# Marketing Analysis

## 1) Load data and create spark dataframe -

### a) Load the data file marketanalysis-

```
val data =
sc.textFile(file:///home/hadoop/Downloads/datasets/marketanalysis.csv)
```

### b) Clean the data file(using replace and split function)-

```
val banks = data.map( x => x.replace(“\””,””).split(“;”))
```

### c) Drop the first line(header)-

```
val bankf = banks.mapPartitionsWithIndex{(idx,iter) => if (idx==0) iter.drop(1)
else iter}
```

### d) Create the schema(using case class)-

```
case class Bank(age: Int, job: String, marital: String, education: String,
creditdefault: String, balance: Int, housing: String, loan: String, contact: String,
day: Int, month: String, duration: Int, campign: Int, pdays: Int, previous: Int,
poutcome: String, y: String)
```

### e) Transform each element into case class(map transformation)-

```
val bankrdd = bankf.map(x =>Bank(x(0).toInt, x(1), x(2), x(3), x(4), x(5).toInt,
x(6), x(7), x(8), x(9).toInt, x(10), x(11).toInt, x(12).toInt, x(13).toInt, x(14).toInt,
x(15), x(16)))
```

### f) Converting into Dataframe-

```
val bankdf = bankrdd.toDF()
```

### g) Register as temp table-

```
bankdf.registerTempTable(“bank”)
bankdf.createOrReplaceTempView(“bank”)
spark.table(“bank”)
```

### Performing basic sql –

```
spark.sql(“select age from bank”).show()
spark.sql(“select count(*) from bank”).show()
```

## 2) Marketing success rate-

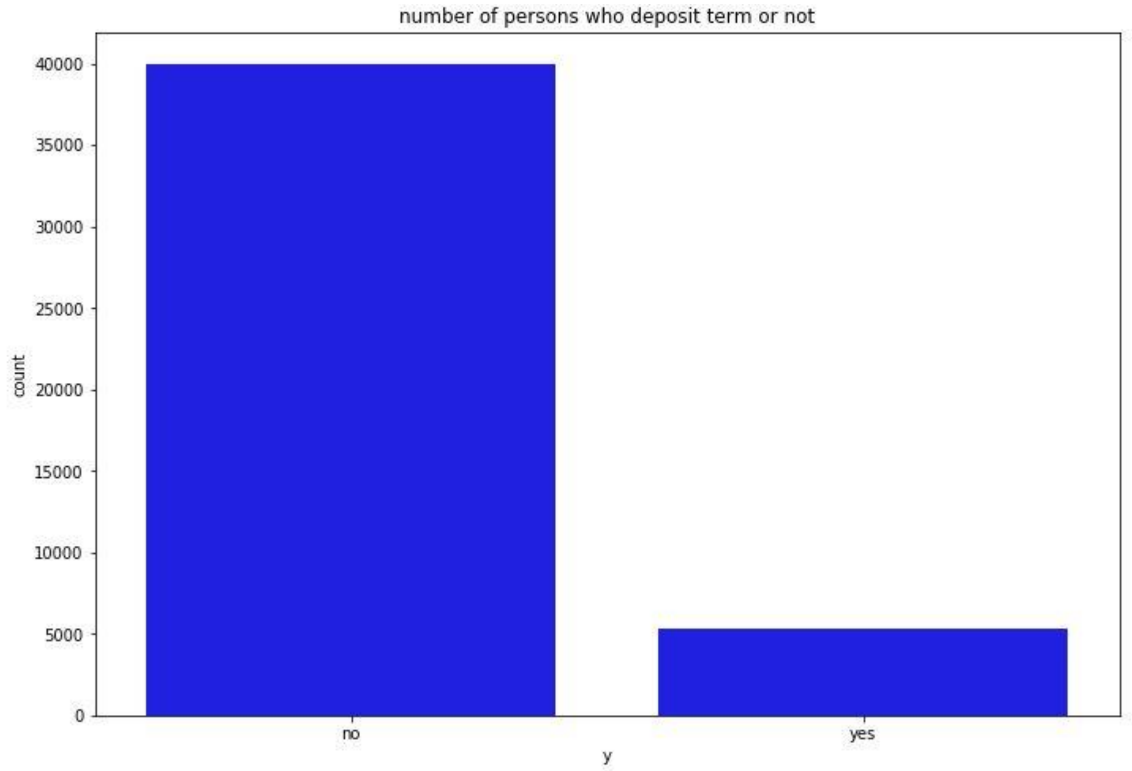
```
val success_ratio = bankdf.filter($”y” === “yes”).count().toDouble / bankdf.count()
```

