris-virginica|
+-------+----------------+------------------+-------------------+------------------+------------------+
--------------+
```

In [0]:

```
df_iris.select('SepalLengthCm').summary().show()
```

```
+-------+------------------+
|summary|     SepalLengthCm|
+-------+------------------+
|  count|               150|
|   mean| 5.843333333333335|
| stddev|0.8280661279778638|
|    min|               4.3|
|    25%|               5.1|
|    50%|               5.8|
|    75%|               6.4|
|    max|               7.9|
+-------+------------------+
```

Select specific columns

In [0]:

```
df_iris.select("SepalLengthCm","PetalLengthCm").show()
```

```
+-------------+-------------+
|SepalLengthCm|PetalLengthCm|
+-------------+-------------+
|          5.8|          5.1|
|          5.7|          5.0|
|          6.3|          5.0|
|          5.8|          5.1|
|          5.8|          5.1|
|          6.4|          5.6|
|          5.6|          4.9|
|          6.4|          5.3|
|          6.4|          5.6|
|          6.7|          5.2|
|          6.3|          4.9|
|          6.2|          4.8|
|          6.4|          5.3|
|          6.0|          5.0|
|          6.5|          5.2|
|          6.7|          5.6|
|          6.0|          5.1|
|          6.1|          4.9|
|          5.9|          5.1|
|          6.0|          4.8|
+-------------+-------------+
only showing top 20 rows
```

In [0]:

```
df_iris.createOrReplaceTempView("iristab")
```

Select the flowers where the petal length less than 2.5

In [0]:

```
df_iris1=spark.sql("select * from iristab where PetalLengthCm<2.5")
```

In [0]:

```
df_iris1.show()
```

```
+---+-------------+------------+-------------+------------+-----------+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|    Species|
+---+-------------+------------+-------------+------------+-----------+
| 24|          5.1|         3.3|          1.7|         0.5|Iris-setosa|
| 45|          5.1|         3.8|          1.9|         0.4|Iris-setosa|
| 44|          5.0|         3.5|          1.6|         0.6|Iris-setosa|
| 25|          4.8|         3.4|          1.9|         0.2|Iris-setosa|
| 21|          5.4|         3.4|          1.7|         0.2|Iris-setosa|
| 27|          5.0|         3.4|          1.6|         0.4|Iris-setosa|
| 32|          5.4|         3.4|          1.5|         0.4|Iris-setosa|
| 26|          5.0|         3.0|          1.6|         0.2|Iris-setosa|
| 19|          5.7|         3.8|          1.7|         0.3|Iris-setosa|
|  6|          5.4|         3.9|          1.7|         0.4|Iris-setosa|
| 31|          4.8|         3.1|          1.6|         0.2|Iris-setosa|
| 30|          4.7|         3.2|          1.6|         0.2|Iris-setosa|
| 46|          4.8|         3.0|          1.4|         0.3|Iris-setosa|
| 22|          5.1|         3.7|          1.5|         0.4|Iris-setosa|
| 12|          4.8|         3.4|          1.6|         0.2|Iris-setosa|
| 40|          5.1|         3.4|          1.5|         0.2|Iris-setosa|
| 28|          5.2|         3.5|          1.5|         0.2|Iris-setosa|
|  2|          4.9|         3.0|          1.4|         0.2|Iris-setosa|
|  8|          5.0|         3.4|          1.5|         0.2|Iris-setosa|
| 10|          4.9|         3.1|          1.5|         0.1|Iris-setosa|
+---+-------------+------------+-------------+------------+-----------+
only showing top 20 rows
```

In [0]:

```
df_iris1.count()
```

```
Out[14]: 50
```

Select the flowers where the petal length between 2.5 and 4.5

In [0]:

```
df_iris1=spark.sql("select * from iristab where
                   PetalLengthCm between 2.5 and 4.5")
```

In [0]:

```
df_iris1.show()
```

```
+---+-------------+------------+-------------+------------+--------------+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|       Species|
+---+-------------+------------+-------------+------------+--------------+
| 69|          6.2|         2.2|          4.5|         1.5|Iris-versicolor|
| 79|          6.0|         2.9|          4.5|         1.5|Iris-versicolor|
|107|          4.9|         2.5|          4.5|         1.7| Iris-virginica|
| 88|          6.3|         2.3|          4.4|         1.3|Iris-versicolor|
| 67|          5.6|         3.0|          4.5|         1.5|Iris-versicolor|
| 85|          5.4|         3.0|          4.5|         1.5|Iris-versicolor|
| 56|          5.7|         2.8|          4.5|         1.3|Iris-versicolor|
| 62|          5.9|         3.0|          4.2|         1.5|Iris-versicolor|
| 52|          6.4|         3.2|          4.5|         1.5|Iris-versicolor|
| 86|          6.0|         3.4|          4.5|         1.6|Iris-versicolor|
| 95|          5.6|         2.7|          4.2|         1.3|Iris-versicolor|
| 91|          5.5|         2.6|          4.4|         1.2|Iris-versicolor|
| 54|          5.5|         2.3|          4.0|         1.3|Iris-versicolor|
| 98|          6.2|         2.9|          4.3|         1.3|Iris-versicolor|
| 76|          6.6|         3.0|          4.4|         1.4|Iris-versicolor|
| 75|          6.4|         2.9|          4.3|         1.3|Iris-versicolor|
| 90|          5.5|         2.5|          4.0|         1.3|Iris-versicolor|
| 97|          5.7|         2.9|          4.2|         1.3|Iris-versicolor|
|100|          5.7|         2.8|          4.1|         1.3|Iris-versicolor|
| 72|          6.1|         2.8|          4.0|         1.3|Iris-versicolor|
+---+-------------+------------+-------------+------------+--------------+
only showing top 20 rows
```

In [0]:

```
df_iris1.count()
```

```
Out[17]: 37
```

Spark query to see the count of each species

In [0]:

```
df_iris.groupBy('Species').count().show()
```

```
+--------------+-----+
|       Species|count|
+--------------+-----+
| Iris-virginica|   50|
|    Iris-setosa|   50|
|Iris-versicolor|   50|
+--------------+-----+
```

In [0]:

```
df_iris1=spark.sql("Select Species, count(*)
                   from iristab group by Species")
```

In [0]:

```
df_iris1.show()
```

```
+--------------+--------+
|       Species|count(1)|
+--------------+--------+
| Iris-virginica|      50|
|    Iris-setosa|      50|
|Iris-versicolor|      50|
+--------------+--------+
```

Get the sepal length and sepal width of the species iris-setosa

In [0]:

```
df_iris2=spark.sql("select SepalLengthCm,SepalWidthCm,
                   Species from iristab
                   where Species=\"Iris-setosa\"")
```

In [0]:

```
df_iris2.show()
```

```
+-------------+------------+-----------+
|SepalLengthCm|SepalWidthCm|    Species|
+-------------+------------+-----------+
|          5.1|         3.3|Iris-setosa|
|          5.1|         3.8|Iris-setosa|
|          5.0|         3.5|Iris-setosa|
|          4.8|         3.4|Iris-setosa|
|          5.4|         3.4|Iris-setosa|
|          5.0|         3.4|Iris-setosa|
|          5.4|         3.4|Iris-setosa|
|          5.0|         3.0|Iris-setosa|
|          5.7|         3.8|Iris-setosa|
|          5.4|         3.9|Iris-setosa|
|          4.8|         3.1|Iris-setosa|
|          4.7|         3.2|Iris-setosa|
|          4.8|         3.0|Iris-setosa|
|          5.1|         3.7|Iris-setosa|
|          4.8|         3.4|Iris-setosa|
|          5.1|         3.4|Iris-setosa|
|          5.2|         3.5|Iris-setosa|
|          4.9|         3.0|Iris-setosa|
|          5.0|         3.4|Iris-setosa|
|          4.9|         3.1|Iris-setosa|
+-------------+------------+-----------+
only showing top 20 rows
```

Find the minimum sepal length for each species

In [0]:

```
df_iris2=spark.sql("select min(SepalLengthCm) as
                   min_sep_len,species from iristab group by Species")
```

In [0]:

```
df_iris2.show()
```

```
+-----------+--------------+
|min_sep_len|       species|
+-----------+--------------+
|        4.9| Iris-virginica|
|        4.3|    Iris-setosa|
|        4.9|Iris-versicolor|
+-----------+--------------+
```

In [0]:

```
df_iris.groupBy('Species').min("SepalLengthCm").show()
```

```
+--------------+-----------------+
|       Species|min(SepalLengthCm)|
+--------------+-----------------+
| Iris-virginica|              4.9|
|    Iris-setosa|              4.3|
|Iris-versicolor|              4.9|
+--------------+-----------------+
```

Find out the average petal length of the each species in the Iris dataset

In [0]:

```
df_iris.groupBy("Species").mean("PetalLengthCm").show()
```

```
+--------------+------------------+
|       Species|avg(PetalLengthCm)|
+--------------+------------------+
| Iris-virginica| 5.551999999999998|
|    Iris-setosa|1.4639999999999995|
|Iris-versicolor|              4.26|
+--------------+------------------+
```

To run multiple aggregates using sql

In [0]:

```
spark.sql("select Species,min(PetalLengthCm) as min_petal_len,max(PetalLengthCm)
          as max_petal_len,avg(PetalLengthCm) as
          mean_petal_len from iristab group by Species").show()
```

```
+--------------+-------------+-------------+------------------+
|       Species|min_petal_len|max_petal_len|    mean_petal_len|
+--------------+-------------+-------------+------------------+
| Iris-virginica|          4.5|          6.9| 5.551999999999998|
|    Iris-setosa|          1.0|          1.9|1.4639999999999995|
|Iris-versicolor|          3.0|          5.1|              4.26|
+--------------+-------------+-------------+------------------+
```

Print the petal length and petal width and species where average petal length is more than 4

In [0]:

```
spark.sql("select Species,PetalLengthCm,PetalWidthCm
          from iristab group by Species,PetalLengthCm,
          PetalWidthCm having mean(PetalLengthCm)>4").show()
```

```
+--------------+-------------+------------+
|       Species|PetalLengthCm|PetalWidthCm|
+--------------+-------------+------------+
|Iris-versicolor|          4.6|         1.4|
| Iris-virginica|          5.8|         1.8|
| Iris-virginica|          5.4|         2.3|
| Iris-virginica|          5.4|         2.1|
| Iris-virginica|          5.5|         2.1|
| Iris-virginica|          6.4|         2.0|
| Iris-virginica|          6.1|         1.9|
| Iris-virginica|          5.9|         2.1|
| Iris-virginica|          5.7|         2.5|
|Iris-versicolor|          4.2|         1.5|
|Iris-versicolor|          4.4|         1.3|
|Iris-versicolor|          4.6|         1.3|
|Iris-versicolor|          4.3|         1.3|
|Iris-versicolor|          4.7|         1.5|
| Iris-virginica|          5.0|         2.0|
|Iris-versicolor|          4.8|         1.4|
| Iris-virginica|          5.3|         1.9|
| Iris-virginica|          5.8|         2.2|
|Iris-versicolor|          4.1|         1.3|
| Iris-virginica|          5.7|         2.3|
+--------------+-------------+------------+
only showing top 20 rows
```

To print the data using spark where species is Iris-setosa

In [0]:

```
df_iris.where("Species==\"Iris-setosa\"").show()
```

```
+---+-------------+------------+-------------+------------+-----------+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|    Species|
+---+-------------+------------+-------------+------------+-----------+
| 24|          5.1|         3.3|          1.7|         0.5|Iris-setosa|
| 45|          5.1|         3.8|          1.9|         0.4|Iris-setosa|
| 44|          5.0|         3.5|          1.6|         0.6|Iris-setosa|
| 25|          4.8|         3.4|          1.9|         0.2|Iris-setosa|
| 21|          5.4|         3.4|          1.7|         0.2|Iris-setosa|
| 27|          5.0|         3.4|          1.6|         0.4|Iris-setosa|
| 32|          5.4|         3.4|          1.5|         0.4|Iris-setosa|
| 26|          5.0|         3.0|          1.6|         0.2|Iris-setosa|
| 19|          5.7|         3.8|          1.7|         0.3|Iris-setosa|
|  6|          5.4|         3.9|          1.7|         0.4|Iris-setosa|
| 31|          4.8|         3.1|          1.6|         0.2|Iris-setosa|
| 30|          4.7|         3.2|          1.6|         0.2|Iris-setosa|
| 46|          4.8|         3.0|          1.4|         0.3|Iris-setosa|
| 22|          5.1|         3.7|          1.5|         0.4|Iris-setosa|
| 12|          4.8|         3.4|          1.6|         0.2|Iris-setosa|
| 40|          5.1|         3.4|          1.5|         0.2|Iris-setosa|
| 28|          5.2|         3.5|          1.5|         0.2|Iris-setosa|
|  2|          4.9|         3.0|          1.4|         0.2|Iris-setosa|
|  8|          5.0|         3.4|          1.5|         0.2|Iris-setosa|
| 10|          4.9|         3.1|          1.5|         0.1|Iris-setosa|
+---+-------------+------------+-------------+------------+-----------+
only showing top 20 rows
```

Rename the column petal length to petal_len

In [0]:

```
df_iris2=df_iris.withColumnRenamed("PetalLengthCm","petal_len")
```

In [0]:

```
df_iris2.show()
```

```
+---+-------------+------------+---------+------------+---------------+
| Id|SepalLengthCm|SepalWidthCm|petal_len|PetalWidthCm|        Species|
+---+-------------+------------+---------+------------+---------------+
|115|          5.8|         2.8|      5.1|         2.4| Iris-virginica|
|114|          5.7|         2.5|      5.0|         2.0| Iris-virginica|
|147|          6.3|         2.5|      5.0|         1.9| Iris-virginica|
|102|          5.8|         2.7|      5.1|         1.9| Iris-virginica|
|143|          5.8|         2.7|      5.1|         1.9| Iris-virginica|
|133|          6.4|         2.8|      5.6|         2.2| Iris-virginica|
|122|          5.6|         2.8|      4.9|         2.0| Iris-virginica|
|112|          6.4|         2.7|      5.3|         1.9| Iris-virginica|
|129|          6.4|         2.8|      5.6|         2.1| Iris-virginica|
|146|          6.7|         3.0|      5.2|         2.3| Iris-virginica|
|124|          6.3|         2.7|      4.9|         1.8| Iris-virginica|
|127|          6.2|         2.8|      4.8|         1.8| Iris-virginica|
|116|          6.4|         3.2|      5.3|         2.3| Iris-virginica|
|120|          6.0|         2.2|      5.0|         1.5| Iris-virginica|
|148|          6.5|         3.0|      5.2|         2.0| Iris-virginica|
|141|          6.7|         3.1|      5.6|         2.4| Iris-virginica|
| 84|          6.0|         2.7|      5.1|         1.6|Iris-versicolor|
|128|          6.1|         3.0|      4.9|         1.8| Iris-virginica|
|150|          5.9|         3.0|      5.1|         1.8| Iris-virginica|
|139|          6.0|         3.0|      4.8|         1.8| Iris-virginica|
+---+-------------+------------+---------+------------+---------------+
only showing top 20 rows
```

Convert iris spark dataset to pandas

In [0]:

```
pd_iris=df_iris.toPandas()
```

In [0]:

```
pd_iris.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	115	5.8	2.8	5.1	2.4	Iris-virginica
1	114	5.7	2.5	5.0	2.0	Iris-virginica
2	147	6.3	2.5	5.0	1.9	Iris-virginica
3	102	5.8	2.7	5.1	1.9	Iris-virginica
4	143	5.8	2.7	5.1	1.9	Iris-virginica

Practical : 6

Aim : Data cleaning and transformation using ml pipeline

Import Libraries

In [1]:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score,classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
```

Load Dataset

In [2]:

