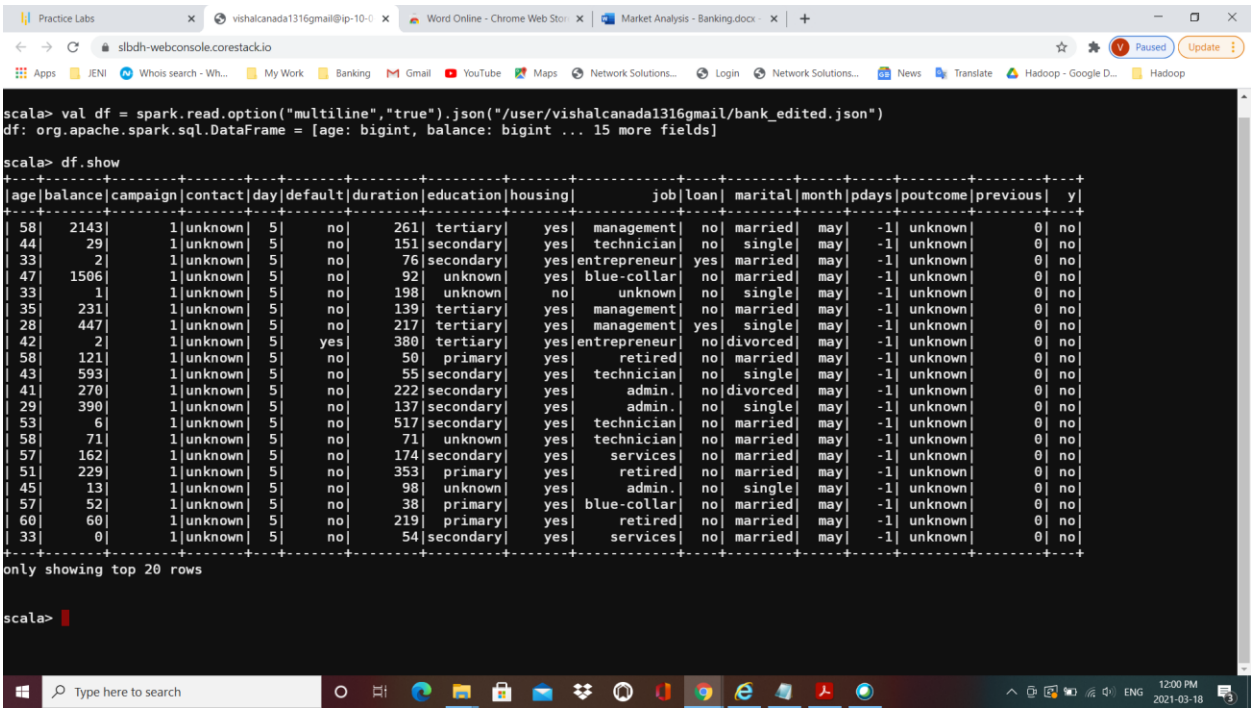
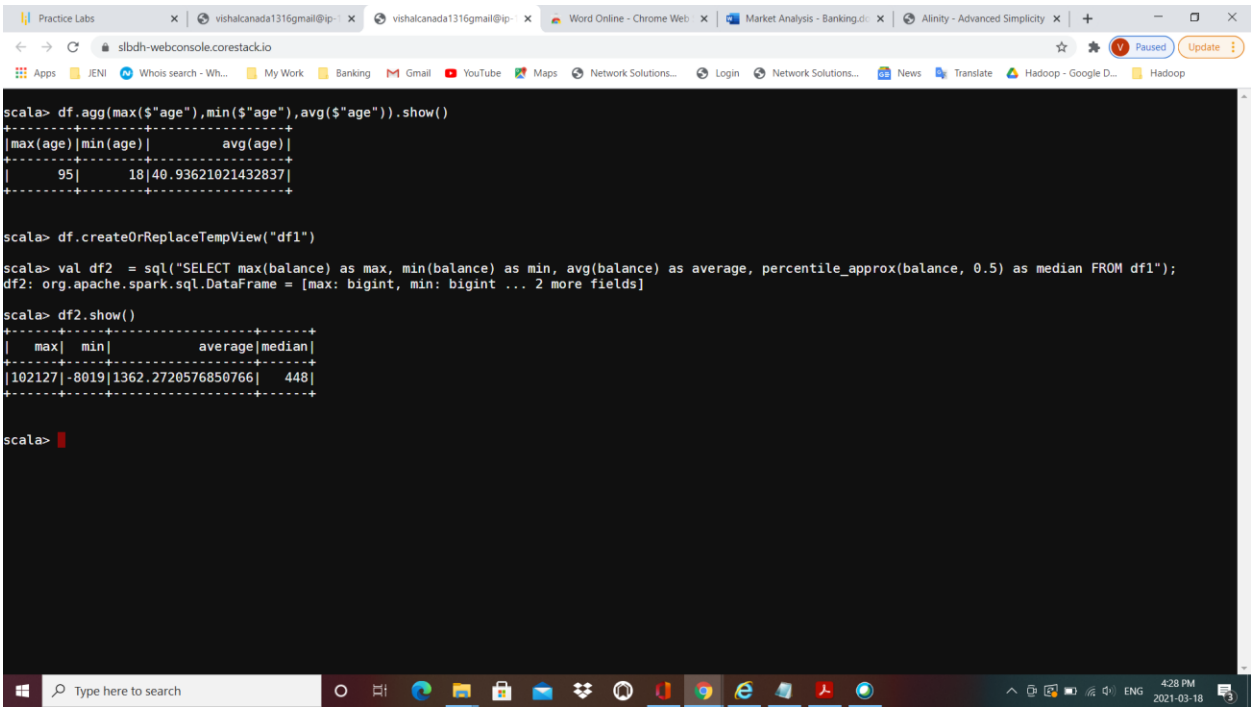
→ Uploaded Data in HDFS as bank_edited.json and Loaded Data (from HDFS) and Created Spark Dataframe



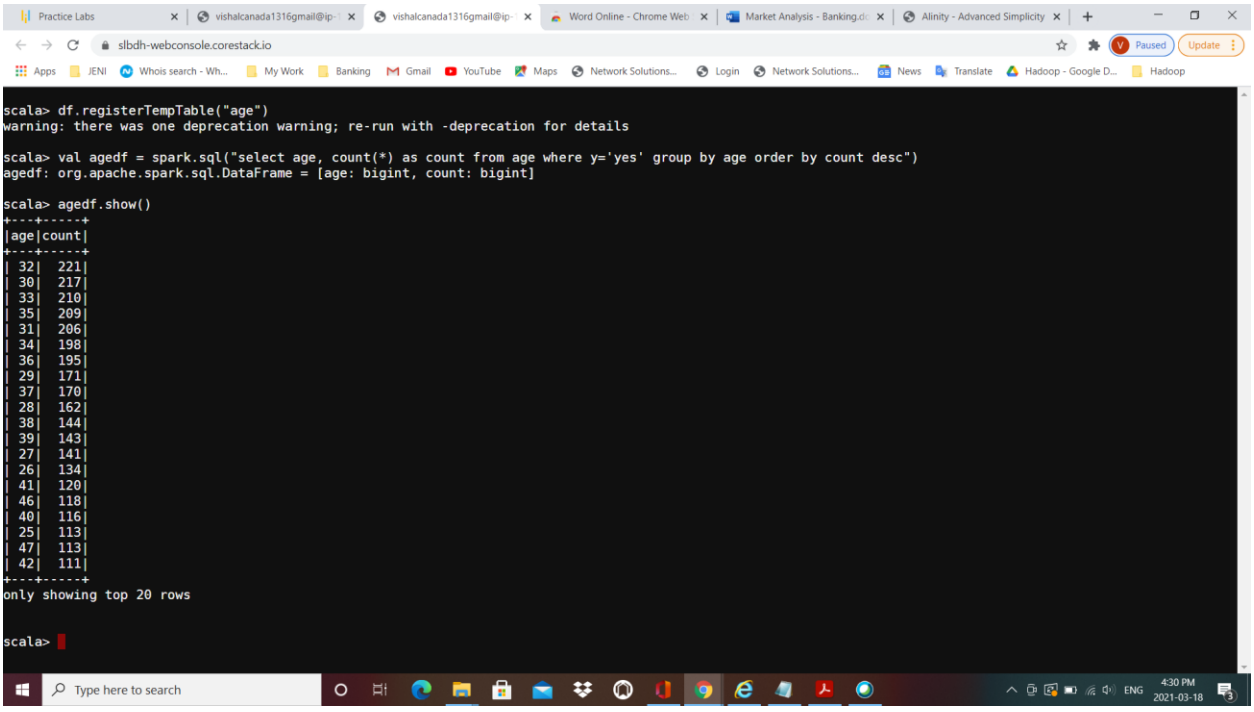
```
scala> val df = spark.read.option("multiline","true").json("/user/vishalcanada1316gmail/bank_edited.json")
df: org.apache.spark.sql.DataFrame = [age: bigint, balance: bigint ... 15 more fields]

scala> df.show
+---+-------+--------+-------+---+-------+--------+---------+-------+-----------+----+--------+-----+-----+--------+--------+---+
|age|balance|campaign|contact|day|default|duration|education|housing|        job|loan| marital|month|pdays|poutcome|previous|  y|
+---+-------+--------+-------+---+-------+--------+---------+-------+-----------+----+--------+-----+-----+--------+--------+---+
| 58|   2143|       1|unknown|  5|     no|     261| tertiary|    yes|  management|  no| married|  may|   -1| unknown|       0| no|
| 44|     29|       1|unknown|  5|     no|     151|secondary|    yes|  technician|  no|  single|  may|   -1| unknown|       0| no|