Answer- 0.1169(11.69%)

## 2a) Marketing Failure rate-

```
val failure_ratio = bankdf.filter($”y” === “no”).count().toDouble / bankdf.count()
```

Answer- 0.8830(88.30%)



Answer- We also see using the plot the number of customer is No groups is more

## 3) Maximum,Minimum and Mean age of targeted customer-

```
bankdf.select(max($”age”)).show()
bankdf.select(avg($”age”)).show()
bankdf.select(min($”age”)).show()
```

Answer- Maximum age is 95

Minimum age is 18

Mean age is 40.93 ==41

## 4) Avg and Median Balance of customer-

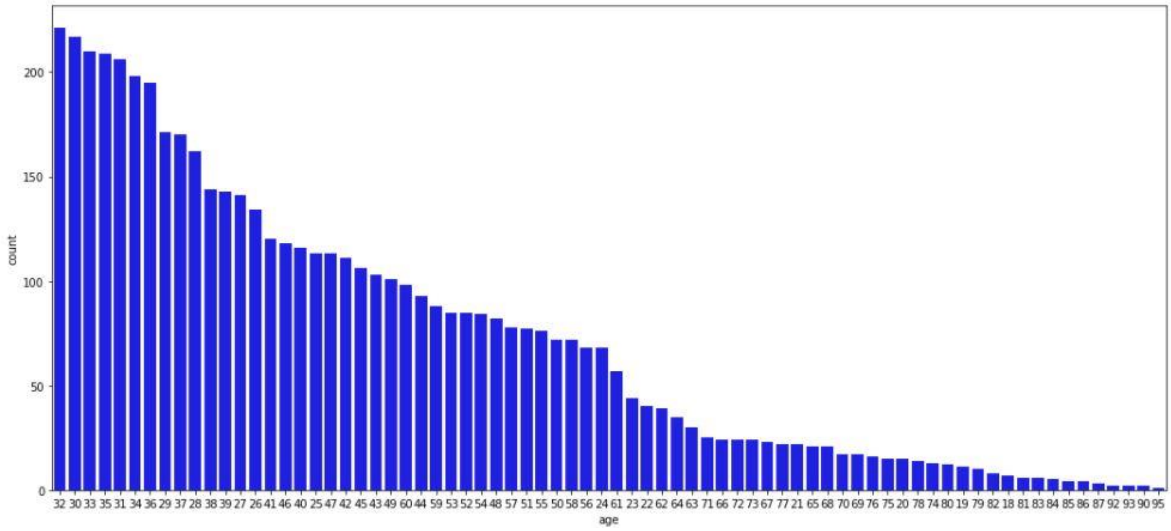
```
bankdf.select(avg($”balance”)).show()
spark.sql(“SELECT percentile_approx(balance, 0.5) FROM bank”).show()
```