```
df_train = pd.read_csv('/home/sumon/Documents/Datasets/titan_train.csv')
df_test = pd.read_csv('/home/sumon/Documents/Datasets/titan_test.csv')
```

Dataset Checking

In [3]:

```
df_train.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [4]:

```
df_test.head()
```

Out[4]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [5]:

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [6]:

```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

In [7]:

```
df_train.isna().sum()
```

Out[7]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [8]:

```
df_test.isna().sum()
```

Out[8]:

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [9]:

```
def cleaning_data(df):
    m=np.mean(df['Age'])
    df['Age'].fillna(m,inplace=True)
    df['Embarked'].fillna('S',inplace=True)
    df.drop(['Ticket','Name','Cabin','PassengerId'],axis=1,inplace=True)
    m_fare = np.median(df[df['Fare'].isna()==False]['Fare'])
    df['Fare'].fillna(m_fare,inplace=True)
    return df
```

In [10]:

```
def encoding_data(df):
    df = pd.get_dummies(data=df,columns=['Sex','Embarked'])
    return df
```

In [11]:

```
df_cleaned_tr = cleaning_data(df_train)
```

In [12]:

```
df_cleaned_test = cleaning_data(df_test)
```

In [13]:

```
df_encode_tr = encoding_data(df_cleaned_tr)
df_encode_test = encoding_data(df_cleaned_test)
```

Pipeline

In [14]:

```
logit_p1 = Pipeline([('scaling',StandardScaler()),
                     ('pca',PCA(n_components=3)),
                     ('model_logit',LogisticRegression())])
```

In [15]:

```
dtree_p1 = Pipeline([('scaling',StandardScaler()),
                     ('pca',PCA(n_components=3)),
                     ('model_dtree',DecisionTreeClassifier())])
```

In [16]:

```
naive_p1 = Pipeline([('scaling',StandardScaler()),
                     ('pca',PCA(n_components=3)),
                     ('model_bayes',GaussianNB())])
```

In [17]:

```
x = df_encode_tr.drop('Survived',axis=1)
```

In [18]:

```
y = df_encode_tr['Survived']
```

In [19]:

```
x_tr,x_test,y_tr,y_test = train_test_split(x,y,test_size=0.2,random_state=100)
```

In [20]:

```
my_pipeline = [logit_p1,dtree_p1,naive_p1]
```

In [21]:

```
pipeline_dict = {0:'Logistic_Regression',1:'Decision_Tree',2:'Naive_Bayes'}
```

Training and Testing

In [22]:

```
for i in my_pipeline:
    i.fit(x_tr,y_tr)
```

In [23]:

```
for i,model in enumerate(my_pipeline):
    print(f"{pipeline_dict[i]}'s training accuracy is : {model.score(x_tr,y_tr)}")
```

```
Logistic_Regression's training accuracy is : 0.7851123595505618
Decision_Tree's training accuracy is : 0.9873595505617978
Naive_Bayes's training accuracy is : 0.7598314606741573
```

In [24]:

```
for i,model in enumerate(my_pipeline):
    print(f"{pipeline_dict[i]}'s testing accuracy is : {model.score(x_test,y_test)}")
```

```
Logistic_Regression's testing accuracy is : 0.776536312849162
Decision_Tree's testing accuracy is : 0.7430167597765364
Naive_Bayes's testing accuracy is : 0.770949720670391
```

Practical : 7

Aim : Perform classification using ml pipeline with pyspark in databricks

Load required libraries

In [0]:

```
from pyspark.ml import Pipeline
from pyspark.ml.feature import StringIndexer,OneHotEncoder,VectorAssembler
```

Load csv file

In [0]:

```
df_ipl = spark.read.option("inferSchema",True).option("header",True)
.csv("/FileStore/tables/ind_ban_comment.csv")
```

To print the summary of the columns

In [0]:

```
df_ipl.show()
```

```
+-------+-----------------+------+------------------+--------------------+------+---------+---+------+-
---------+--------+----+----+-------------------+
|Batsman|     Batsman_Name|Bowler|       Bowler_Name|          Commentary|Detail|Dismissed| Id|Isball|I
sboundary|Iswicket|Over|Runs|          Timestamp|
+-------+-----------------+------+------------------+--------------------+------+---------+---+------+-
---------+--------+----+----+-------------------+
|  28994|   Mohammed Shami| 63881| Mustafizur Rahman|OUT! Bowled! 5-fe...|     W|    28994|346|  true|
null|       1|49.6|   0|2019-07-02 13:18:47|
|   5132|Bhuvneshwar Kumar| 63881| Mustafizur Rahman|WIDE AND RUN OUT!...|  W+wd|     5132|344|  true|
null|       1|49.6|   1|2019-07-02 13:17:28|
|  28994|   Mohammed Shami| 63881| Mustafizur Rahman|Back of a length ...|  null|     null|343|  true|
null|    null|49.5|   1|2019-07-02 13:16:03|
|   5132|Bhuvneshwar Kumar| 63881| Mustafizur Rahman|Just 1 run off th...|  null|     null|342|  true|
null|    null|49.4|   1|2019-07-02 13:15:17|
|   3676|         MS Dhoni| 63881| Mustafizur Rahman|OUT! No Dhoni mag...|     W|     3676|340|  true|
null|       1|49.3|   0|2019-07-02 13:13:39|
|   3676|         MS Dhoni| 63881| Mustafizur Rahman|Another dot. Bang...|  null|     null|339|  true|
null|    null|49.2|   0|2019-07-02 13:12:47|
|   3676|         MS Dhoni| 63881| Mustafizur Rahman|Good length ball ...|  null|     null|338|  true|
null|    null|49.1|   0|2019-07-02 13:12:21|
|   3676|         MS Dhoni| 64151|Mohammad Saifuddin|Good length ball ...|  null|     null|337|  true|
null|    null|48.6|   1|2019-07-02 13:10:58|
|   3676|         MS Dhoni| 64151|Mohammad Saifuddin|FOUR! Dhoni rolli...|  null|     null|336|  true|
1|    null|48.5|   4|2019-07-02 13:10:12|
|   3676|         MS Dhoni| 64151|Mohammad Saifuddin|Slower delivery o...|  null|     null|335|  true|
null|    null|48.4|   0|2019-07-02 13:09:46|
|   3676|         MS Dhoni| 64151|Mohammad Saifuddin|Fuller on off, Dh...|  null|     null|334|  true|
null|    null|48.3|   0|2019-07-02 13:09:06|
|   3676|         MS Dhoni| 64151|Mohammad Saifuddin|FOUR! Driven with...|  null|     null|333|  true|
1|    null|48.2|   4|2019-07-02 13:08:18|
|   3676|         MS Dhoni| 64151|Mohammad Saifuddin|Good length ball ...|  null|     null|332|  true|
null|    null|48.1|   2|2019-07-02 13:07:42|
|   5132|Bhuvneshwar Kumar| 63881| Mustafizur Rahman|Slower bouncer to...|  null|     null|331|  true|
null|    null|47.6|   0|2019-07-02 13:06:42|
|   3676|         MS Dhoni| 63881| Mustafizur Rahman|Length delivery a...|  null|     null|330|  true|
null|    null|47.5|   1|2019-07-02 13:06:15|
|   3676|         MS Dhoni| 63881| Mustafizur Rahman|Good length ball ...|  null|     null|329|  true|
null|    null|47.4|   0|2019-07-02 13:05:32|
|   5132|Bhuvneshwar Kumar| 63881| Mustafizur Rahman|Good length ball ...|  null|     null|328|  true|
null|    null|47.3|   1|2019-07-02 13:04:58|
|   3632|   Dinesh Karthik| 63881| Mustafizur Rahman|OUT! Caught! Kart...|     W|     3632|326|  true|
null|       1|47.2|   0|2019-07-02 13:03:29|
|   3676|         MS Dhoni| 63881| Mustafizur Rahman|On the pads, Dhon...|  null|     null|325|  true|
null|    null|47.1|   1|2019-07-02 13:02:17|
|   3676|         MS Dhoni| 64151|Mohammad Saifuddin|Almost a calamity...|  null|     null|324|  true|
null|    null|46.6|   1|2019-07-02 13:01:16|
+-------+-----------------+------+------------------+--------------------+------+---------+---+------+-
---------+--------+----+----+-------------------+
only showing top 20 rows
```

In [0]:

```
df_ipl.printSchema()
```

```
root
 |-- Batsman: integer (nullable = true)
 |-- Batsman_Name: string (nullable = true)
 |-- Bowler: integer (nullable = true)
 |-- Bowler_Name: string (nullable = true)
 |-- Commentary: string (nullable = true)
 |-- Detail: string (nullable = true)
 |-- Dismissed: integer (nullable = true)
 |-- Id: integer (nullable = true)
 |-- Isball: boolean (nullable = true)
 |-- Isboundary: integer (nullable = true)
 |-- Iswicket: integer (nullable = true)
 |-- Over: double (nullable = true)
 |-- Runs: integer (nullable = true)
 |-- Timestamp: timestamp (nullable = true)
```

In [0]:

```
df_ipl_mod = StringIndexer(inputCol="Batsman_Name",outputCol="bt_idx")
.fit(df_ipl).transform(df_ipl)
```

In [0]:

```
df_ipl_mod.select("Batsman_name","bt_idx").show()
```

```
+-----------------+------+
|     Batsman_name|bt_idx|
+-----------------+------+
|   Mohammed Shami|  18.0|
|Bhuvneshwar Kumar|  16.0|
|   Mohammed Shami|  18.0|
|Bhuvneshwar Kumar|  16.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|Bhuvneshwar Kumar|  16.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
|Bhuvneshwar Kumar|  16.0|
|   Dinesh Karthik|  13.0|
|         MS Dhoni|   7.0|
|         MS Dhoni|   7.0|
+-----------------+------+
only showing top 20 rows
```

In [0]:

```
df_ipl.columns
```

```
Out[11]: ['Batsman',
 'Batsman_Name',
 'Bowler',
 'Bowler_Name',
 'Commentary',
 'Detail',
 'Dismissed',
 'Id',
 'Isball',
 'Isboundary',
 'Iswicket',
 'Over',
 'Runs',
 'Timestamp']
```

In [0]:

```
df_ipl = df_ipl.drop(*["Batsman","Bowler","Id"])
```

In [0]:

```
stage2 = StringIndexer(inputCol='Batsman_Name',outputCol='bm_index')
stage3 = StringIndexer(inputCol='Bowler_Name',outputCol='bowler_index')
stage4 = StringIndexer(inputCols=['bm_index','bowler_index'],
                       outputCols=['bm_ohe','bowler_ohe'])
```

In [0]:

```
stage5 = VectorAssembler(inputCols=['Dismissed','Isball','Isboundary','Over',
                                    'Runs','bm_ohe','bowler_ohe'],outputCol="features")
```

In [0]:

```
from pyspark.ml.classification import LogisticRegression
stage6 = LogisticRegression(featuresCol="features",labelCol="Iswicket")
```

In [0]:

```
pl = Pipeline(stages=[stage2,stage3,stage4,stage5,stage6])
```

In [0]:

```
df_ipl = df_ipl.fillna(0)
```

In [0]:

```
logit_model = pl.fit(df_ipl).transform(df_ipl)
```

In [0]:

```
logit_model.select("features","Iswicket","prediction").show()
```

```
+--------------------+--------+----------+
|            features|Iswicket|prediction|
+--------------------+--------+----------+
|[28994.0,1.0,0.0,...|       1|       1.0|
|[5132.0,1.0,0.0,4...|       1|       1.0|
|[0.0,1.0,0.0,49.5...|       0|       0.0|
|[0.0,1.0,0.0,49.4...|       0|       0.0|
|[3676.0,1.0,0.0,4...|       1|       1.0|
|(7,[1,3,5],[1.0,4...|       0|       0.0|
|(7,[1,3,5],[1.0,4...|       0|       0.0|
|[0.0,1.0,0.0,48.6...|       0|       0.0|
|[0.0,1.0,1.0,48.5...|       0|       0.0|
|[0.0,1.0,0.0,48.4...|       0|       0.0|
|[0.0,1.0,0.0,48.3...|       0|       0.0|
|[0.0,1.0,1.0,48.2...|       0|       0.0|
|[0.0,1.0,0.0,48.1...|       0|       0.0|
|(7,[1,3,5],[1.0,4...|       0|       0.0|
|[0.0,1.0,0.0,47.5...|       0|       0.0|
|(7,[1,3,5],[1.0,4...|       0|       0.0|
|[0.0,1.0,0.0,47.3...|       0|       0.0|
|[3632.0,1.0,0.0,4...|       1|       1.0|
|[0.0,1.0,0.0,47.1...|       0|       0.0|
|[0.0,1.0,0.0,46.6...|       0|       0.0|
+--------------------+--------+----------+
only showing top 20 rows
```

In [0]:

```
logit_model.columns
```

```
Out[21]: ['Batsman_Name',
 'Bowler_Name',
 'Commentary',
 'Detail',
 'Dismissed',
 'Isball',
 'Isboundary',
 'Iswicket',
 'Over',
 'Runs',
 'Timestamp',
 'bm_index',
 'bowler_index',
 'bm_ohe',
 'bowler_ohe',
 'features',
 'rawPrediction',
 'probability',
 'prediction']
```

In [0]:

```
pl1 = Pipeline(stages=[stage2,stage3,stage4])
```

In [0]:

```
df_mod = pl1.fit(df_ipl).transform(df_ipl)
```

In [0]:

```
df_mod.show()
```

```
+-----------------+-----------------+--------------------+------+---------+------+----------+--------+
----+----+------------------+--------+------------+------+----------+
|     Batsman_Name|       Bowler_Name|          Commentary|Detail|Dismissed|Isball|Isboundary|Iswicket|
Over|Runs|          Timestamp|bm_index|bowler_index|bm_ohe|bowler_ohe|
+-----------------+-----------------+--------------------+------+---------+------+----------+--------+
----+----+------------------+--------+------------+------+----------+
|   Mohammed Shami| Mustafizur Rahman|OUT! Bowled! 5-fe...|     W|    28994|  true|         0|       1|
49.6|   0|2019-07-02 13:18:47|    18.0|         0.0|  18.0|       0.0|
|Bhuvneshwar Kumar| Mustafizur Rahman|WIDE AND RUN OUT!...|  W+wd|     5132|  true|         0|       1|
49.6|   1|2019-07-02 13:17:28|    16.0|         0.0|  16.0|       0.0|
|   Mohammed Shami| Mustafizur Rahman|Back of a length ...|  null|        0|  true|         0|       0|
49.5|   1|2019-07-02 13:16:03|    18.0|         0.0|  18.0|       0.0|
|Bhuvneshwar Kumar| Mustafizur Rahman|Just 1 run off th...|  null|        0|  true|         0|       0|
49.4|   1|2019-07-02 13:15:17|    16.0|         0.0|  16.0|       0.0|
|         MS Dhoni| Mustafizur Rahman|OUT! No Dhoni mag...|     W|     3676|  true|         0|       1|
49.3|   0|2019-07-02 13:13:39|     7.0|         0.0|   7.0|       0.0|
|         MS Dhoni| Mustafizur Rahman|Another dot. Bang...|  null|        0|  true|         0|       0|
49.2|   0|2019-07-02 13:12:47|     7.0|         0.0|   7.0|       0.0|
|         MS Dhoni| Mustafizur Rahman|Good length ball ...|  null|        0|  true|         0|       0|
49.1|   0|2019-07-02 13:12:21|     7.0|         0.0|   7.0|       0.0|
|         MS Dhoni|Mohammad Saifuddin|Good length ball ...|  null|        0|  true|         0|       0|
48.6|   1|2019-07-02 13:10:58|     7.0|         8.0|   7.0|       8.0|
|         MS Dhoni|Mohammad Saifuddin|FOUR! Dhoni rolli...|  null|        0|  true|         1|       0|
48.5|   4|2019-07-02 13:10:12|     7.0|         8.0|   7.0|       8.0|
|         MS Dhoni|Mohammad Saifuddin|Slower delivery o...|  null|        0|  true|         0|       0|
48.4|   0|2019-07-02 13:09:46|     7.0|         8.0|   7.0|       8.0|
|         MS Dhoni|Mohammad Saifuddin|Fuller on off, Dh...|  null|        0|  true|         0|       0|
48.3|   0|2019-07-02 13:09:06|     7.0|         8.0|   7.0|       8.0|
|         MS Dhoni|Mohammad Saifuddin|FOUR! Driven with...|  null|        0|  true|         1|       0|
48.2|   4|2019-07-02 13:08:18|     7.0|         8.0|   7.0|       8.0|
|         MS Dhoni|Mohammad Saifuddin|Good length ball ...|  null|        0|  true|         0|       0|
48.1|   2|2019-07-02 13:07:42|     7.0|         8.0|   7.0|       8.0|
|Bhuvneshwar Kumar| Mustafizur Rahman|Slower bouncer to...|  null|        0|  true|         0|       0|
47.6|   0|2019-07-02 13:06:42|    16.0|         0.0|  16.0|       0.0|
|         MS Dhoni| Mustafizur Rahman|Length delivery a...|  null|        0|  true|         0|       0|
47.5|   1|2019-07-02 13:06:15|     7.0|         0.0|   7.0|       0.0|
|         MS Dhoni| Mustafizur Rahman|Good length ball ...|  null|        0|  true|         0|       0|
47.4|   0|2019-07-02 13:05:32|     7.0|         0.0|   7.0|       0.0|
|Bhuvneshwar Kumar| Mustafizur Rahman|Good length ball ...|  null|        0|  true|         0|       0|
47.3|   1|2019-07-02 13:04:58|    16.0|         0.0|  16.0|       0.0|
|   Dinesh Karthik| Mustafizur Rahman|OUT! Caught! Kart...|     W|     3632|  true|         0|       1|
47.2|   0|2019-07-02 13:03:29|    13.0|         0.0|  13.0|       0.0|
|         MS Dhoni| Mustafizur Rahman|On the pads, Dhon...|  null|        0|  true|         0|       0|
47.1|   1|2019-07-02 13:02:17|     7.0|         0.0|   7.0|       0.0|
|         MS Dhoni|Mohammad Saifuddin|Almost a calamity...|  null|        0|  true|         0|       0|
46.6|   1|2019-07-02 13:01:16|     7.0|         8.0|   7.0|       8.0|
+-----------------+-----------------+--------------------+------+---------+------+----------+--------+
----+----+------------------+--------+------------+------+----------+
only showing top 20 rows
```

In [0]:

```
df_vec = VectorAssembler(inputCols=['Dismissed','Isball','Isboundary',
                                    'Over','Runs','bm_ohe','bowler_ohe'],
                         outputCol="features").transform(df_mod)
```

In [0]:

```
df_vec.show()
```

```
+-----------------+-----------------+-------------------+------+---------+------+----------+--------+
----+----+-------------------+--------+------------+------+----------+--------------------+
|     Batsman_Name|       Bowler_Name|          Commentary|Detail|Dismissed|Isball|Isboundary|Iswicket|
Over|Runs|          Timestamp|bm_index|bowler_index|bm_ohe|bowler_ohe|            features|
+-----------------+-----------------+-------------------+------+---------+------+----------+--------+
----+----+-------------------+--------+------------+------+----------+--------------------+
|   Mohammed Shami| Mustafizur Rahman|OUT! Bowled! 5-fe...|     W|    28994|  true|         0|       1|
49.6|   0|2019-07-02 13:18:47|    18.0|         0.0|  18.0|       0.0|[28994.0,1.0,0.0,...|
|Bhuvneshwar Kumar| Mustafizur Rahman|WIDE AND RUN OUT!...|  W+wd|     5132|  true|         0|       1|
49.6|   1|2019-07-02 13:17:28|    16.0|         0.0|  16.0|       0.0|[5132.0,1.0,0.0,4...|
|   Mohammed Shami| Mustafizur Rahman|Back of a length ...|  null|        0|  true|         0|       0|
49.5|   1|2019-07-02 13:16:03|    18.0|         0.0|  18.0|       0.0|[0.0,1.0,0.0,49.5...|
|Bhuvneshwar Kumar| Mustafizur Rahman|Just 1 run off th...|  null|        0|  true|         0|       0|
49.4|   1|2019-07-02 13:15:17|    16.0|         0.0|  16.0|       0.0|[0.0,1.0,0.0,49.4...|
|         MS Dhoni| Mustafizur Rahman|OUT! No Dhoni mag...|     W|     3676|  true|         0|       1|
49.3|   0|2019-07-02 13:13:39|     7.0|         0.0|   7.0|       0.0|[3676.0,1.0,0.0,4...|
|         MS Dhoni| Mustafizur Rahman|Another dot. Bang...|  null|        0|  true|         0|       0|
49.2|   0|2019-07-02 13:12:47|     7.0|         0.0|   7.0|       0.0|(7,[1,3,5],[1.0,4...|
|         MS Dhoni| Mustafizur Rahman|Good length ball ...|  null|        0|  true|         0|       0|
49.1|   0|2019-07-02 13:12:21|     7.0|         0.0|   7.0|       0.0|(7,[1,3,5],[1.0,4...|
|         MS Dhoni|Mohammad Saifuddin|Good length ball ...|  null|        0|  true|         0|       0|
48.6|   1|2019-07-02 13:10:58|     7.0|         8.0|   7.0|       8.0|[0.0,1.0,0.0,48.6...|
|         MS Dhoni|Mohammad Saifuddin|FOUR! Dhoni rolli...|  null|        0|  true|         1|       0|
48.5|   4|2019-07-02 13:10:12|     7.0|         8.0|   7.0|       8.0|[0.0,1.0,1.0,48.5...|
|         MS Dhoni|Mohammad Saifuddin|Slower delivery o...|  null|        0|  true|         0|       0|
48.4|   0|2019-07-02 13:09:46|     7.0|         8.0|   7.0|       8.0|[0.0,1.0,0.0,48.4...|
|         MS Dhoni|Mohammad Saifuddin|Fuller on off, Dh...|  null|        0|  true|         0|       0|
48.3|   0|2019-07-02 13:09:06|     7.0|         8.0|   7.0|       8.0|[0.0,1.0,0.0,48.3...|
|         MS Dhoni|Mohammad Saifuddin|FOUR! Driven with...|  null|        0|  true|         1|       0|
48.2|   4|2019-07-02 13:08:18|     7.0|         8.0|   7.0|       8.0|[0.0,1.0,1.0,48.2...|
|         MS Dhoni|Mohammad Saifuddin|Good length ball ...|  null|        0|  true|         0|       0|
48.1|   2|2019-07-02 13:07:42|     7.0|         8.0|   7.0|       8.0|[0.0,1.0,0.0,48.1...|
|Bhuvneshwar Kumar| Mustafizur Rahman|Slower bouncer to...|  null|        0|  true|         0|       0|
47.6|   0|2019-07-02 13:06:42|    16.0|         0.0|  16.0|       0.0|(7,[1,3,5],[1.0,4...|
|         MS Dhoni| Mustafizur Rahman|Length delivery a...|  null|        0|  true|         0|       0|
47.5|   1|2019-07-02 13:06:15|     7.0|         0.0|   7.0|       0.0|[0.0,1.0,0.0,47.5...|
|         MS Dhoni| Mustafizur Rahman|Good length ball ...|  null|        0|  true|         0|       0|
47.4|   0|2019-07-02 13:05:32|     7.0|         0.0|   7.0|       0.0|(7,[1,3,5],[1.0,4...|
|Bhuvneshwar Kumar| Mustafizur Rahman|Good length ball ...|  null|        0|  true|         0|       0|
47.3|   1|2019-07-02 13:04:58|    16.0|         0.0|  16.0|       0.0|[0.0,1.0,0.0,47.3...|
|   Dinesh Karthik| Mustafizur Rahman|OUT! Caught! Kart...|     W|     3632|  true|         0|       1|
47.2|   0|2019-07-02 13:03:29|    13.0|         0.0|  13.0|       0.0|[3632.0,1.0,0.0,4...|
|         MS Dhoni| Mustafizur Rahman|On the pads, Dhon...|  null|        0|  true|         0|       0|
47.1|   1|2019-07-02 13:02:17|     7.0|         0.0|   7.0|       0.0|[0.0,1.0,0.0,47.1...|
|         MS Dhoni|Mohammad Saifuddin|Almost a calamity...|  null|        0|  true|         0|       0|
46.6|   1|2019-07-02 13:01:16|     7.0|         8.0|   7.0|       8.0|[0.0,1.0,0.0,46.6...|
+-----------------+-----------------+-------------------+------+---------+------+----------+--------+
----+----+-------------------+--------+------------+------+----------+--------------------+
only showing top 20 rows
```

In [0]:

```
df_vector = df_vec.select("features","Iswicket")
```

In [0]:

```
splits = df_vector.randomSplit([.7,.3])
```

In [0]:

```
df_train = splits[0]
df_test = splits[1]
```

In [0]:

```
df_train.show(10)
```

```
+--------------------+--------+
|            features|Iswicket|
+--------------------+--------+
| (7,[1,3],[1.0,2.2])|       0|
| (7,[1,3],[1.0,4.1])|       0|
| (7,[1,3],[1.0,4.2])|       0|
| (7,[1,3],[1.0,6.2])|       0|
| (7,[1,3],[1.0,6.6])|       0|
| (7,[1,3],[1.0,8.3])|       0|
| (7,[1,3],[1.0,8.4])|       0|
|(7,[1,3],[1.0,23.4])|       0|
|(7,[1,3,4],[1.0,2...|       0|
|(7,[1,3,4],[1.0,4...|       0|
+--------------------+--------+
only showing top 10 rows
```

In [0]:

```
logit_mod_tr = LogisticRegression(featuresCol="features",
                                  labelCol="Iswicket").fit(df_train).transform(df_train)
```

In [0]:

```
logit_mod_test = LogisticRegression(featuresCol="features",
                                    labelCol="Iswicket").fit(df_train).transform(df_test)
```

In [0]:

```
logit_mod_test.select(["features","Iswicket","prediction"]).show()
```

```
+--------------------+--------+----------+
|            features|Iswicket|prediction|
+--------------------+--------+----------+
| (7,[1,3],[1.0,2.3])|       0|       0.0|
| (7,[1,3],[1.0,2.4])|       0|       0.0|
| (7,[1,3],[1.0,4.3])|       0|       0.0|
|(7,[1,3,4],[1.0,4...|       0|       0.0|
|(7,[1,3,4],[1.0,8...|       0|       0.0|
|(7,[1,3,4],[1.0,2...|       0|       0.0|
|(7,[1,3,5],[1.0,3...|       0|       0.0|
|(7,[1,3,6],[1.0,0...|       0|       0.0|
|(7,[1,3,6],[1.0,5...|       0|       0.0|
|(7,[1,3,6],[1.0,1...|       0|       0.0|
|(7,[1,3,6],[1.0,1...|       0|       0.0|
|(7,[1,3,6],[1.0,1...|       0|       0.0|
|(7,[1,3,6],[1.0,1...|       0|       0.0|
|(7,[1,3,6],[1.0,1...|       0|       0.0|
|(7,[1,3,6],[1.0,2...|       0|       0.0|
|(7,[1,3,6],[1.0,2...|       0|       0.0|
|[0.0,1.0,0.0,0.2,...|       0|       0.0|
|[0.0,1.0,0.0,0.2,...|       0|       0.0|
|[0.0,1.0,0.0,0.5,...|       0|       0.0|
|[0.0,1.0,0.0,0.6,...|       0|       0.0|
+--------------------+--------+----------+
only showing top 20 rows
```

Evaluate the classifier

In [0]:

```
from pyspark.ml.evaluation import BinaryClassificationEvaluator
```

In [0]:

```
eval = BinaryClassificationEvaluator(labelCol="Iswicket",rawPredictionCol="prediction",metricName="areaUnderROC")
```

In [0]:

```
eval.evaluate(logit_mod_test)
```

Out[38]: 1.0

Practical : 8

Aim : Perform regression using ml pipeline with pyspark in databricks on US cars dataset

Import Libraries

In [0]:

```
import pyspark
import pandas as pd
from pyspark import SparkContext,SparkConf
from pyspark.sql import SparkSession
```

In [0]:

```
spark_new = SparkSession.builder.master("local[2]").getOrCreate()
```

Load Dataset

In [0]:

```
data = spark_new.read.option("header",True).option("inferschema",True).csv("/FileStore/tables/USA_cars_datasets.csv")
```

In [0]:

```
from pyspark.ml.feature import OneHotEncoder,StringIndexer
string_index = StringIndexer(inputCols=["brand","model","color","state"],outputCols=["brand_idx","model_idx","color_idx","stat
data = string_index.fit(data).transform(data)
```

Explore Dataset

In [0]:

```
data.show()
```

```
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+
|_c0|price|    brand|  model|year| title_status| mileage| color|               vin|      lot|         state|co
untry|    condition|brand_idx|model_idx|color_idx|state_idx|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+
|  0| 6300|   toyota|cruiser|2008|clean vehicle|274117.0| black|  jtezu11f88k007763|159348797|    new jersey|
usa| 10 days left|     27.0|     89.0|      1.0|     10.0|
|  1| 2899|     ford|     se|2011|clean vehicle|190552.0|silver|  2fmdk3gc4bbb02217|166951262|     tennessee|
usa|  6 days left|      0.0|    113.0|      3.0|     21.0|
|  2| 5350|    dodge|    mpv|2018|clean vehicle| 39590.0|silver|  3c4pdcgg5jt346413|167655728|       georgia|
usa|  2 days left|      1.0|      4.0|      3.0|     15.0|
|  3|25000|     ford|   door|2014|clean vehicle| 64146.0|  blue|  1ftfw1et4efc23745|167753855|      virginia|
usa|22 hours left|      0.0|      0.0|      5.0|      9.0|
|  4|27700|chevrolet|   1500|2018|clean vehicle|  6654.0|   red|  3gcpcrec2jg473991|167763266|       florida|
usa|22 hours left|      3.0|     15.0|      4.0|      1.0|
|  5| 5700|    dodge|    mpv|2018|clean vehicle| 45561.0| white|  2c4rdgeg9jr237989|167655771|         texas|
usa|  2 days left|      1.0|      4.0|      0.0|      2.0|
|  6| 7300|chevrolet|     pk|2010|clean vehicle|149050.0| black|  1gcsksea1az121133|167753872|       georgia|
usa|22 hours left|      3.0|     54.0|      1.0|     15.0|
|  7|13350|      gmc|   door|2017|clean vehicle| 23525.0|  gray|  1gks2gkc3hr326762|167692494|    california|
usa|20 hours left|      4.0|      0.0|      2.0|      3.0|
|  8|14600|chevrolet| malibu|2018|clean vehicle|  9371.0|silver|  1g1zd5st5jf191860|167763267|       florida|
usa|22 hours left|      3.0|     36.0|      3.0|      1.0|
|  9| 5250|     ford|    mpv|2017|clean vehicle| 63418.0| black|  2fmpk3j92hbc12542|167656121|         texas|
usa|  2 days left|      0.0|      4.0|      1.0|      2.0|
| 10|10400|    dodge|  coupe|2009|clean vehicle|107856.0|orange|  2b3lj54t49h509675|167753874|       georgia|
usa|22 hours left|      1.0|     49.0|      8.0|     15.0|
| 11|12920|      gmc|    mpv|2017|clean vehicle| 39650.0| white|  1gks2bkc6hr136280|167692496|    california|
usa|20 hours left|      4.0|      4.0|      0.0|      3.0|
| 12|31900|chevrolet|   1500|2018|clean vehicle| 22909.0| black|  3gcukrec0jg176059|167763273|     tennessee|
usa|22 hours left|      3.0|     15.0|      1.0|     21.0|
| 13| 5430| chrysler|  wagon|2017|clean vehicle|138650.0|  gray|  2c4rc1cg5hr616095|167656123|         texas|
usa|  2 days left|      6.0|     22.0|      2.0|      2.0|
| 14|20700|     ford|   door|2013|clean vehicle|100757.0| black|  1ftfw1et7dfa47790|167753916|      virginia|
usa|22 hours left|      0.0|      0.0|      1.0|      9.0|
| 15|12710|      gmc|   door|2017|clean vehicle| 25747.0| white|  1gks2gkc6hr328389|167692497|    california|
usa|20 hours left|      4.0|      0.0|      0.0|      3.0|
| 16| 5200|      kia|  forte|2018|clean vehicle| 46194.0|  blue|  3kpfk4a77je198723|167801757|north carolina|
usa|  2 days left|     10.0|     77.0|      5.0|      5.0|
| 17|16500|    buick| encore|2018|clean vehicle| 20002.0|   red|  kl4cj1sb6jb688844|167763275|     tennessee|
usa|22 hours left|      9.0|     65.0|      4.0|     21.0|
| 18| 5210|     ford|    mpv|2017|clean vehicle| 35714.0| white|  2fmpk3j95hbb73607|167656124|         texas|
usa|  2 days left|      0.0|      4.0|      0.0|      2.0|
| 19|38100|     ford|   door|2013|clean vehicle| 54380.0|  gray|  1ft8w3dt5deb68569|167753923|      virginia|
usa|22 hours left|      0.0|      0.0|      2.0|      9.0|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+
only showing top 20 rows
```

In [0]:

```
ohe_mod = OneHotEncoder(inputCols=["brand_idx","model_idx","color_idx",
                                   "state_idx"],outputCols=["brand_ohe",
                                                            "model_ohe",
                                                            "color_ohe","state_ohe"])
data = ohe_mod.fit(data).transform(data)
```

In [0]:

```
data.show()
```

```
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+--------------+-----------------+-------------+--
-------------+
|_c0|price|    brand|  model|year| title_status| mileage| color|               vin|      lot|         state|co
untry|    condition|brand_idx|model_idx|color_idx|state_idx|      brand_ohe|        model_ohe|     color_ohe|
state_ohe|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+--------------+-----------------+-------------+--
-------------+
|  0| 6300|   toyota|cruiser|2008|clean vehicle|274117.0| black|  jtezu11f88k007763|159348797|    new jersey|
usa| 10 days left|     27.0|     89.0|      1.0|     10.0|     (27,[],[])| (126,[89],[1.0])|(48,[1],[1.0])|(43,
[10],[1.0])|
|  1| 2899|     ford|     se|2011|clean vehicle|190552.0|silver|  2fmdk3gc4bbb02217|166951262|     tennessee|
usa|  6 days left|      0.0|    113.0|      3.0|     21.0| (27,[0],[1.0])|(126,[113],[1.0])|(48,[3],[1.0])|(43,
[21],[1.0])|
|  2| 5350|    dodge|    mpv|2018|clean vehicle| 39590.0|silver|  3c4pdcgg5jt346413|167655728|       georgia|
usa|  2 days left|      1.0|      4.0|      3.0|     15.0| (27,[1],[1.0])|  (126,[4],[1.0])|(48,[3],[1.0])|(43,
[15],[1.0])|
|  3|25000|     ford|   door|2014|clean vehicle| 64146.0|  blue|  1ftfw1et4efc23745|167753855|      virginia|
usa|22 hours left|      0.0|      0.0|      5.0|      9.0| (27,[0],[1.0])|  (126,[0],[1.0])|(48,[5],[1.0])| (4
3,[9],[1.0])|
|  4|27700|chevrolet|   1500|2018|clean vehicle|  6654.0|   red|  3gcpcrec2jg473991|167763266|       florida|
usa|22 hours left|      3.0|     15.0|      4.0|      1.0| (27,[3],[1.0])| (126,[15],[1.0])|(48,[4],[1.0])| (4
3,[1],[1.0])|
|  5| 5700|    dodge|    mpv|2018|clean vehicle| 45561.0| white|  2c4rdgeg9jr237989|167655771|         texas|
usa|  2 days left|      1.0|      4.0|      0.0|      2.0| (27,[1],[1.0])|  (126,[4],[1.0])|(48,[0],[1.0])| (4
3,[2],[1.0])|
|  6| 7300|chevrolet|     pk|2010|clean vehicle|149050.0| black|  1gcsksea1az121133|167753872|       georgia|
usa|22 hours left|      3.0|     54.0|      1.0|     15.0| (27,[3],[1.0])| (126,[54],[1.0])|(48,[1],[1.0])|(43,
[15],[1.0])|
|  7|13350|      gmc|   door|2017|clean vehicle| 23525.0|  gray|  1gks2gkc3hr326762|167692494|    california|
usa|20 hours left|      4.0|      0.0|      2.0|      3.0| (27,[4],[1.0])|  (126,[0],[1.0])|(48,[2],[1.0])| (4
3,[3],[1.0])|
|  8|14600|chevrolet| malibu|2018|clean vehicle|  9371.0|silver|  1g1zd5st5jf191860|167763267|       florida|
usa|22 hours left|      3.0|     36.0|      3.0|      1.0| (27,[3],[1.0])| (126,[36],[1.0])|(48,[3],[1.0])| (4
3,[1],[1.0])|
|  9| 5250|     ford|    mpv|2017|clean vehicle| 63418.0| black|  2fmpk3j92hbc12542|167656121|         texas|
usa|  2 days left|      0.0|      4.0|      1.0|      2.0| (27,[0],[1.0])|  (126,[4],[1.0])|(48,[1],[1.0])| (4
3,[2],[1.0])|
| 10|10400|    dodge|  coupe|2009|clean vehicle|107856.0|orange|  2b3lj54t49h509675|167753874|       georgia|
usa|22 hours left|      1.0|     49.0|      8.0|     15.0| (27,[1],[1.0])| (126,[49],[1.0])|(48,[8],[1.0])|(43,
[15],[1.0])|
| 11|12920|      gmc|    mpv|2017|clean vehicle| 39650.0| white|  1gks2bkc6hr136280|167692496|    california|
usa|20 hours left|      4.0|      4.0|      0.0|      3.0| (27,[4],[1.0])|  (126,[4],[1.0])|(48,[0],[1.0])| (4
3,[3],[1.0])|
| 12|31900|chevrolet|   1500|2018|clean vehicle| 22909.0| black|  3gcukrec0jg176059|167763273|     tennessee|
usa|22 hours left|      3.0|     15.0|      1.0|     21.0| (27,[3],[1.0])| (126,[15],[1.0])|(48,[1],[1.0])|(43,
[21],[1.0])|
| 13| 5430| chrysler|  wagon|2017|clean vehicle|138650.0|  gray|  2c4rc1cg5hr616095|167656123|         texas|
usa|  2 days left|      6.0|     22.0|      2.0|      2.0| (27,[6],[1.0])| (126,[22],[1.0])|(48,[2],[1.0])| (4
3,[2],[1.0])|
| 14|20700|     ford|   door|2013|clean vehicle|100757.0| black|  1ftfw1et7dfa47790|167753916|      virginia|
usa|22 hours left|      0.0|      0.0|      1.0|      9.0| (27,[0],[1.0])|  (126,[0],[1.0])|(48,[1],[1.0])| (4
3,[9],[1.0])|
| 15|12710|      gmc|   door|2017|clean vehicle| 25747.0| white|  1gks2gkc6hr328389|167692497|    california|
usa|20 hours left|      4.0|      0.0|      0.0|      3.0| (27,[4],[1.0])|  (126,[0],[1.0])|(48,[0],[1.0])| (4
3,[3],[1.0])|
| 16| 5200|      kia|  forte|2018|clean vehicle| 46194.0|  blue|  3kpfk4a77je198723|167801757|north carolina|
usa|  2 days left|     10.0|     77.0|      5.0|      5.0|(27,[10],[1.0])| (126,[77],[1.0])|(48,[5],[1.0])| (4
3,[5],[1.0])|
| 17|16500|    buick| encore|2018|clean vehicle| 20002.0|   red|  kl4cj1sb6jb688844|167763275|     tennessee|
usa|22 hours left|      9.0|     65.0|      4.0|     21.0| (27,[9],[1.0])| (126,[65],[1.0])|(48,[4],[1.0])|(43,
[21],[1.0])|
| 18| 5210|     ford|    mpv|2017|clean vehicle| 35714.0| white|  2fmpk3j95hbb73607|167656124|         texas|
usa|  2 days left|      0.0|      4.0|      0.0|      2.0| (27,[0],[1.0])|  (126,[4],[1.0])|(48,[0],[1.0])| (4
3,[2],[1.0])|
| 19|38100|     ford|   door|2013|clean vehicle| 54380.0|  gray|  1ft8w3dt5deb68569|167753923|      virginia|
usa|22 hours left|      0.0|      0.0|      2.0|      9.0| (27,[0],[1.0])|  (126,[0],[1.0])|(48,[2],[1.0])| (4
3,[9],[1.0])|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+--------------+-----------------+-------------+--
-------------+
only showing top 20 rows
```

In [0]:

```
data.printSchema()
```

```
root
 |-- _c0: integer (nullable = true)
 |-- price: integer (nullable = true)
 |-- brand: string (nullable = true)
 |-- model: string (nullable = true)
 |-- year: integer (nullable = true)
 |-- title_status: string (nullable = true)
 |-- mileage: double (nullable = true)
 |-- color: string (nullable = true)
 |-- vin: string (nullable = true)
 |-- lot: integer (nullable = true)
 |-- state: string (nullable = true)
 |-- country: string (nullable = true)
 |-- condition: string (nullable = true)
```

In [0]:

```
data.show()
```

```
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+
|_c0|price|    brand|  model|year| title_status| mileage| color|               vin|      lot|         state|co
untry|    condition|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+
|  0| 6300|   toyota|cruiser|2008|clean vehicle|274117.0| black|  jtezu11f88k007763|159348797|    new jersey|
usa| 10 days left|
|  1| 2899|     ford|     se|2011|clean vehicle|190552.0|silver|  2fmdk3gc4bbb02217|166951262|     tennessee|
usa|  6 days left|
|  2| 5350|    dodge|    mpv|2018|clean vehicle| 39590.0|silver|  3c4pdcgg5jt346413|167655728|       georgia|
usa|  2 days left|
|  3|25000|     ford|   door|2014|clean vehicle| 64146.0|  blue|  1ftfw1et4efc23745|167753855|      virginia|
usa|22 hours left|
|  4|27700|chevrolet|   1500|2018|clean vehicle|  6654.0|   red|  3gcpcrec2jg473991|167763266|       florida|
usa|22 hours left|
|  5| 5700|    dodge|    mpv|2018|clean vehicle| 45561.0| white|  2c4rdgeg9jr237989|167655771|         texas|
usa|  2 days left|
|  6| 7300|chevrolet|     pk|2010|clean vehicle|149050.0| black|  1gcsksea1az121133|167753872|       georgia|
usa|22 hours left|
|  7|13350|      gmc|   door|2017|clean vehicle| 23525.0|  gray|  1gks2gkc3hr326762|167692494|    california|
usa|20 hours left|
|  8|14600|chevrolet| malibu|2018|clean vehicle|  9371.0|silver|  1g1zd5st5jf191860|167763267|       florida|
usa|22 hours left|
|  9| 5250|     ford|    mpv|2017|clean vehicle| 63418.0| black|  2fmpk3j92hbc12542|167656121|         texas|
usa|  2 days left|
| 10|10400|    dodge|  coupe|2009|clean vehicle|107856.0|orange|  2b3lj54t49h509675|167753874|       georgia|
usa|22 hours left|
| 11|12920|      gmc|    mpv|2017|clean vehicle| 39650.0| white|  1gks2bkc6hr136280|167692496|    california|
usa|20 hours left|
| 12|31900|chevrolet|   1500|2018|clean vehicle| 22909.0| black|  3gcukrec0jg176059|167763273|     tennessee|
usa|22 hours left|
| 13| 5430| chrysler|  wagon|2017|clean vehicle|138650.0|  gray|  2c4rc1cg5hr616095|167656123|         texas|
usa|  2 days left|
| 14|20700|     ford|   door|2013|clean vehicle|100757.0| black|  1ftfw1et7dfa47790|167753916|      virginia|
usa|22 hours left|
| 15|12710|      gmc|   door|2017|clean vehicle| 25747.0| white|  1gks2gkc6hr328389|167692497|    california|
usa|20 hours left|
| 16| 5200|      kia|  forte|2018|clean vehicle| 46194.0|  blue|  3kpfk4a77je198723|167801757|north carolina|
usa|  2 days left|
| 17|16500|    buick| encore|2018|clean vehicle| 20002.0|   red|  kl4cj1sb6jb688844|167763275|     tennessee|
usa|22 hours left|
| 18| 5210|     ford|    mpv|2017|clean vehicle| 35714.0| white|  2fmpk3j95hbb73607|167656124|         texas|
usa|  2 days left|
| 19|38100|     ford|   door|2013|clean vehicle| 54380.0|  gray|  1ft8w3dt5deb68569|167753923|      virginia|
usa|22 hours left|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+
only showing top 20 rows
```

In [0]:

```
df_panda = data.toPandas()
```

In [0]:

```
df_panda.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2499 entries, 0 to 2498
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   _c0           2499 non-null   int32  
 1   price         2499 non-null   int32  
 2   brand         2499 non-null   object 
 3   model         2499 non-null   object 
 4   year          2499 non-null   int32  
 5   title_status  2499 non-null   object 
 6   mileage       2499 non-null   float64
 7   color         2499 non-null   object 
 8   vin           2499 non-null   object 
 9   lot           2499 non-null   int32  
 10  state         2499 non-null   object 
 11  country       2499 non-null   object 
 12  condition     2499 non-null   object 
dtypes: float64(1), int32(4), object(8)
memory usage: 214.9+ KB
```

In [0]:

```
df_panda.dtypes
```

```
Out[8]: _c0              int32
price             int32
brand            object
model            object
year              int32
title_status     object
mileage         float64
color            object
vin              object
lot               int32
state            object
country          object
condition        object
dtype: object
```

In [0]:

```
df_panda.isna().sum()
```

```
Out[9]: _c0             0
price           0
brand           0
model           0
year            0
title_status    0
mileage         0
color           0
vin             0
lot             0
state           0
country         0
condition       0
dtype: int64
```

In [0]:

```
df_panda.describe()
```

	_c0	price	year	mileage	lot
count	2499.000000	2499.000000	2499.000000	2.499000e+03	2.499000e+03
mean	1249.000000	18767.671469	2016.714286	5.229869e+04	1.676914e+08
std	721.543484	12116.094936	3.442656	5.970552e+04	2.038772e+05
min	0.000000	0.000000	1973.000000	0.000000e+00	1.593488e+08
25%	624.500000	10200.000000	2016.000000	2.146650e+04	1.676253e+08
50%	1249.000000	16900.000000	2018.000000	3.536500e+04	1.677451e+08
75%	1873.500000	25555.500000	2019.000000	6.347250e+04	1.677798e+08
max	2498.000000	84900.000000	2020.000000	1.017936e+06	1.678055e+08

In [0]:

```
df_panda.columns
```

```
Out[11]: Index(['_c0', 'price', 'brand', 'model', 'year', 'title_status', 'mileage',
       'color', 'vin', 'lot', 'state', 'country', 'condition'],
      dtype='object')
```

In [0]:

```
## Check without encoding data i.e with non numeric columns
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols = ["brand_ohe","model_ohe","color_ohe","state_ohe",
                                         '_c0', 'price', 'brand', 'model',
                                         'year', 'title_status',
                                         'mileage',
                                         'color', 'vin', 'lot', 'state',
                                         'country', 'condition'], outputCol = 'feature_vec')
```

In [0]:

```
assembler = VectorAssembler(inputCols = [ "brand_ohe","model_ohe",
                                         "color_ohe","state_ohe",'year',
                                         'mileage','lot'], outputCol = 'feature_vector')
```

In [0]:

```
data_new = assembler.transform(data)
```

In [0]:

```
data_new.show()
```

```
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+--------------+-----------------+-------------+--
-------------+--------------------+
|_c0|price|    brand|  model|year| title_status| mileage| color|               vin|      lot|         state|co
untry|    condition|brand_idx|model_idx|color_idx|state_idx|      brand_ohe|        model_ohe|     color_ohe|
state_ohe|      feature_vector|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+--------------+-----------------+-------------+--
-------------+--------------------+
|  0| 6300|   toyota|cruiser|2008|clean vehicle|274117.0| black|  jtezu11f88k007763|159348797|    new jersey|
usa| 10 days left|     27.0|     89.0|      1.0|     10.0|     (27,[],[])| (126,[89],[1.0])|(48,[1],[1.0])|(43,
[10],[1.0])|(247,[116,154,211...|
|  1| 2899|     ford|     se|2011|clean vehicle|190552.0|silver|  2fmdk3gc4bbb02217|166951262|     tennessee|
usa|  6 days left|      0.0|    113.0|      3.0|     21.0| (27,[0],[1.0])|(126,[113],[1.0])|(48,[3],[1.0])|(43,
[21],[1.0])|(247,[0,140,156,2...|
|  2| 5350|    dodge|    mpv|2018|clean vehicle| 39590.0|silver|  3c4pdcgg5jt346413|167655728|       georgia|
usa|  2 days left|      1.0|      4.0|      3.0|     15.0| (27,[1],[1.0])|  (126,[4],[1.0])|(48,[3],[1.0])|(43,
[15],[1.0])|(247,[1,31,156,21...|
|  3|25000|     ford|   door|2014|clean vehicle| 64146.0|  blue|  1ftfw1et4efc23745|167753855|      virginia|
usa|22 hours left|      0.0|      0.0|      5.0|      9.0| (27,[0],[1.0])|  (126,[0],[1.0])|(48,[5],[1.0])| (4
3,[9],[1.0])|(247,[0,27,158,21...|
|  4|27700|chevrolet|   1500|2018|clean vehicle|  6654.0|   red|  3gcpcrec2jg473991|167763266|       florida|
usa|22 hours left|      3.0|     15.0|      4.0|      1.0| (27,[3],[1.0])| (126,[15],[1.0])|(48,[4],[1.0])| (4
3,[1],[1.0])|(247,[3,42,157,20...|
|  5| 5700|    dodge|    mpv|2018|clean vehicle| 45561.0| white|  2c4rdgeg9jr237989|167655771|         texas|
usa|  2 days left|      1.0|      4.0|      0.0|      2.0| (27,[1],[1.0])|  (126,[4],[1.0])|(48,[0],[1.0])| (4
3,[2],[1.0])|(247,[1,31,153,20...|
|  6| 7300|chevrolet|     pk|2010|clean vehicle|149050.0| black|  1gcsksea1az121133|167753872|       georgia|
usa|22 hours left|      3.0|     54.0|      1.0|     15.0| (27,[3],[1.0])| (126,[54],[1.0])|(48,[1],[1.0])|(43,
[15],[1.0])|(247,[3,81,154,21...|
|  7|13350|      gmc|   door|2017|clean vehicle| 23525.0|  gray|  1gks2gkc3hr326762|167692494|    california|
usa|20 hours left|      4.0|      0.0|      2.0|      3.0| (27,[4],[1.0])|  (126,[0],[1.0])|(48,[2],[1.0])| (4
3,[3],[1.0])|(247,[4,27,155,20...|
|  8|14600|chevrolet| malibu|2018|clean vehicle|  9371.0|silver|  1g1zd5st5jf191860|167763267|       florida|
usa|22 hours left|      3.0|     36.0|      3.0|      1.0| (27,[3],[1.0])| (126,[36],[1.0])|(48,[3],[1.0])| (4
3,[1],[1.0])|(247,[3,63,156,20...|
|  9| 5250|     ford|    mpv|2017|clean vehicle| 63418.0| black|  2fmpk3j92hbc12542|167656121|         texas|
usa|  2 days left|      0.0|      4.0|      1.0|      2.0| (27,[0],[1.0])|  (126,[4],[1.0])|(48,[1],[1.0])| (4
3,[2],[1.0])|(247,[0,31,154,20...|
| 10|10400|    dodge|  coupe|2009|clean vehicle|107856.0|orange|  2b3lj54t49h509675|167753874|       georgia|
usa|22 hours left|      1.0|     49.0|      8.0|     15.0| (27,[1],[1.0])| (126,[49],[1.0])|(48,[8],[1.0])|(43,
[15],[1.0])|(247,[1,76,161,21...|
| 11|12920|      gmc|    mpv|2017|clean vehicle| 39650.0| white|  1gks2bkc6hr136280|167692496|    california|
usa|20 hours left|      4.0|      4.0|      0.0|      3.0| (27,[4],[1.0])|  (126,[4],[1.0])|(48,[0],[1.0])| (4
3,[3],[1.0])|(247,[4,31,153,20...|
| 12|31900|chevrolet|   1500|2018|clean vehicle| 22909.0| black|  3gcukrec0jg176059|167763273|     tennessee|
usa|22 hours left|      3.0|     15.0|      1.0|     21.0| (27,[3],[1.0])| (126,[15],[1.0])|(48,[1],[1.0])|(43,
[21],[1.0])|(247,[3,42,154,22...|
| 13| 5430| chrysler|  wagon|2017|clean vehicle|138650.0|  gray|  2c4rc1cg5hr616095|167656123|         texas|
usa|  2 days left|      6.0|     22.0|      2.0|      2.0| (27,[6],[1.0])| (126,[22],[1.0])|(48,[2],[1.0])| (4
3,[2],[1.0])|(247,[6,49,155,20...|
| 14|20700|     ford|   door|2013|clean vehicle|100757.0| black|  1ftfw1et7dfa47790|167753916|      virginia|
usa|22 hours left|      0.0|      0.0|      1.0|      9.0| (27,[0],[1.0])|  (126,[0],[1.0])|(48,[1],[1.0])| (4
3,[9],[1.0])|(247,[0,27,154,21...|
| 15|12710|      gmc|   door|2017|clean vehicle| 25747.0| white|  1gks2gkc6hr328389|167692497|    california|
usa|20 hours left|      4.0|      0.0|      0.0|      3.0| (27,[4],[1.0])|  (126,[0],[1.0])|(48,[0],[1.0])| (4
3,[3],[1.0])|(247,[4,27,153,20...|
| 16| 5200|      kia|  forte|2018|clean vehicle| 46194.0|  blue|  3kpfk4a77je198723|167801757|north carolina|
usa|  2 days left|     10.0|     77.0|      5.0|      5.0|(27,[10],[1.0])| (126,[77],[1.0])|(48,[5],[1.0])| (4
3,[5],[1.0])|(247,[10,104,158,...|
| 17|16500|    buick| encore|2018|clean vehicle| 20002.0|   red|  kl4cj1sb6jb688844|167763275|     tennessee|
usa|22 hours left|      9.0|     65.0|      4.0|     21.0| (27,[9],[1.0])| (126,[65],[1.0])|(48,[4],[1.0])|(43,
[21],[1.0])|(247,[9,92,157,22...|
| 18| 5210|     ford|    mpv|2017|clean vehicle| 35714.0| white|  2fmpk3j95hbb73607|167656124|         texas|
usa|  2 days left|      0.0|      4.0|      0.0|      2.0| (27,[0],[1.0])|  (126,[4],[1.0])|(48,[0],[1.0])| (4
3,[2],[1.0])|(247,[0,31,153,20...|
| 19|38100|     ford|   door|2013|clean vehicle| 54380.0|  gray|  1ft8w3dt5deb68569|167753923|      virginia|
usa|22 hours left|      0.0|      0.0|      2.0|      9.0| (27,[0],[1.0])|  (126,[0],[1.0])|(48,[2],[1.0])| (4
3,[9],[1.0])|(247,[0,27,155,21...|
+---+-----+---------+-------+----+-------------+--------+------+------------------+---------+-------------+--
-----+-------------+---------+---------+---------+---------+--------------+-----------------+-------------+--
-------------+--------------------+
only showing top 20 rows
```

In [0]:

```
data_new.select('feature_vector').show()
```

```
+--------------------+
|      feature_vector|
+--------------------+
|(247,[116,154,211...|
|(247,[0,140,156,2...|
|(247,[1,31,156,21...|
|(247,[0,27,158,21...|
|(247,[3,42,157,20...|
|(247,[1,31,153,20...|
|(247,[3,81,154,21...|
|(247,[4,27,155,20...|
|(247,[3,63,156,20...|
|(247,[0,31,154,20...|
|(247,[1,76,161,21...|
|(247,[4,31,153,20...|
|(247,[3,42,154,22...|
|(247,[6,49,155,20...|
|(247,[0,27,154,21...|
|(247,[4,27,153,20...|
|(247,[10,104,158,...|
|(247,[9,92,157,22...|
|(247,[0,31,153,20...|
|(247,[0,27,155,21...|
+--------------------+
only showing top 20 rows
```

In [0]:

```
from pyspark.ml.regression import RandomForestRegressor
rf_mod = RandomForestRegressor(featuresCol="feature_vector",labelCol="price")
```

cross-validation

In [0]:

```
from pyspark.ml.tuning import ParamGridBuilder,CrossValidator
```

In [0]:

```
pm = ParamGridBuilder().addGrid(rf_mod.numTrees,[100,1500]).build()
```

In [0]:

```
from pyspark.ml.evaluation import RegressionEvaluator
cf = CrossValidator(estimator=rf_mod,estimatorParamMaps= pm,evaluator=RegressionEvaluator(labelCol="price"),numFolds=3)
```

Spliting of the data into train and test

In [0]:

```
tr_data,test_data = data_new.randomSplit([.75,.25],seed=100)
```

In [0]:

```
cvmode = cf.fit(tr_data)
```

In [0]:

```
bestmodel = cvmode.bestModel
```

In [0]:

```
bestmodel
```

Out[64]: RandomForestRegressionModel: uid=RandomForestRegressor_12c15e8d0040, numTrees=1500, numFeatures=247

In [0]:

```
pred = cvmode.transform(test_data)
```

In [0]:

```
pred.show()
```

```
+---+-----+-------------+-----------+----+-------------+--------+--------+-------------------+---------+-------
-------+-------+-------------+---------+---------+---------+---------+---------------+----------------+--------
------+---------------+--------------------+------------------+
|_c0|price|        brand|      model|year| title_status| mileage|   color|                vin|      lot|
state|country|    condition|brand_idx|model_idx|color_idx|state_idx|      brand_ohe|       model_ohe|     color
_ohe|      state_ohe|      feature_vector|        prediction|
+---+-----+-------------+-----------+----+-------------+--------+--------+-------------------+---------+-------
-------+-------+-------------+---------+---------+---------+---------+---------------+----------------+--------
------+---------------+--------------------+------------------+
|  3|25000|         ford|       door|2014|clean vehicle| 64146.0|    blue|  1ftfw1et4efc23745|167753855|      v
irginia|    usa|22 hours left|      0.0|      0.0|      5.0|      9.0| (27,[0],[1.0])| (126,[0],[1.0])|(48,[5],
[1.0])| (43,[9],[1.0])|(247,[0,27,158,21...|17985.262054819155|
| 10|10400|        dodge|      coupe|2009|clean vehicle|107856.0|  orange|  2b3lj54t49h509675|167753874|
georgia|    usa|22 hours left|      1.0|     49.0|      8.0|     15.0| (27,[1],[1.0])|(126,[49],[1.0])|(48,[8],
[1.0])|(43,[15],[1.0])|(247,[1,76,161,21...| 7427.310419044121|
| 11|12920|          gmc|        mpv|2017|clean vehicle| 39650.0|   white|  1gks2bkc6hr136280|167692496|    cal
ifornia|    usa|20 hours left|      4.0|      4.0|      0.0|      3.0| (27,[4],[1.0])| (126,[4],[1.0])|(48,[0],
[1.0])| (43,[3],[1.0])|(247,[4,31,153,20...|13062.167738111148|
| 14|20700|         ford|       door|2013|clean vehicle|100757.0|   black|  1ftfw1et7dfa47790|167753916|      v
irginia|    usa|22 hours left|      0.0|      0.0|      1.0|      9.0| (27,[0],[1.0])| (126,[0],[1.0])|(48,[1],
[1.0])| (43,[9],[1.0])|(247,[0,27,154,21...|10936.694215037629|
| 20|12520|          gmc|       door|2017|clean vehicle| 30114.0|   white|  1gks2gkcxhr328380|167692498|    cal
ifornia|    usa|20 hours left|      4.0|      0.0|      0.0|      3.0| (27,[4],[1.0])| (126,[0],[1.0])|(48,[0],
[1.0])| (43,[3],[1.0])|(247,[4,27,153,20...|15893.324746073875|
| 25|11900|          gmc|       door|2017|clean vehicle| 28040.0|  silver|  1gks2gkc2hr327918|167692503|    cal
ifornia|    usa|20 hours left|      4.0|      0.0|      3.0|      3.0| (27,[4],[1.0])| (126,[0],[1.0])|(48,[3],
[1.0])| (43,[3],[1.0])|(247,[4,27,156,20...| 16674.40233117507|
| 29|13000|         ford|    chassis|2008|clean vehicle| 93698.0|   white|  1fdwf37r28eb95687|167754098|     te
nnessee|    usa|22 hours left|      0.0|     59.0|      0.0|     21.0| (27,[0],[1.0])|(126,[59],[1.0])|(48,[0],
[1.0])|(43,[21],[1.0])|(247,[0,86,153,22...| 9149.201892507921|
| 34|13000|         ford|convertible|2003|clean vehicle| 58817.0|    gray|  1fahp60a63y107799|167754173|      v
irginia|    usa|22 hours left|      0.0|     88.0|      2.0|      9.0| (27,[0],[1.0])|(126,[88],[1.0])|(48,[2],
[1.0])| (43,[9],[1.0])|(247,[0,115,155,2...|17537.066767813463|
| 35|11900|          gmc|       door|2017|clean vehicle| 27965.0|   black|  1gks2gkc2hr328793|167692507|    cal
ifornia|    usa|20 hours left|      4.0|      0.0|      1.0|      3.0| (27,[4],[1.0])| (126,[0],[1.0])|(48,[1],
[1.0])| (43,[3],[1.0])|(247,[4,27,154,20...|16692.163584279926|
| 37|19200|    chevrolet|       1500|2018|clean vehicle|  2430.0|   white|  1gcncneh5jz302471|167763391|     te
nnessee|    usa|22 hours left|      3.0|     15.0|      0.0|     21.0| (27,[3],[1.0])|(126,[15],[1.0])|(48,[0],
[1.0])|(43,[21],[1.0])|(247,[3,42,153,22...| 23941.67903123921|
| 38| 7320|         ford|        mpv|2017|clean vehicle| 32366.0|    gold|  2fmpk4k96hbb16692|167656128|
texas|    usa|  2 days left|      0.0|      4.0|      9.0|      2.0| (27,[0],[1.0])| (126,[4],[1.0])|(48,[9],
[1.0])| (43,[2],[1.0])|(247,[0,31,162,20...|14572.887182775361|
| 40|12710|          gmc|       door|2017|clean vehicle| 30049.0|   white|  1gks2gkc6hr325461|167692508|    cal
ifornia|    usa|20 hours left|      4.0|      0.0|      0.0|      3.0| (27,[4],[1.0])| (126,[0],[1.0])|(48,[0],
[1.0])| (43,[3],[1.0])|(247,[4,27,153,20...|15893.324746073875|
| 41|16600|mercedes-benz|       vans|2016|clean vehicle| 97983.0|   white|  wd4pe7dd1gp253504|167802269|      n
ew york|    usa|19 hours left|     14.0|     81.0|      0.0|     14.0|(27,[14],[1.0])|(126,[81],[1.0])|(48,[0],
[1.0])|(43,[14],[1.0])|(247,[14,108,153,...|15750.629459485319|
| 48| 5580|        dodge|        mpv|2017|clean vehicle| 46829.0|    gray|  3c4pddeg7ht595704|167656130|
texas|    usa|  2 days left|      1.0|      4.0|      2.0|      2.0| (27,[1],[1.0])| (126,[4],[1.0])|(48,[2],
[1.0])| (43,[2],[1.0])|(247,[1,31,155,20...|  13122.8662956779|
| 52|27000|        buick|    enclave|2017|clean vehicle| 32107.0|no_color|  5gakrckdxhj113512|167763677|     te
nnessee|    usa|22 hours left|      9.0|     76.0|      6.0|     21.0| (27,[9],[1.0])|(126,[76],[1.0])|(48,[6],
[1.0])|(43,[21],[1.0])|(247,[9,103,159,2...|20866.172502336634|
| 58|23000|     chrysler|   pacifica|2020|clean vehicle|  2473.0|   white|  2c4rc1fg5lr143335|167656205|
ohio|    usa|  2 days left|      6.0|     69.0|      0.0|     18.0| (27,[6],[1.0])|(126,[69],[1.0])|(48,[0],[1.
0])|(43,[18],[1.0])|(247,[6,96,153,21...|21247.784286992854|
| 59|18500|     chrysler|        300|2019|clean vehicle| 38586.0|    gray|  2c3ccaeg5kh585628|167754410|south c
arolina|    usa|22 hours left|      6.0|     46.0|      2.0|     13.0| (27,[6],[1.0])|(126,[46],[1.0])|(48,[2],
[1.0])|(43,[13],[1.0])|(247,[6,73,155,21...|20446.727019346567|
| 65|12710|          gmc|       door|2017|clean vehicle| 32504.0|  silver|  1gks2gkc6hr327937|167692524|    cal
ifornia|    usa|20 hours left|      4.0|      0.0|      3.0|      3.0| (27,[4],[1.0])| (126,[0],[1.0])|(48,[3],
[1.0])| (43,[3],[1.0])|(247,[4,27,156,20...|14927.240771295355|
| 66|21100|         jeep|   cherokee|2019|clean vehicle| 28649.0|  silver|  1c4pjmbx6kd298174|167802534|north c
arolina|    usa|  2 days left|      5.0|     63.0|      3.0|      5.0| (27,[5],[1.0])|(126,[63],[1.0])|(48,[3],
[1.0])| (43,[5],[1.0])|(247,[5,90,156,20...| 21320.38590957216|
| 69|19300|     chrysler|        300|2019|clean vehicle| 34801.0|    gray|  2c3ccakg4kh594972|167754413|south c
arolina|    usa|22 hours left|      6.0|     46.0|      2.0|     13.0| (27,[6],[1.0])|(126,[46],[1.0])|(48,[2],
[1.0])|(43,[13],[1.0])|(247,[6,73,155,21...|20676.443006101563|
+---+-----+-------------+-----------+----+-------------+--------+--------+-------------------+---------+-------
-------+-------+-------------+---------+---------+---------+---------+---------------+----------------+--------
------+---------------+--------------------+------------------+
only showing top 20 rows
```

In [0]:

```
pred_mod = pred.select("price","prediction")
```

In [0]:

```
pred_mod.show()
```

```
+-----+------------------+
|price|        prediction|
+-----+------------------+
|25000|17985.262054819155|
|10400| 7427.310419044121|
|12920|13062.167738111148|
|20700|10936.694215037629|
|12520|15893.324746073875|
|11900| 16674.40233117507|
|13000| 9149.201892507921|
|13000|17537.066767813463|
|11900|16692.163584279926|
|19200| 23941.67903123921|
| 7320|14572.887182775361|
|12710|15893.324746073875|
|16600|15750.629459485319|
| 5580|  13122.8662956779|
|27000|20866.172502336634|
|23000|21247.784286992854|
|18500|20446.727019346567|
|12710|14927.240771295355|
|21100| 21320.38590957216|
|19300|20676.443006101563|
+-----+------------------+
only showing top 20 rows
```

Evaluate the regressor

In [0]:

```
evalutor = RegressionEvaluator()
```

In [0]:

```
pred = pred.withColumnRenamed("price","label")
```

In [0]:

```
eval_score = evalutor.evaluate(pred,{evalutor.metricName:'r2'})
```

In [0]:

```
eval_score
```

Out[72]: 0.47207570596995074

Practical : 9

Aim : Perform clustering with Pyspark

Import Libraries

In [0]:

```
import pyspark
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator
import seaborn as sns
import pandas as pd
from pyspark.sql.functions import *
```

Load Dataset

In [0]:

```
df = spark.read.option('header',True).option('inferSchema',True).csv('/FileStore/tables/uber_data-1.csv')
```

Explore Dataset

In [0]:

```
df.show()
```

```
+----------------+-------+--------+------+
|       Date/Time|    Lat|     Lon|  Base|
+----------------+-------+--------+------+
|4/1/2014 0:11:00| 40.769|-73.9549|B02512|
|4/1/2014 0:17:00|40.7267|-74.0345|B02512|
|4/1/2014 0:21:00|40.7316|-73.9873|B02512|
|4/1/2014 0:28:00|40.7588|-73.9776|B02512|
|4/1/2014 0:33:00|40.7594|-73.9722|B02512|
|4/1/2014 0:33:00|40.7383|-74.0403|B02512|
|4/1/2014 0:39:00|40.7223|-73.9887|B02512|
|4/1/2014 0:45:00| 40.762| -73.979|B02512|
|4/1/2014 0:55:00|40.7524| -73.996|B02512|
|4/1/2014 1:01:00|40.7575|-73.9846|B02512|
|4/1/2014 1:19:00|40.7256|-73.9869|B02512|
|4/1/2014 1:48:00|40.7591|-73.9684|B02512|
|4/1/2014 1:49:00|40.7271|-73.9803|B02512|
|4/1/2014 2:11:00|40.6463|-73.7896|B02512|
|4/1/2014 2:25:00|40.7564|-73.9167|B02512|
|4/1/2014 2:31:00|40.7666|-73.9531|B02512|
|4/1/2014 2:43:00| 40.758|-73.9761|B02512|
|4/1/2014 3:22:00|40.7238|-73.9821|B02512|
|4/1/2014 3:35:00|40.7531|-74.0039|B02512|
|4/1/2014 3:35:00|40.7389|-74.0393|B02512|
+----------------+-------+--------+------+
only showing top 20 rows
```

In [0]:

```
df.printSchema()
```

```
root
 |-- Date/Time: string (nullable = true)
 |-- Lat: double (nullable = true)
 |-- Lon: double (nullable = true)
 |-- Base: string (nullable = true)
```

In [0]:

```
spark.sql("set spark.sql.legacy.timeParserPolicy=LEGACY")
```

```
Out[43]: DataFrame[key: string, value: string]
```

In [0]:

```
df_2 = df.withColumn("date_new",to_date(col("Date/Time"),"MM/dd/yyyy HH:mm:ss"))
```

In [0]:

```
df_2.show()
```

```
+----------------+-------+--------+------+----------+
|       Date/Time|    Lat|     Lon|  Base|  date_new|
+----------------+-------+--------+------+----------+
|4/1/2014 0:11:00| 40.769|-73.9549|B02512|2014-04-01|
|4/1/2014 0:17:00|40.7267|-74.0345|B02512|2014-04-01|
|4/1/2014 0:21:00|40.7316|-73.9873|B02512|2014-04-01|
|4/1/2014 0:28:00|40.7588|-73.9776|B02512|2014-04-01|
|4/1/2014 0:33:00|40.7594|-73.9722|B02512|2014-04-01|
|4/1/2014 0:33:00|40.7383|-74.0403|B02512|2014-04-01|
|4/1/2014 0:39:00|40.7223|-73.9887|B02512|2014-04-01|
|4/1/2014 0:45:00| 40.762| -73.979|B02512|2014-04-01|
|4/1/2014 0:55:00|40.7524| -73.996|B02512|2014-04-01|
|4/1/2014 1:01:00|40.7575|-73.9846|B02512|2014-04-01|
|4/1/2014 1:19:00|40.7256|-73.9869|B02512|2014-04-01|
|4/1/2014 1:48:00|40.7591|-73.9684|B02512|2014-04-01|
|4/1/2014 1:49:00|40.7271|-73.9803|B02512|2014-04-01|
|4/1/2014 2:11:00|40.6463|-73.7896|B02512|2014-04-01|
|4/1/2014 2:25:00|40.7564|-73.9167|B02512|2014-04-01|
|4/1/2014 2:31:00|40.7666|-73.9531|B02512|2014-04-01|
|4/1/2014 2:43:00| 40.758|-73.9761|B02512|2014-04-01|
|4/1/2014 3:22:00|40.7238|-73.9821|B02512|2014-04-01|
|4/1/2014 3:35:00|40.7531|-74.0039|B02512|2014-04-01|
|4/1/2014 3:35:00|40.7389|-74.0393|B02512|2014-04-01|
+----------------+-------+--------+------+----------+
only showing top 20 rows
```

In [0]:

```
from pyspark.sql.functions import year,dayofweek,month
df_3 = df_2.withColumn('year',year(col('date_new')))
df_3.show()
```

```
+----------------+-------+--------+------+----------+----+
|       Date/Time|    Lat|     Lon|  Base|  date_new|year|
+----------------+-------+--------+------+----------+----+
|4/1/2014 0:11:00| 40.769|-73.9549|B02512|2014-04-01|2014|
|4/1/2014 0:17:00|40.7267|-74.0345|B02512|2014-04-01|2014|
|4/1/2014 0:21:00|40.7316|-73.9873|B02512|2014-04-01|2014|
|4/1/2014 0:28:00|40.7588|-73.9776|B02512|2014-04-01|2014|
|4/1/2014 0:33:00|40.7594|-73.9722|B02512|2014-04-01|2014|
|4/1/2014 0:33:00|40.7383|-74.0403|B02512|2014-04-01|2014|
|4/1/2014 0:39:00|40.7223|-73.9887|B02512|2014-04-01|2014|
|4/1/2014 0:45:00| 40.762| -73.979|B02512|2014-04-01|2014|
|4/1/2014 0:55:00|40.7524| -73.996|B02512|2014-04-01|2014|
|4/1/2014 1:01:00|40.7575|-73.9846|B02512|2014-04-01|2014|
|4/1/2014 1:19:00|40.7256|-73.9869|B02512|2014-04-01|2014|
|4/1/2014 1:48:00|40.7591|-73.9684|B02512|2014-04-01|2014|
|4/1/2014 1:49:00|40.7271|-73.9803|B02512|2014-04-01|2014|
|4/1/2014 2:11:00|40.6463|-73.7896|B02512|2014-04-01|2014|
|4/1/2014 2:25:00|40.7564|-73.9167|B02512|2014-04-01|2014|
|4/1/2014 2:31:00|40.7666|-73.9531|B02512|2014-04-01|2014|
|4/1/2014 2:43:00| 40.758|-73.9761|B02512|2014-04-01|2014|
|4/1/2014 3:22:00|40.7238|-73.9821|B02512|2014-04-01|2014|
|4/1/2014 3:35:00|40.7531|-74.0039|B02512|2014-04-01|2014|
|4/1/2014 3:35:00|40.7389|-74.0393|B02512|2014-04-01|2014|
+----------------+-------+--------+------+----------+----+
only showing top 20 rows
```

In [0]:

```
df_3.printSchema()
```

```
root
 |-- Date/Time: string (nullable = true)
 |-- Lat: double (nullable = true)
 |-- Lon: double (nullable = true)
 |-- Base: string (nullable = true)
 |-- date_new: date (nullable = true)
 |-- year: integer (nullable = true)
```

In [0]:

```
input_features = ['Lat','Lon']
va = VectorAssembler(inputCols=input_features,outputCol="features")
df_3 = va.transform(df_3)
df_3.show()
```

```
+----------------+-------+--------+------+----------+----+------------------+
|       Date/Time|    Lat|     Lon|  Base|  date_new|year|          features|
+----------------+-------+--------+------+----------+----+------------------+
|4/1/2014 0:11:00| 40.769|-73.9549|B02512|2014-04-01|2014| [40.769,-73.9549]|
|4/1/2014 0:17:00|40.7267|-74.0345|B02512|2014-04-01|2014|[40.7267,-74.0345]|
|4/1/2014 0:21:00|40.7316|-73.9873|B02512|2014-04-01|2014|[40.7316,-73.9873]|
|4/1/2014 0:28:00|40.7588|-73.9776|B02512|2014-04-01|2014|[40.7588,-73.9776]|
|4/1/2014 0:33:00|40.7594|-73.9722|B02512|2014-04-01|2014|[40.7594,-73.9722]|
|4/1/2014 0:33:00|40.7383|-74.0403|B02512|2014-04-01|2014|[40.7383,-74.0403]|
|4/1/2014 0:39:00|40.7223|-73.9887|B02512|2014-04-01|2014|[40.7223,-73.9887]|
|4/1/2014 0:45:00| 40.762| -73.979|B02512|2014-04-01|2014|  [40.762,-73.979]|
|4/1/2014 0:55:00|40.7524| -73.996|B02512|2014-04-01|2014| [40.7524,-73.996]|
|4/1/2014 1:01:00|40.7575|-73.9846|B02512|2014-04-01|2014|[40.7575,-73.9846]|
|4/1/2014 1:19:00|40.7256|-73.9869|B02512|2014-04-01|2014|[40.7256,-73.9869]|
|4/1/2014 1:48:00|40.7591|-73.9684|B02512|2014-04-01|2014|[40.7591,-73.9684]|
|4/1/2014 1:49:00|40.7271|-73.9803|B02512|2014-04-01|2014|[40.7271,-73.9803]|
|4/1/2014 2:11:00|40.6463|-73.7896|B02512|2014-04-01|2014|[40.6463,-73.7896]|
|4/1/2014 2:25:00|40.7564|-73.9167|B02512|2014-04-01|2014|[40.7564,-73.9167]|
|4/1/2014 2:31:00|40.7666|-73.9531|B02512|2014-04-01|2014|[40.7666,-73.9531]|
|4/1/2014 2:43:00| 40.758|-73.9761|B02512|2014-04-01|2014| [40.758,-73.9761]|
|4/1/2014 3:22:00|40.7238|-73.9821|B02512|2014-04-01|2014|[40.7238,-73.9821]|
|4/1/2014 3:35:00|40.7531|-74.0039|B02512|2014-04-01|2014|[40.7531,-74.0039]|
|4/1/2014 3:35:00|40.7389|-74.0393|B02512|2014-04-01|2014|[40.7389,-74.0393]|
+----------------+-------+--------+------+----------+----+------------------+
only showing top 20 rows
```

In [0]:

```
df_new = df_2.toPandas()
df_new.drop(['Date/Time','Base','date_new'],axis=1,inplace=True)
df_new
```

	Lat	Lon
0	40.7690	-73.9549
1	40.7267	-74.0345
2	40.7316	-73.9873
3	40.7588	-73.9776
4	40.7594	-73.9722
...	...	...
564511	40.7640	-73.9744
564512	40.7629	-73.9672
564513	40.7443	-73.9889
564514	40.6756	-73.9405
564515	40.6880	-73.9608

564516 rows × 2 columns

Elbow Method

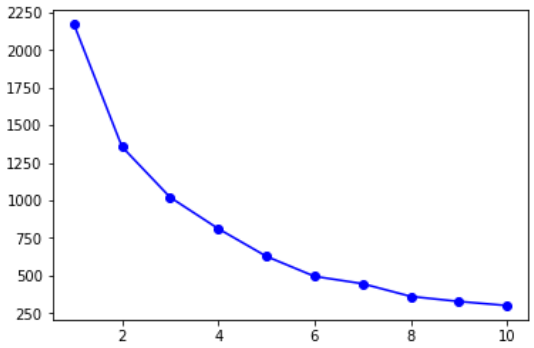
In [0]:

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmod = KMeans(n_clusters=i,init='random').fit(df_new)
    wcss.append(kmod.inertia_)
```

In [0]:

```
import matplotlib.pyplot as plt
plt.plot(range(1,11),wcss,marker='o',c='blue')
```

Out[51]: [<matplotlib.lines.Line2D at 0x7f0f91431130>]



Spliting data

In [0]:

```
training,testing = df_3.randomSplit([.75,.25])
```

In [0]:

```
training.show()
```

```
+----------------+-------+--------+------+----------+----+------------------+
|       Date/Time|    Lat|     Lon|  Base|  date_new|year|          features|
+----------------+-------+--------+------+----------+----+------------------+
|4/1/2014 0:00:00|40.7188|-73.9863|B02598|2014-04-01|2014|[40.7188,-73.9863]|
|4/1/2014 0:00:00|40.7637|  -73.96|B02598|2014-04-01|2014|  [40.7637,-73.96]|
|4/1/2014 0:02:00|40.7444|-73.9964|B02598|2014-04-01|2014|[40.7444,-73.9964]|
|4/1/2014 0:03:00|40.7741|-73.8722|B02598|2014-04-01|2014|[40.7741,-73.8722]|
|4/1/2014 0:11:00|40.7322|-73.9867|B02598|2014-04-01|2014|[40.7322,-73.9867]|
|4/1/2014 0:11:00| 40.769|-73.9549|B02512|2014-04-01|2014| [40.769,-73.9549]|
|4/1/2014 0:16:00|40.7748| -73.954|B02598|2014-04-01|2014| [40.7748,-73.954]|
|4/1/2014 0:21:00|40.7092|-73.7974|B02598|2014-04-01|2014|[40.7092,-73.7974]|
|4/1/2014 0:25:00|40.6483|-73.7826|B02598|2014-04-01|2014|[40.6483,-73.7826]|
|4/1/2014 0:25:00|40.7362| -73.992|B02598|2014-04-01|2014| [40.7362,-73.992]|
|4/1/2014 0:29:00|40.6909|-73.9595|B02598|2014-04-01|2014|[40.6909,-73.9595]|
|4/1/2014 0:31:00|40.7641| -73.976|B02598|2014-04-01|2014| [40.7641,-73.976]|
|4/1/2014 0:33:00|40.7383|-74.0403|B02512|2014-04-01|2014|[40.7383,-74.0403]|
|4/1/2014 0:33:00|40.7977|  -73.94|B02598|2014-04-01|2014|  [40.7977,-73.94]|
|4/1/2014 0:39:00|40.7387|-74.0033|B02598|2014-04-01|2014|[40.7387,-74.0033]|
|4/1/2014 0:40:00|40.7561| -73.987|B02598|2014-04-01|2014| [40.7561,-73.987]|
|4/1/2014 0:40:00|40.7594|-73.9766|B02598|2014-04-01|2014|[40.7594,-73.9766]|
|4/1/2014 0:43:00|40.6679|-73.8732|B02598|2014-04-01|2014|[40.6679,-73.8732]|
|4/1/2014 0:43:00|40.7019|-74.0112|B02598|2014-04-01|2014|[40.7019,-74.0112]|
|4/1/2014 0:46:00|40.6448|-73.7824|B02598|2014-04-01|2014|[40.6448,-73.7824]|
+----------------+-------+--------+------+----------+----+------------------+
only showing top 20 rows
```

Kmeans Algorithm

In [0]:

```
from pyspark.ml.clustering import KMeans
kmod = KMeans(k=3,initMode="k-means||",featuresCol='features',predictionCol='prediction').fit(training)
```

In [0]:

```
p = kmod.transform(testing)
p.show()
```

```
+-----------------+-------+--------+------+----------+----+------------------+----------+
|        Date/Time|    Lat|     Lon|  Base|  date_new|year|          features|prediction|
+-----------------+-------+--------+------+----------+----+------------------+----------+
| 4/1/2014 0:02:00|40.7556|-73.9874|B02598|2014-04-01|2014|[40.7556,-73.9874]|         1|
| 4/1/2014 0:05:00|40.7572|-73.9781|B02598|2014-04-01|2014|[40.7572,-73.9781]|         1|
| 4/1/2014 0:11:00|40.8134|-73.9453|B02598|2014-04-01|2014|[40.8134,-73.9453]|         1|
| 4/1/2014 0:17:00|40.7267|-74.0345|B02512|2014-04-01|2014|[40.7267,-74.0345]|         0|
| 4/1/2014 0:21:00|40.7316|-73.9873|B02512|2014-04-01|2014|[40.7316,-73.9873]|         0|
| 4/1/2014 0:22:00|40.7141|-74.0148|B02598|2014-04-01|2014|[40.7141,-74.0148]|         0|
| 4/1/2014 0:25:00|40.7551| -73.984|B02598|2014-04-01|2014| [40.7551,-73.984]|         1|
| 4/1/2014 0:28:00|40.7588|-73.9776|B02512|2014-04-01|2014|[40.7588,-73.9776]|         1|
| 4/1/2014 0:33:00|40.7594|-73.9722|B02512|2014-04-01|2014|[40.7594,-73.9722]|         1|
| 4/1/2014 0:39:00|40.7223|-73.9887|B02512|2014-04-01|2014|[40.7223,-73.9887]|         0|
| 4/1/2014 0:44:00|40.7438|-73.9923|B02598|2014-04-01|2014|[40.7438,-73.9923]|         0|
| 4/1/2014 0:45:00| 40.762| -73.979|B02512|2014-04-01|2014|  [40.762,-73.979]|         1|
| 4/1/2014 0:48:00|40.7263|-73.9917|B02598|2014-04-01|2014|[40.7263,-73.9917]|         0|
| 4/1/2014 0:49:00|40.7652|-73.9714|B02598|2014-04-01|2014|[40.7652,-73.9714]|         1|
| 4/1/2014 0:52:00|40.7595|-73.9681|B02598|2014-04-01|2014|[40.7595,-73.9681]|         1|
| 4/1/2014 0:52:00|40.7611|-73.9694|B02598|2014-04-01|2014|[40.7611,-73.9694]|         1|
| 4/1/2014 0:55:00|40.7524| -73.996|B02512|2014-04-01|2014| [40.7524,-73.996]|         0|
| 4/1/2014 0:58:00|40.7594|-73.9633|B02598|2014-04-01|2014|[40.7594,-73.9633]|         1|
|4/1/2014 10:02:00|  40.75|-73.9728|B02598|2014-04-01|2014|  [40.75,-73.9728]|         1|
|4/1/2014 10:04:00|40.7309|-73.9894|B02598|2014-04-01|2014|[40.7309,-73.9894]|         0|
+-----------------+-------+--------+------+----------+----+------------------+----------+
only showing top 20 rows
```

In [0]:

```
p1=p.toPandas()
```

```
/databricks/spark/python/pyspark/sql/pandas/conversion.py:119: UserWarning: toPandas attempted Arrow op
timization because 'spark.sql.execution.arrow.pyspark.enabled' is set to true; however, failed by the r
eason below:
  Unable to convert the field features. If this column is not necessary, you may consider dropping it o
r converting to primitive type before the conversion.
Direct cause: Unsupported type in conversion to Arrow: VectorUDT()
Attempting non-optimization as 'spark.sql.execution.arrow.pyspark.fallback.enabled' is set to true.
  warn(msg)
```

In [0]:

```
p1['prediction'].value_counts()
```

```
Out[57]: 0    77245
1    59161
2     4241
Name: prediction, dtype: int64
```

In [0]:

```
p1.shape
```

```
Out[58]: (140647, 8)
```

In [0]:

```
p.createOrReplaceTempView('uber')
```

In [0]:

```
df = spark.sql('select prediction,count(*) from uber group by prediction')
```

In [0]:

```
df.show()
```

```
+----------+--------+
|prediction|count(1)|
+----------+--------+
|         1|   59161|
|         2|    4241|
|         0|   77245|
+----------+--------+
```

Practical : 10

Aim : Perform Deep Learning in Keras in distributed computing using Pyspark

Import Libraries

In [0]:

```
from pyspark.ml import Pipeline
from pyspark.ml.feature import OneHotEncoder,StringIndexer,StandardScaler,VectorAssembler
from pyspark.mllib.evaluation import MulticlassMetrics
!pip install keras
!pip install tensorflow
!pip install elephas
from keras.models import Sequential
from keras.layers.core import Dense,Dropout,Activation
from keras import optimizers,regularizers
from keras.optimizers import Adam
from elephas.ml_model import ElephasEstimator
```

```
Collecting keras
  Downloading keras-2.11.0-py2.py3-none-any.whl (1.7 MB)
     |████████████████████████████████| 1.7 MB 5.8 MB/s eta 0:00:01
Installing collected packages: keras
Successfully installed keras-2.11.0
WARNING: You are using pip version 21.2.4; however, version 22.3.1 is available.
You should consider upgrading via the '/local_disk0/.ephemeral_nfs/envs/pythonEnv-f36f4242-7f5b-4d1a-a
431-f62072a4356a/bin/python -m pip install --upgrade pip' command.
Collecting tensorflow
  Downloading tensorflow-2.11.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (588.3 MB)
     |                                | 1.8 MB 5.8 MB/s eta 0:01:41

*** WARNING: max output size exceeded, skipping output. ***

     |████████████████████████████████| 438 kB 40.9 MB/s eta 0:00:01
Collecting keras-preprocessing>=1.1.1
  Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
     |████████████████████████████████| 42 kB 1.4 MB/s  eta 0:00:01
Requirement already satisfied: opt-einsum>=2.3.2 in /local_disk0/.ephemeral_nfs/envs/pythonEnv-f36f424
```

Load Dataset

In [0]:

```
df = spark.read.option('header',True).option('inferSchema',True).csv('/FileStore/tables/bank_new.csv')
```

Explore Dataset

In [0]:

```
df.show(10)
```

```
+---+----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+---
--+--------+--------+-------+
|age|       job| marital|education|default|balance|housing|loan|contact|day|month|duration|campaign|pda
ys|previous|poutcome|deposit|
+---+----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+---
--+--------+--------+-------+
| 59|    admin.| married|secondary|     no|   2343|    yes|  no|unknown|  5|  may|    1042|       1|
-1|       0| unknown|    yes|
| 56|    admin.| married|secondary|     no|     45|     no|  no|unknown|  5|  may|    1467|       1|
-1|       0| unknown|    yes|
| 41|technician| married|secondary|     no|   1270|    yes|  no|unknown|  5|  may|    1389|       1|
-1|       0| unknown|    yes|
| 55|  services| married|secondary|     no|   2476|    yes|  no|unknown|  5|  may|     579|       1|
-1|       0| unknown|    yes|
| 54|    admin.| married| tertiary|     no|    184|     no|  no|unknown|  5|  may|     673|       2|
-1|       0| unknown|    yes|
| 42|management|  single| tertiary|     no|      0|    yes| yes|unknown|  5|  may|     562|       2|
-1|       0| unknown|    yes|
| 56|management| married| tertiary|     no|    830|    yes| yes|unknown|  6|  may|    1201|       1|
-1|       0| unknown|    yes|
| 60|   retired|divorced|secondary|     no|    545|    yes|  no|unknown|  6|  may|    1030|       1|
-1|       0| unknown|    yes|
| 37|technician| married|secondary|     no|      1|    yes|  no|unknown|  6|  may|     608|       1|
-1|       0| unknown|    yes|
| 28|  services|  single|secondary|     no|   5090|    yes|  no|unknown|  6|  may|    1297|       3|
-1|       0| unknown|    yes|
+---+----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+---
--+--------+--------+-------+
only showing top 10 rows
```

In [0]:

```
df.printSchema()
```

```
root
 |-- age: integer (nullable = true)
 |-- job: string (nullable = true)
 |-- marital: string (nullable = true)
 |-- education: string (nullable = true)
 |-- default: string (nullable = true)
 |-- balance: integer (nullable = true)
 |-- housing: string (nullable = true)
 |-- loan: string (nullable = true)
 |-- contact: string (nullable = true)
 |-- day: integer (nullable = true)
 |-- month: string (nullable = true)
 |-- duration: integer (nullable = true)
 |-- campaign: integer (nullable = true)
 |-- pdays: integer (nullable = true)
 |-- previous: integer (nullable = true)
 |-- poutcome: string (nullable = true)
 |-- deposit: string (nullable = true)
```

In [0]:

```
df1 = df.toPandas()
dt = df1.dtypes
numeric = dt[dt=='int32'].index
categorical = dt[dt=='object'].index
categorical
```

```
Out[5]: Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome', 'deposit'],
      dtype='object')
```

In [0]:

```
numeric
```

```
Out[6]: Index(['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous'], dtype='object')
```

String to Numeric

In [0]:

```
si = StringIndexer(inputCols=['job', 'marital', 'education', 'default',
                              'housing', 'loan', 'contact',
       'month', 'poutcome', 'deposit'],outputCols=['job_idx', 'marital_idx',
                                                   'education_idx',
                                                   'default_idx',
                                                   'housing_idx',
                                                   'loan_idx',
                                                   'contact_idx',
                                                   'month_idx', 'poutcome_idx',
                                                   'label'])
```

In [0]:

```
si
```

Out[8]: StringIndexer_2eeeb7e3e1b9

In [0]:

```
df_si = si.fit(df)
df_si = df_si.transform(df)
```

In [0]:

```
df_si.columns
```

Out[10]: ['age',
 'job',
 'marital',
 'education',
 'default',
 'balance',
 'housing',
 'loan',
 'contact',
 'day',
 'month',
 'duration',
 'campaign',
 'pdays',
 'previous',
 'poutcome',
 'deposit',
 'job_idx',
 'marital_idx',
 'education_idx',
 'default_idx',
 'housing_idx',
 'loan_idx',
 'contact_idx',
 'month_idx',
 'poutcome_idx',
 'label']

One hot encoding

In [0]:

```
ohe = OneHotEncoder(inputCols=['job_idx', 'marital_idx',
                               'education_idx', 'default_idx',
                               'housing_idx', 'loan_idx', 'contact_idx',
                               'month_idx', 'poutcome_idx'],
                    outputCols=['job_ohe', 'marital_ohe',
                                'education_ohe', 'default_ohe',
                                'housing_ohe', 'loan_ohe',
                                'contact_ohe','month_ohe', 'poutcome_ohe'])
```

