```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null


!wget -q https://dlcdn.apache.org/spark/spark-3.3.2/spark-3.3.2-bin-hadoop3.tgz
# Unzip the file
!tar xf spark-3.3.2-bin-hadoop3.tgz


import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = '/content/spark-3.3.2-bin-hadoop3'


# Install library for finding Spark
!pip install -q findspark
# Import the libary
import findspark
# Initiate findspark
findspark.init()
# Check the location for Spark
findspark.find()
```

```
'/content/spark-3.3.2-bin-hadoop3'
```

```
# Import SparkSession
from pyspark.sql import SparkSession
# Create a Spark Session
spark = SparkSession.builder.master("local[*]").getOrCreate()
# Check Spark Session Information
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.3.2
Master
local[*]
AppName
pyspark-shell

```
#Creating a dataframe with joined dataset
df = spark.read.csv("/content/Joined-data-final.csv", inferSchema=True, header=True)


df.show() #Displaying the data
```

```
+----+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+---
|year|tavg_jan|tavg_feb|tavg_mar|tavg_apr|tavg_may|tavg_jun|tavg_jul|tavg_aug|tavg_sep|tavg_oct|tavg_nov|tavg_dec|tav
+----+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+---
|1990|   24.36|   26.83|   28.66|   30.43|   29.63|    30.9|   29.48|   29.53|   28.93|   27.42|   26.51|    25.5|
|1991|    25.5|   25.08|   26.36|   28.24|   30.79|   31.99|    29.8|   30.23|   29.34|   27.97|   25.69|   24.95|
|1992|   23.74|   26.09|   27.24|   29.63|   31.24|   31.76|   30.44|   29.05|    29.1|   27.64|   26.06|   24.67|
|1993|   23.95|   25.49|   28.13|   29.82|   31.31|   30.96|   30.37|   29.25|   28.94|   27.23|   25.99|   24.83|
|1994|   24.67|   26.01|   27.77|   30.34|   31.62|    31.6|   29.82|   29.16|   29.66|   27.53|   25.21|   24.17|
|1995|   24.22|   26.07|   27.84|    30.1|   30.36|   31.75|   29.76|   29.01|   29.06|   27.75|   26.72|   24.87|
|1996|   24.77|   26.21|   27.59|   30.18|   32.09|   30.07|   29.69|   29.82|    28.5|   27.42|    26.3|   24.74|
|1997|   24.39|   25.57|    28.1|   29.62|   31.52|   31.72|   31.15|   30.79|   29.27|   27.79|    26.0|   26.38|
|1998|    26.4|   27.39|   28.78|   30.23|   32.33|   32.51|   30.36|   29.19|   29.64|   27.42|   26.68|   25.15|
|1999|   24.84|   26.25|   28.29|    30.5|   31.33|   31.17|   30.99|   29.46|   29.53|   28.03|   26.46|   24.73|
|2000|   26.34|   27.09|   27.72|   30.52|   31.76|   30.32|   29.55|   28.81|   28.87|   27.33|   26.56|   24.96|
|2001|   25.36|   26.66|   28.58|    29.5|    32.4|   30.98|   30.63|   29.81|   29.01|   27.64|   26.15|   24.61|
|2002|   25.01|   25.97|    28.1|   29.91|   32.27|   31.41|   31.54|   29.86|   29.57|    27.5|   25.76|   24.92|
|2003|   25.31|   26.67|   28.12|   30.37|   32.68|   32.26|   29.75|    29.2|   29.69|   27.65|   26.15|   25.09|
|2004|   24.73|   25.37|   27.94|   30.82|   29.15|   30.64|   29.98|    30.8|   28.45|   27.39|   26.39|   25.29|
|2005|   25.46|   26.32|    28.6|   28.64|   30.92|   31.88|    30.4|    30.1|   29.33|   26.26|   26.01|   24.17|
|2006|   24.54|   25.76|   28.46|   30.57|    31.4|   30.83|   30.75|   29.89|   29.04|    27.5|   25.41|   25.31|
|2007|   24.67|   25.75|   27.66|    29.6|   32.05|   29.72|   28.94|   28.26|   28.53|   27.43|   25.61|   24.85|
|2008|   24.64|   26.07|   26.74|   29.37|   31.99|    30.8|   30.04|   29.04|   29.03|   27.02|   25.37|   25.14|
|2009|   25.11|   26.61|   28.45|   30.82|    32.1|   32.34|   31.85|    29.6|   30.23|   29.72|   26.49|   25.15|
+----+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+---
only showing top 20 rows
```