Answer- Avg balance is 1362.27

Median balance is 448

## 5) Check if age matters in the marketing subscription for deposit-

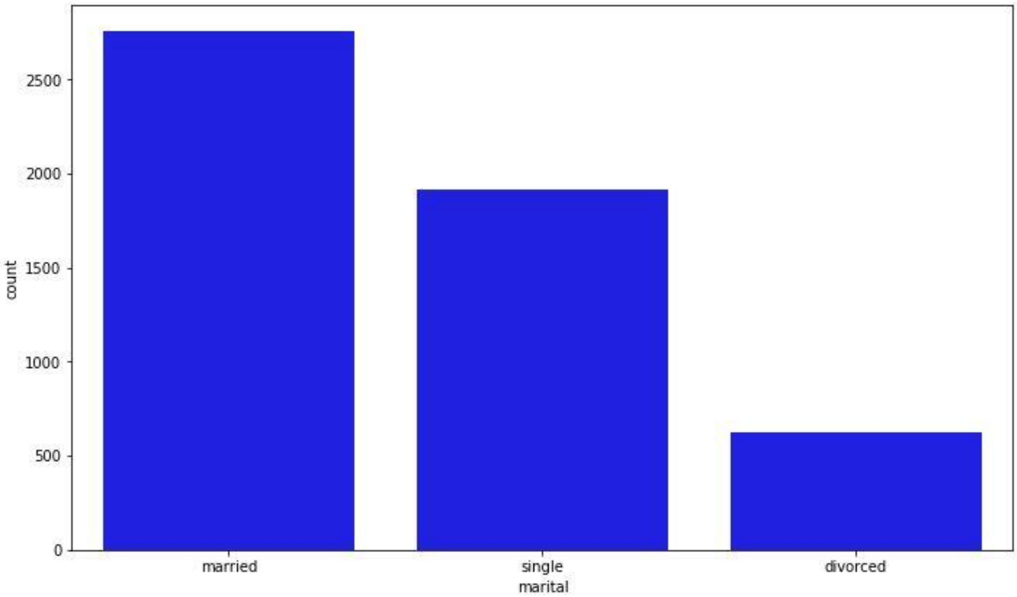
```
val age_target = bankdf.filter($”y” === “yes”).groupBy(“age”).count()
age_target.sort($”count”.desc).show()
```



Answer- we clearly see that age matters. most of the age data is lies between age 30 to 40

## 6) Check if marital status mattered for subscription to deposit-

```
val success_marital = bankdf.filter($"y" === "yes").groupBy("marital")
success_marital.count().show()
```



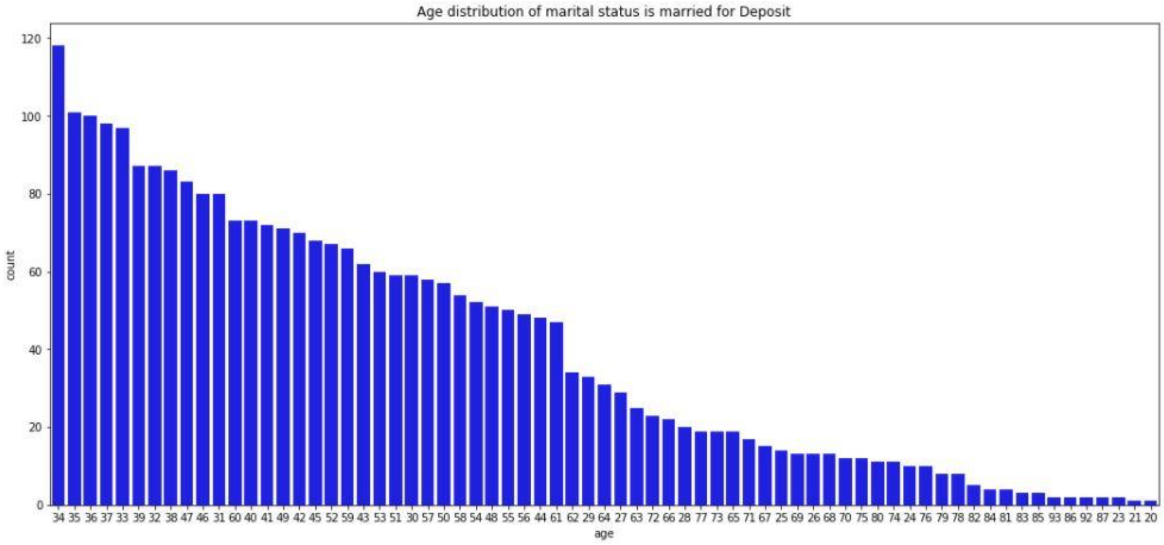
Answer- we see that married couples have most of the Subscription means marital matters and divorced have less subscription