In [0]:

```
df_si.show()
```

```
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+--
---+--------+--------+-------+-------+-----------+-------------+-----------+-----------+--------+------
-----+---------+------------+-----+
|age|        job| marital|education|default|balance|housing|loan|contact|day|month|duration|campaign|pd
ays|previous|poutcome|deposit|job_idx|marital_idx|education_idx|default_idx|housing_idx|loan_idx|contac
t_idx|month_idx|poutcome_idx|label|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+--
---+--------+--------+-------+-------+-----------+-------------+-----------+-----------+--------+------
-----+---------+------------+-----+
| 59|     admin.| married|secondary|     no|   2343|    yes|  no|unknown|  5|  may|    1042|       1|
-1|       0| unknown|    yes|    3.0|        0.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 56|     admin.| married|secondary|     no|     45|     no|  no|unknown|  5|  may|    1467|       1|
-1|       0| unknown|    yes|    3.0|        0.0|          0.0|        0.0|        0.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 41| technician| married|secondary|     no|   1270|    yes|  no|unknown|  5|  may|    1389|       1|
-1|       0| unknown|    yes|    2.0|        0.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 55|   services| married|secondary|     no|   2476|    yes|  no|unknown|  5|  may|     579|       1|
-1|       0| unknown|    yes|    4.0|        0.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 54|     admin.| married| tertiary|     no|    184|     no|  no|unknown|  5|  may|     673|       2|
-1|       0| unknown|    yes|    3.0|        0.0|          1.0|        0.0|        0.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 42| management|  single| tertiary|     no|      0|    yes| yes|unknown|  5|  may|     562|       2|
-1|       0| unknown|    yes|    0.0|        1.0|          1.0|        0.0|        1.0|     1.0|
1.0|      0.0|         0.0|  1.0|
| 56| management| married| tertiary|     no|    830|    yes| yes|unknown|  6|  may|    1201|       1|
-1|       0| unknown|    yes|    0.0|        0.0|          1.0|        0.0|        1.0|     1.0|
1.0|      0.0|         0.0|  1.0|
| 60|    retired|divorced|secondary|     no|    545|    yes|  no|unknown|  6|  may|    1030|       1|
-1|       0| unknown|    yes|    5.0|        2.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 37| technician| married|secondary|     no|      1|    yes|  no|unknown|  6|  may|     608|       1|
-1|       0| unknown|    yes|    2.0|        0.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 28|   services|  single|secondary|     no|   5090|    yes|  no|unknown|  6|  may|    1297|       3|
-1|       0| unknown|    yes|    4.0|        1.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 38|     admin.|  single|secondary|     no|    100|    yes|  no|unknown|  7|  may|     786|       1|
-1|       0| unknown|    yes|    3.0|        1.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 30|blue-collar| married|secondary|     no|    309|    yes|  no|unknown|  7|  may|    1574|       2|
-1|       0| unknown|    yes|    1.0|        0.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 29| management| married| tertiary|     no|    199|    yes| yes|unknown|  7|  may|    1689|       4|
-1|       0| unknown|    yes|    0.0|        0.0|          1.0|        0.0|        1.0|     1.0|
1.0|      0.0|         0.0|  1.0|
| 46|blue-collar|  single| tertiary|     no|    460|    yes|  no|unknown|  7|  may|    1102|       2|
-1|       0| unknown|    yes|    1.0|        1.0|          1.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 31| technician|  single| tertiary|     no|    703|    yes|  no|unknown|  8|  may|     943|       2|
-1|       0| unknown|    yes|    2.0|        1.0|          1.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 35| management|divorced| tertiary|     no|   3837|    yes|  no|unknown|  8|  may|    1084|       1|
-1|       0| unknown|    yes|    0.0|        2.0|          1.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 32|blue-collar|  single|  primary|     no|    611|    yes|  no|unknown|  8|  may|     541|       3|
-1|       0| unknown|    yes|    1.0|        1.0|          2.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 49|   services| married|secondary|     no|     -8|    yes|  no|unknown|  8|  may|    1119|       1|
-1|       0| unknown|    yes|    4.0|        0.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 41|     admin.| married|secondary|     no|     55|    yes|  no|unknown|  8|  may|    1120|       2|
-1|       0| unknown|    yes|    3.0|        0.0|          0.0|        0.0|        1.0|     0.0|
1.0|      0.0|         0.0|  1.0|
| 49|     admin.|divorced|secondary|     no|    168|    yes| yes|unknown|  8|  may|     513|       1|
-1|       0| unknown|    yes|    3.0|        2.0|          0.0|        0.0|        1.0|     1.0|
1.0|      0.0|         0.0|  1.0|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-----+--------+--------+--
---+--------+--------+-------+-------+-----------+-------------+-----------+-----------+--------+------
-----+---------+------------+-----+
only showing top 20 rows
```

In [0]:

```
df_ohe = ohe.fit(df_si).transform(df_si)
```

In [0]:

```
df_ohe.select("job_ohe","marital_ohe").show()
```

```
+--------------+-------------+
|       job_ohe|  marital_ohe|
+--------------+-------------+
|(11,[3],[1.0])|(2,[0],[1.0])|
|(11,[3],[1.0])|(2,[0],[1.0])|
|(11,[2],[1.0])|(2,[0],[1.0])|
|(11,[4],[1.0])|(2,[0],[1.0])|
|(11,[3],[1.0])|(2,[0],[1.0])|
|(11,[0],[1.0])|(2,[1],[1.0])|
|(11,[0],[1.0])|(2,[0],[1.0])|
|(11,[5],[1.0])|    (2,[],[])|
|(11,[2],[1.0])|(2,[0],[1.0])|
|(11,[4],[1.0])|(2,[1],[1.0])|
|(11,[3],[1.0])|(2,[1],[1.0])|
|(11,[1],[1.0])|(2,[0],[1.0])|
|(11,[0],[1.0])|(2,[0],[1.0])|
|(11,[1],[1.0])|(2,[1],[1.0])|
|(11,[2],[1.0])|(2,[1],[1.0])|
|(11,[0],[1.0])|    (2,[],[])|
|(11,[1],[1.0])|(2,[1],[1.0])|
|(11,[4],[1.0])|(2,[0],[1.0])|
|(11,[3],[1.0])|(2,[0],[1.0])|
|(11,[3],[1.0])|    (2,[],[])|
+--------------+-------------+
only showing top 20 rows
```

In [0]:

```
numeric
```

```
Out[15]: Index(['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous'], dtype='object')
```

In [0]:

```
from pyspark.ml.feature import VectorAssembler
va = VectorAssembler(inputCols=['age', 'balance', 'day',
                                'duration', 'campaign',
                                'pdays', 'previous','job_ohe',
                                'marital_ohe', 'default_ohe',
                                'housing_ohe', 'loan_ohe',
                                'contact_ohe','month_ohe',
                                'poutcome_ohe'],outputCol="features")
```

In [0]:

```
df_new = va.transform(df_ohe)
```

In [0]:

```
d1 = df_new.select("features").toPandas()
```

```
/databricks/spark/python/pyspark/sql/pandas/conversion.py:119: UserWarning: toPandas attempted Arrow op
timization because 'spark.sql.execution.arrow.pyspark.enabled' is set to true; however, failed by the r
eason below:
  Unable to convert the field features. If this column is not necessary, you may consider dropping it o
r converting to primitive type before the conversion.
Direct cause: Unsupported type in conversion to Arrow: VectorUDT()
Attempting non-optimization as 'spark.sql.execution.arrow.pyspark.fallback.enabled' is set to true.
  warn(msg)
```

In [0]:

```
d1.head()
```

	features
0	(59.0, 2343.0, 5.0, 1042.0, 1.0, -1.0, 0.0, 0....
1	(56.0, 45.0, 5.0, 1467.0, 1.0, -1.0, 0.0, 0.0,...
2	(41.0, 1270.0, 5.0, 1389.0, 1.0, -1.0, 0.0, 0....
3	(55.0, 2476.0, 5.0, 579.0, 1.0, -1.0, 0.0, 0.0...
4	(54.0, 184.0, 5.0, 673.0, 2.0, -1.0, 0.0, 0.0,...

In [0]:

```
from pyspark.sql.functions import rand
df_new = df_new.select("features","label")
df_new = df_new.orderBy(rand())
df_new.show()
```

```
+--------------------+-----+
|            features|label|
+--------------------+-----+
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,2,3,4,5,12...|  1.0|
|(39,[0,2,3,4,5,13...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,2,3,4,5,9,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
+--------------------+-----+
only showing top 20 rows
```

In [0]:

```
df2 = df_new.toPandas()
df2.head()
```

```
/databricks/spark/python/pyspark/sql/pandas/conversion.py:119: UserWarning: toPandas attempted Arrow op
timization because 'spark.sql.execution.arrow.pyspark.enabled' is set to true; however, failed by the r
eason below:
  Unable to convert the field features. If this column is not necessary, you may consider dropping it o
r converting to primitive type before the conversion.
Direct cause: Unsupported type in conversion to Arrow: VectorUDT()
Attempting non-optimization as 'spark.sql.execution.arrow.pyspark.fallback.enabled' is set to true.
  warn(msg)
```

	features	label
0	(30.0, 778.0, 18.0, 22.0, 2.0, 346.0, 2.0, 0.0...	0.0
1	(55.0, 1320.0, 28.0, 387.0, 2.0, -1.0, 0.0, 0....	1.0
2	(56.0, 6507.0, 29.0, 134.0, 1.0, 196.0, 1.0, 1...	0.0
3	(38.0, 1354.0, 29.0, 50.0, 2.0, -1.0, 0.0, 1.0...	0.0
4	(36.0, 3949.0, 28.0, 130.0, 2.0, -1.0, 0.0, 1....	1.0

In [0]:

```
df_new.select("features").show(5)
```

```
+--------------------+
|            features|
+--------------------+
|(39,[0,1,2,3,4,5,...|
|(39,[0,1,2,3,4,5,...|
|(39,[0,1,2,3,4,5,...|
|(39,[0,1,2,3,4,5,...|
|(39,[0,1,2,3,4,5,...|
+--------------------+
only showing top 5 rows
```

Data Spliting

In [0]:

```
df_tr,df_test = df_new.randomSplit([.8,.2])
```

In [0]:

```
df_tr.show()
```

```
+--------------------+-----+
|            features|label|
+--------------------+-----+
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
|(39,[0,1,2,3,4,5,...|  1.0|
|(39,[0,1,2,3,4,5,...|  0.0|
+--------------------+-----+
only showing top 20 rows
```

In [0]:

```
n_classes = df_tr.select("label").distinct().count()
```

In [0]:

```
n_classes
```

Out[26]: 2

In [0]:

```
input_dim = len(df_tr.select("features").first()[0])
input_dim
```

Out[27]: 39

Model

In [0]:

```
model = Sequential()
model.add(Dense(64,input_shape=(input_dim,),activity_regularizer=regularizers.l2(0.01)))
model.add(Activation('relu'))
model.add(Dropout(rate=0.3))
model.add(Dense(64,input_shape=(input_dim,),activity_regularizer=regularizers.l2(0.01)))
model.add(Activation('relu'))
model.add(Dropout(rate=0.3))
model.add(Dense(n_classes))
model.add(Activation('sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam')
optimizer_conf= optimizers.Adam (learning_rate=0.01)
opt_conf = optimizers.serialize(optimizer_conf)
```

Initialize SparkML Estimator and Get Settings

In [0]:

```
estimator = ElephasEstimator()
estimator.setFeaturesCol("features")
estimator.setLabelCol("label")
estimator.set_keras_model_config(model.to_json())
estimator.set_categorical_labels (True)
estimator.set_nb_classes (n_classes)
estimator.set_num_workers (1)
estimator.set_epochs (25)
estimator.set_batch_size(64)
estimator.set_verbosity (1)
estimator.set_validation_split(0.10)
estimator.set_optimizer_config(opt_conf)
estimator.set_mode("synchronous")
estimator.set_loss ("binary_crossentropy")
estimator.set_metrics (['acc'])
```

```
Out[29]: ElephasEstimator_c233dce4799c
```

In [0]:

```
dl_pipeline=Pipeline(stages=[estimator])
fit_pipeline=dl_pipeline.fit(df_new)
```

```
>>> Fit model
>>> Synchronous training complete.
```

In [0]:

```
df_new.printSchema()
```

```
root
 |-- features: vector (nullable = true)
 |-- label: double (nullable = false)
```

In [0]:

```
pred_train = fit_pipeline.transform(df_new)
pred_train.show()
```

```
+--------------------+-----+--------------------+
|            features|label|          prediction|
+--------------------+-----+--------------------+
|(39,[0,1,2,3,4,5,...|  0.0|[0.67701989412307...|
|(39,[0,1,2,3,4,5,...|  1.0|[0.64776903390884...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.56340593099594...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.73841655254364...|
|(39,[0,1,2,3,4,5,...|  1.0|[0.60331034660339...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.72133988142013...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.73504894971847...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.60208499431610...|
|(39,[0,1,2,3,4,5,...|  1.0|[0.05617255344986...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.68627619743347...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.63884025812149...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.61648452281951...|
|(39,[0,2,3,4,5,12...|  1.0|[0.64047831296920...|
|(39,[0,2,3,4,5,13...|  1.0|[0.49282470345497...|
|(39,[0,1,2,3,4,5,...|  1.0|[0.65757822990417...|
|(39,[0,2,3,4,5,9,...|  1.0|[0.57494086027145...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.72053855657577...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.75772869586944...|
|(39,[0,1,2,3,4,5,...|  1.0|[0.67787683010101...|
|(39,[0,1,2,3,4,5,...|  0.0|[0.61352610588073...|
+--------------------+-----+--------------------+
only showing top 20 rows
```

In [0]:

```
pnl_train = pred_train.select("label","prediction")
pnl_train.printSchema()
```

```
root
 |-- label: double (nullable = false)
 |-- prediction: array (nullable = true)
 |    |-- element: double (containsNull = true)
```

In [0]:

```
from pyspark.sql.functions import when
pnl_train.show()
```

```
+-----+--------------------+
|label|          prediction|
+-----+--------------------+
|  0.0|[0.67701989412307...|
|  1.0|[0.64776903390884...|
|  0.0|[0.56340593099594...|
|  0.0|[0.73841655254364...|
|  1.0|[0.60331034660339...|
|  0.0|[0.72133988142013...|
|  0.0|[0.73504894971847...|
|  0.0|[0.60208499431610...|
|  1.0|[0.05617255344986...|
|  0.0|[0.68627619743347...|
|  0.0|[0.63884025812149...|
|  0.0|[0.61648452281951...|
|  1.0|[0.64047831296920...|
|  1.0|[0.49282470345497...|
|  1.0|[0.65757822990417...|
|  1.0|[0.57494086027145...|
|  0.0|[0.72053855657577...|
|  0.0|[0.75772869586944...|
|  1.0|[0.67787683010101...|
|  0.0|[0.61352610588073...|
+-----+--------------------+
only showing top 20 rows
```

In [0]:

```
import numpy as np
df = pnl_train.toPandas()
li_pred = []
for i in df['prediction']:
    li_pred.append(np.argmax(i))
li_pred
df['pred_new'] = li_pred
df.head()
```

	label	prediction	pred_new
0	0.0	[0.6770198941230774, 0.31372272968292236]	0
1	1.0	[0.647769033908844, 0.3434373140335083]	0
2	0.0	[0.5634059309959412, 0.41150370240211487]	0
3	0.0	[0.7384165525436401, 0.2539145350456238]	0
4	1.0	[0.6033103466033936, 0.38043245673179626]	0

Result

In [0]:

```
from sklearn.metrics import confusion_matrix,accuracy_score
```

In [0]:

```
confusion_matrix(df['label'],df['pred_new'])
```

```
Out[37]: array([[5651,  222],
       [4620,  669]])
```

In [0]:

```
accuracy_score(df['label'],df['pred_new'])
```

```
Out[38]: 0.5662067729797527
```