| 33|      2|       1|unknown|  5|     no|      76|secondary|    yes|entrepreneur| yes| married|  may|   -1| unknown|       0| no|
| 47|   1506|       1|unknown|  5|     no|      92|  unknown|    yes| blue-collar|  no| married|  may|   -1| unknown|       0| no|
| 33|      1|       1|unknown|  5|     no|     198|  unknown|     no|     unknown|  no|  single|  may|   -1| unknown|       0| no|
| 35|    231|       1|unknown|  5|     no|     139| tertiary|    yes|  management|  no| married|  may|   -1| unknown|       0| no|
| 28|    447|       1|unknown|  5|     no|     217| tertiary|    yes|  management| yes|  single|  may|   -1| unknown|       0| no|
| 42|      2|       1|unknown|  5|    yes|     380| tertiary|    yes|entrepreneur|  no|divorced|  may|   -1| unknown|       0| no|
| 58|    121|       1|unknown|  5|     no|      50|  primary|    yes|     retired|  no| married|  may|   -1| unknown|       0| no|
| 43|    593|       1|unknown|  5|     no|      55|secondary|    yes|  technician|  no|  single|  may|   -1| unknown|       0| no|
| 41|    270|       1|unknown|  5|     no|     222|secondary|    yes|      admin.|  no|divorced|  may|   -1| unknown|       0| no|