```
#Displaying the schema of the data
df.printSchema()
```

```
root
 |-- year: integer (nullable = true)
```

```
 |-- tavg_jan: double (nullable = true)
 |-- tavg_feb: double (nullable = true)
 |-- tavg_mar: double (nullable = true)
 |-- tavg_apr: double (nullable = true)
 |-- tavg_may: double (nullable = true)
 |-- tavg_jun: double (nullable = true)
 |-- tavg_jul: double (nullable = true)
 |-- tavg_aug: double (nullable = true)
 |-- tavg_sep: double (nullable = true)
 |-- tavg_oct: double (nullable = true)
 |-- tavg_nov: double (nullable = true)
 |-- tavg_dec: double (nullable = true)
 |-- tavg_annual: double (nullable = true)
 |-- tavg_janfeb: double (nullable = true)
 |-- tavg_marmay: double (nullable = true)
 |-- tavg_junsep: double (nullable = true)
 |-- tavg_octdec: double (nullable = true)
 |-- rf_jan: double (nullable = true)
 |-- rf_feb: double (nullable = true)
 |-- rf_mar: double (nullable = true)
 |-- rf_apr: double (nullable = true)
 |-- rf_may: double (nullable = true)
 |-- rf_jun: double (nullable = true)
 |-- rf_jul: double (nullable = true)
 |-- rf_aug: double (nullable = true)
 |-- rf_sep: double (nullable = true)
 |-- rf_oct: double (nullable = true)
 |-- rf_nov: double (nullable = true)
 |-- rf_dec: double (nullable = true)
 |-- rf_annual: double (nullable = true)
 |-- rf_janfeb: double (nullable = true)
 |-- rf_marmay: double (nullable = true)
 |-- rf_junsep: double (nullable = true)
 |-- rf_octdec: double (nullable = true)
```

```
#Displaying data types of the data
df.dtypes
```

```
[('year', 'int'),
 ('tavg_jan', 'double'),
 ('tavg_feb', 'double'),
 ('tavg_mar', 'double'),
 ('tavg_apr', 'double'),
 ('tavg_may', 'double'),
 ('tavg_jun', 'double'),
 ('tavg_jul', 'double'),
 ('tavg_aug', 'double'),
 ('tavg_sep', 'double'),
 ('tavg_oct', 'double'),
 ('tavg_nov', 'double'),
 ('tavg_dec', 'double'),
 ('tavg_annual', 'double'),
 ('tavg_janfeb', 'double'),
 ('tavg_marmay', 'double'),
 ('tavg_junsep', 'double'),
 ('tavg_octdec', 'double'),
 ('rf_jan', 'double'),
 ('rf_feb', 'double'),
 ('rf_mar', 'double'),
 ('rf_apr', 'double'),
 ('rf_may', 'double'),
 ('rf_jun', 'double'),
 ('rf_jul', 'double'),
 ('rf_aug', 'double'),
 ('rf_sep', 'double'),
 ('rf_oct', 'double'),
 ('rf_nov', 'double'),
 ('rf_dec', 'double'),
 ('rf_annual', 'double'),
 ('rf_janfeb', 'double'),
 ('rf_marmay', 'double'),
 ('rf_junsep', 'double'),
 ('rf_octdec', 'double')]
```

```
#Importing VectorAssembler and Type cast
from pyspark.ml.feature import VectorAssembler
from pyspark.sql.types import IntegerType
```

```
#Displaying columns present in joined dataset
df.columns
```

```
['year',
 'tavg_jan',
 'tavg_feb',
```

```
      'tavg_mar',
      'tavg_apr',
      'tavg_may',
      'tavg_jun',
      'tavg_jul',
      'tavg_aug',
      'tavg_sep',
      'tavg_oct',
      'tavg_nov',
      'tavg_dec',
      'tavg_annual',
      'tavg_janfeb',
      'tavg_marmay',
      'tavg_junsep',
      'tavg_octdec',
      'rf_jan',
      'rf_feb',
      'rf_mar',
      'rf_apr',
      'rf_may',
      'rf_jun',
      'rf_jul',
      'rf_aug',
      'rf_sep',
      'rf_oct',
      'rf_nov',
      'rf_dec',
      'rf_annual',
      'rf_janfeb',
      'rf_marmay',
      'rf_junsep',
      'rf_octdec']


#Transforming rainfall columns from millimeters to centimeters
df = df.withColumn("rf_junsep", df["rf_junsep"] * 0.01)
df = df.withColumn("rf_octdec", df["rf_octdec"] * 0.01)
df = df.withColumn("rf_janfeb", df["rf_janfeb"] * 0.01)
df = df.withColumn("rf_marmay", df["rf_marmay"]*0.01)