Means in future  campaign our most of focus on married people

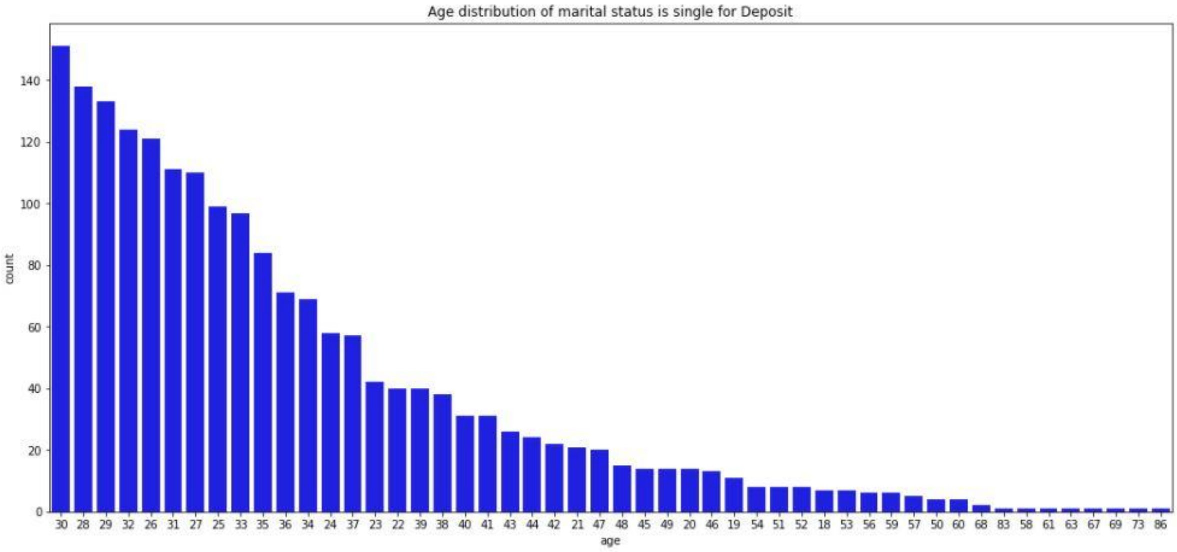
## 7) Check if age, marital status together mattered for subscription to deposit-

```
val success_age_marital_order_count = bankdf.filter($"y" === "yes").groupBy("marital", "age").count()
success_age_marital_order_count.sort($"count".desc).show()
```

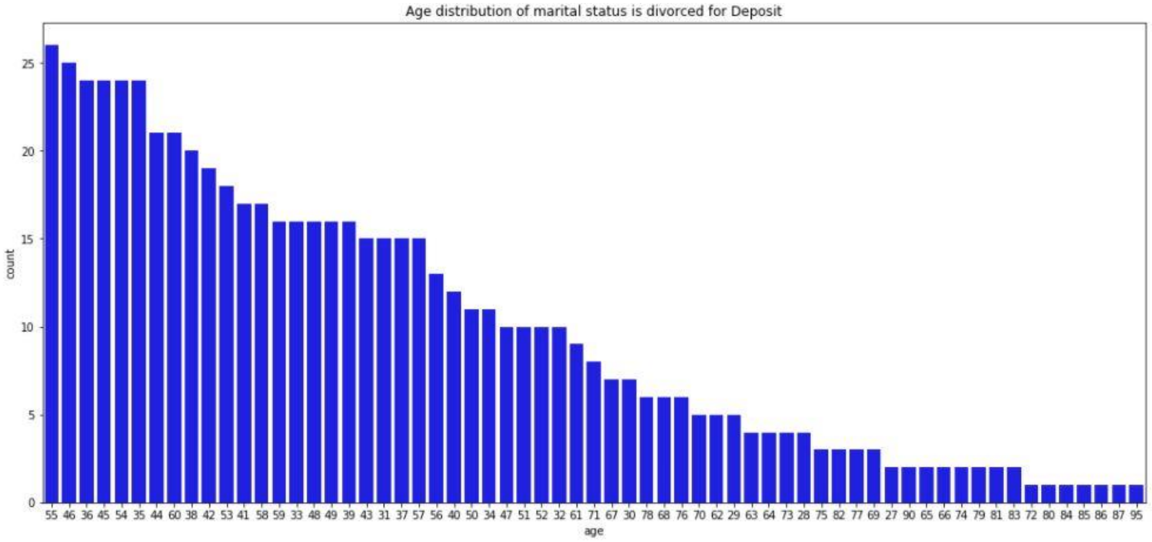
## Age distribution of marital status(married)-