| 29|    390|       1|unknown|  5|     no|     137|secondary|    yes|      admin.|  no|  single|  may|   -1| unknown|       0| no|
| 53|      6|       1|unknown|  5|     no|     517|secondary|    yes|  technician|  no| married|  may|   -1| unknown|       0| no|
| 58|     71|       1|unknown|  5|     no|      71|  unknown|    yes|  technician|  no| married|  may|   -1| unknown|       0| no|
| 57|    162|       1|unknown|  5|     no|     174|secondary|    yes|    services|  no| married|  may|   -1| unknown|       0| no|
| 51|    229|       1|unknown|  5|     no|     353|  primary|    yes|     retired|  no| married|  may|   -1| unknown|       0| no|
| 45|     13|       1|unknown|  5|     no|      98|  unknown|    yes|      admin.|  no|  single|  may|   -1| unknown|       0| no|
| 57|     52|       1|unknown|  5|     no|      38|  primary|    yes| blue-collar|  no| married|  may|   -1| unknown|       0| no|
| 60|     60|       1|unknown|  5|     no|     219|  primary|    yes|     retired|  no| married|  may|   -1| unknown|       0| no|
| 33|      0|       1|unknown|  5|     no|      54|secondary|    yes|    services|  no| married|  may|   -1| unknown|       0| no|
+---+-------+--------+-------+---+-------+--------+---------+-------+-----------+----+--------+-----+-----+--------+--------+---+
only showing top 20 rows

scala>
```

→ Oldest Customer targeted is 95 years of age while the youngest one is 18 years of age and average or mean age of the customer is approx. 41 years

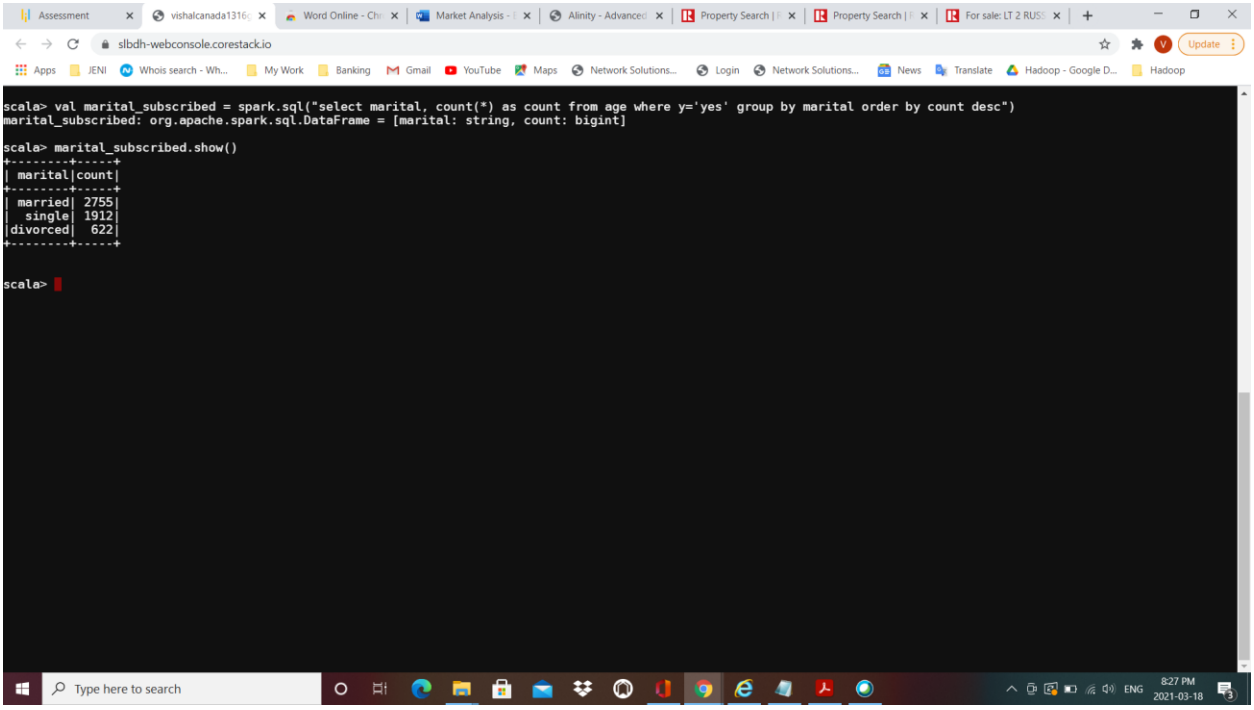
→ Max. Balance of customer is 102127 while Min. Balance is –8019 and Median balance is 448



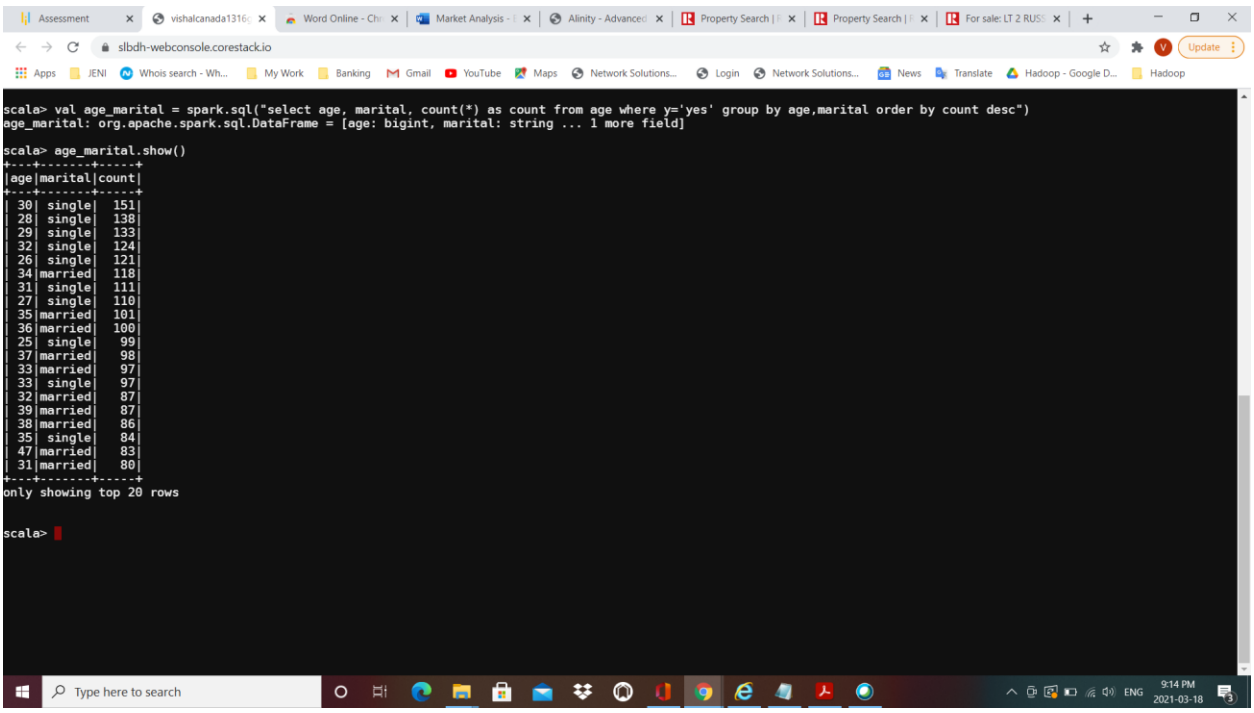
→ Age is dominating factor in subscription for term deposit as it is evident from the data that the interest is more from the younger age customers, compared to older age customers.



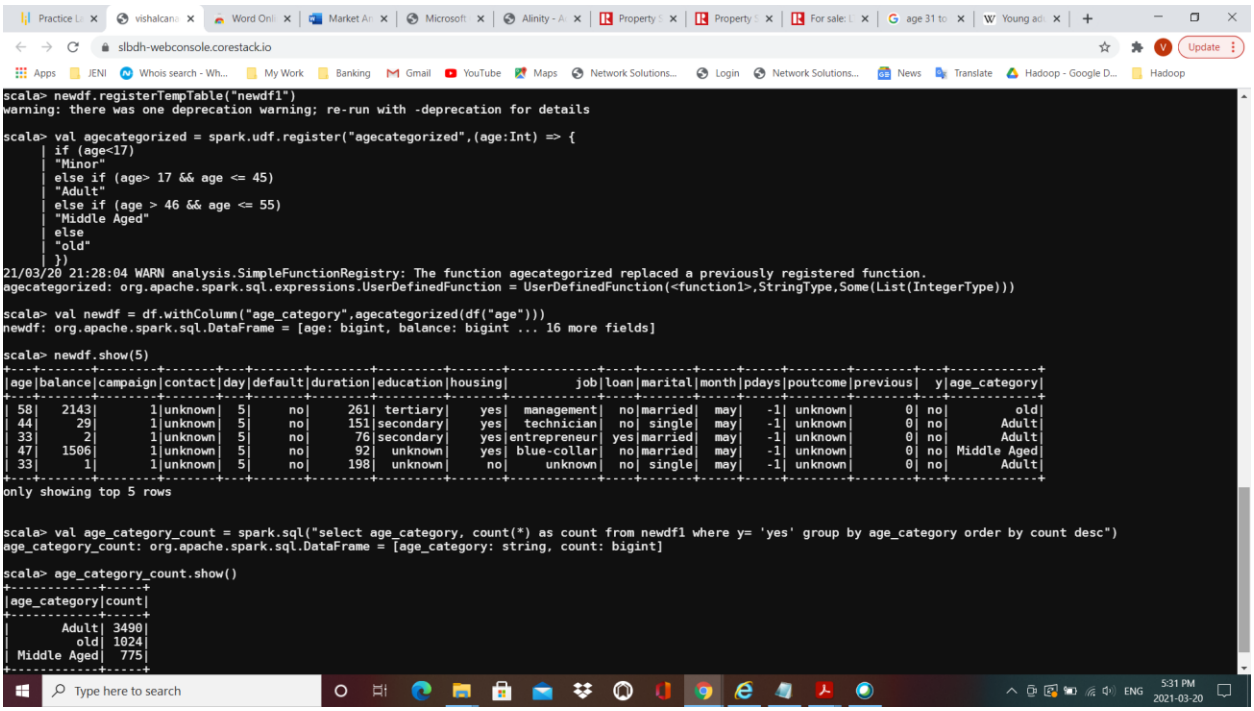
→ Looking at the data it can be concluded that more married customers subscribed that single customers for term deposit



→ Below table clearly suggests that age is prominent factor than marital status for subcrib



 Below analysis mentions that the subscribtion was highest amongst Adult age category


slbdh-webconsole.corestack.io
scala> newdf.registerTempTable("newdf1")
warning: there was one deprecation warning; re-run with -deprecation for details
scala> val agecategorized = spark.udf.register("agecategorized",(age:Int) => {
| if (age<17)
| "Minor"
| else if (age> 17 && age <= 45)
| "Adult"
| else if (age > 46 && age <= 55)
| "Middle Aged"
| else
| "old"
| })
21/03/20 21:28:04 WARN analysis.SimpleFunctionRegistry: The function agecategorized replaced a previously registered function.
agecategorized: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(IntegerType)))
scala> val newdf = df.withColumn("age_category",agecategorized(df("age")))
newdf: org.apache.spark.sql.DataFrame = [age: bigint, balance: bigint ... 16 more fields]
scala> newdf.show(5)
+---+-------+--------+-------+---+-------+--------+---------+-------+-----------+----+-------+-----+-----+--------+--------+---+-----------+
|age|balance|campaign|contact|day|default|duration|education|housing|job|loan|marital|month|pdays|poutcome|previous| y|age_category|
+---+-------+--------+-------+---+-------+--------+---------+-------+-----------+----+-------+-----+-----+--------+--------+---+-----------+
58	2143	1	unknown	5	no	261	tertiary	yes	management	no	married	may	-1	unknown	0	no	old
44	29	1	unknown	5	no	151	secondary	yes	technician	no	single	may	-1	unknown	0	no	Adult
33	2	1	unknown	5	no	76	secondary	yes	entrepreneur	yes	married	may	-1	unknown	0	no	Adult
47	1506	1	unknown	5	no	92	unknown	yes	blue-collar	no	married	may	-1	unknown	0	no	Middle Aged
33	1	1	unknown	5	no	198	unknown	no	unknown	no	single	may	-1	unknown	0	no	Adult
+---+-------+--------+-------+---+-------+--------+---------+-------+-----------+----+-------+-----+-----+--------+--------+---+-----------+
only showing top 5 rows
scala> val age_category_count = spark.sql("select age_category, count(*) as count from newdf1 where y= 'yes' group by age_category order by count desc")
age_category_count: org.apache.spark.sql.DataFrame = [age_category: string, count: bigint]
scala> age_category_count.show()
+-----------+-----+
|age_category|count|
+-----------+-----+
Adult	3490
old	1024
Middle Aged	775
+-----------+-----+
Type here to search
ENG
5:31 PM
2021-03-20