## Age distribution of marital status(single)-



# Age distribution of marital status(divorced)-



Answer- we see that married peoples age between 32-38 have most of the subscriptions

Single peoples  age between 25-33 have most of the subscriptions

Divorced peoples age data not gives us clearify but approx age 41 to 55 they have most subscriptions

Married  groups have most of the subscription compare to single and divorced

Using this result we should target the customer in the future

## 8) Feature Engineering for column “Age” and find right age effect on campaign-

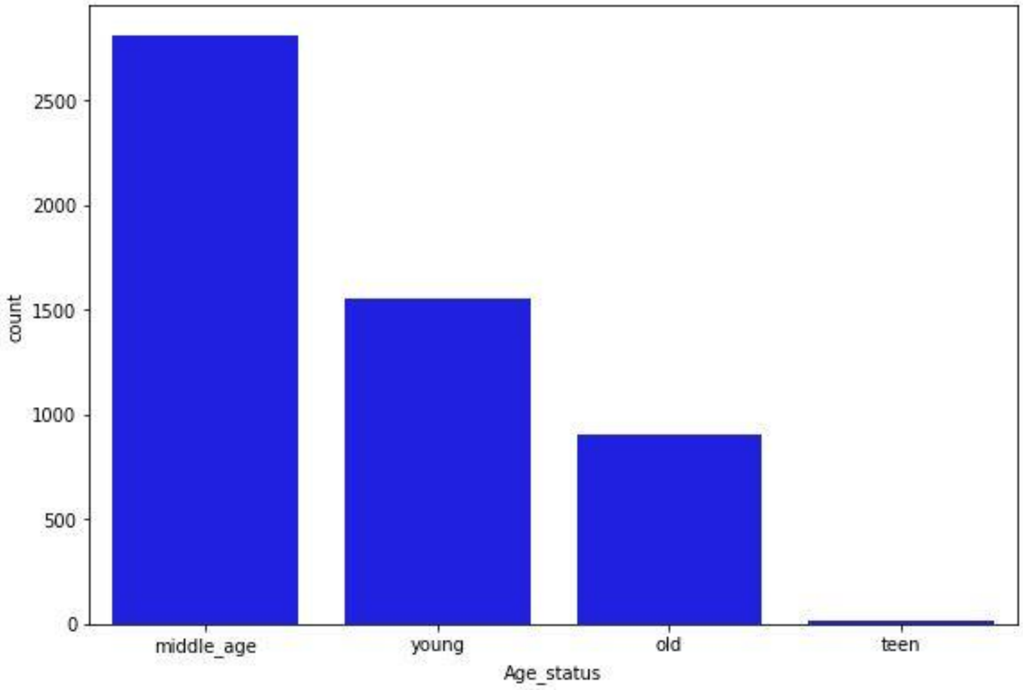
Using age column the data is divide into four groups(here we use udf)

```
val agerdd = sqlContext.udf.register(“agerdd” ,(age: Int) => {
if(age < 20)
“Teen”
else if(age >20 && age <= 32)
“Young”
else if(age > 33 && age <=55)
“Middle Aged”
else
“Old”
})
```

Replacing old column with new age column-

```
val banknewdf = bankdf.withColumn(“age”,agerdd(bankdf(“age”)))
banknewdf.regesterTempTable(“bank_new”)
banknewdf.createOrReplaceTempView(“bank_new”)
spark.table(“bank_new”).limit(5).show()

val age_target = banknewdf.filter($“y” === “yes”).groupBy(“age”)
age_target.count().show()
```



Answer- we see that middle_age customer between age 33-55 have most of the subscription who should be the targeted customers as they subscribe the most

Future campaign our main focus on  middle age or young people