#Changing the type of each column to integer
for col in df.columns:
  df = df.withColumn(col, df[col].cast(IntegerType()))



df.show() #Displaying transformed data
```

mar	rf_apr	rf_may	rf_jun	rf_jul	rf_aug	rf_sep	rf_oct	rf_nov	rf_dec	rf_annual	rf_janfeb	rf_marmay	rf_junsep	rf_octdec
36	24	94	28	40	80	119	194	144	46	903	0	1	2	3
9	37	29	128	54	71	114	225	234	21	954	0	0	3	4
0	22	58	66	77	59	157	123	297	52	918	0	0	3	4
9	11	46	68	60	88	96	214	315	163	1082	0	0	3	6
5	48	66	38	75	66	84	230	229	27	905	0	1	2	4
16	39	139	64	86	121	94	143	107	3	843	0	1	3	2
5	91	34	125	55	112	141	149	106	237	1069	0	1	4	4
2	42	49	50	63	56	116	169	258	135	950	0	0	2	5
4	23	55	43	101	152	121	124	242	197	1079	0	0	4	5
3	56	74	46	55	74	73	268	182	53	922	0	1	2	5
9	41	54	52	45	136	169	120	148	88	972	1	1	4	3
13	93	45	39	37	18	53	69	52	27	483	0	1	1	1
3	9	32	23	11	26	32	93	47	12	318	0	0	0	1
18	17	19	22	38	49	26	86	59	7	348	0	0	1	1
3	16	101	21	98	85	208	271	204	25	1037	0	1	4	5
24	128	80	35	87	93	117	280	353	148	1365	0	2	3	7
52	32	65	57	33	73	116	240	215	26	927	0	1	2	4
1	58	44	73	101	136	89	248	79	219	1067	0	1	3	5
164	31	53	51	73	126	70	242	298	50	1203	0	2	3	5
41	41	74	27	42	96	114	62	314	106	928	0	1	2	4

```
#Generating·vector·assembler·with·year·as·input·column·and·output·will·be·stored·in·features
assembler = VectorAssembler(inputCols=['year'], outputCol="features")


assembler
```

```
VectorAssembler_c0568463c936
```

```
#Adding features vector to dataframe
output = assembler.transform(df)


output.show(30)
```

```
+----+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+---
|year|tavg_jan|tavg_feb|tavg_mar|tavg_apr|tavg_may|tavg_jun|tavg_jul|tavg_aug|tavg_sep|tavg_oct|tavg_nov|tavg_dec|tav
+----+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+---
|1990|      24|      26|      28|      30|      29|      30|      29|      29|      28|      27|      26|      25|
|1991|      25|      25|      26|      28|      30|      31|      29|      30|      29|      27|      25|      24|
|1992|      23|      26|      27|      29|      31|      31|      30|      29|      29|      27|      26|      24|
|1993|      23|      25|      28|      29|      31|      30|      30|      29|      28|      27|      25|      24|
|1994|      24|      26|      27|      30|      31|      31|      29|      29|      29|      27|      25|      24|
|1995|      24|      26|      27|      30|      30|      31|      29|      29|      29|      27|      26|      24|
|1996|      24|      26|      27|      30|      32|      30|      29|      29|      28|      27|      26|      24|
|1997|      24|      25|      28|      29|      31|      31|      31|      30|      29|      27|      26|      26|
|1998|      26|      27|      28|      30|      32|      32|      30|      29|      29|      27|      26|      25|
|1999|      24|      26|      28|      30|      31|      31|      30|      29|      29|      28|      26|      24|
|2000|      26|      27|      27|      30|      31|      30|      29|      28|      28|      27|      26|      24|
|2001|      25|      26|      28|      29|      32|      30|      30|      29|      29|      27|      26|      24|
|2002|      25|      25|      28|      29|      32|      31|      31|      29|      29|      27|      25|      24|
|2003|      25|      26|      28|      30|      32|      32|      29|      29|      29|      27|      26|      25|
|2004|      24|      25|      27|      30|      29|      30|      29|      30|      28|      27|      26|      25|
|2005|      25|      26|      28|      28|      30|      31|      30|      30|      29|      26|      26|      24|
|2006|      24|      25|      28|      30|      31|      30|      30|      29|      29|      27|      25|      25|
|2007|      24|      25|      27|      29|      32|      29|      28|      28|      28|      27|      25|      24|
|2008|      24|      26|      26|      29|      31|      30|      30|      29|      29|      27|      25|      25|
|2009|      25|      26|      28|      30|      32|      32|      31|      29|      30|      29|      26|      25|
|2010|      25|      26|      28|      31|      31|      30|      29|      28|      28|      28|      26|      24|
|2011|      24|      25|      27|      29|      31|      31|      30|      29|      29|      28|      26|      25|
|2012|      24|      25|      28|      30|      33|      32|      30|      29|      29|      27|      26|      25|
|2013|      25|      26|      28|      30|      31|      31|      29|      29|      28|      28|      26|      24|
|2014|      25|      25|      27|      30|      31|      32|      30|      29|      28|      28|      26|      25|
|2015|      25|      26|      28|      30|      31|      31|      31|      30|      30|      28|      26|      26|
+----+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+---
```

```
#Creating dataframe for each target column which will be used to train and test
tavg_janfeb_model_df = output.select("tavg_janfeb","features")
tavg_marmay_model_df = output.select("tavg_marmay","features")
tavg_junsep_model_df = output.select("tavg_junsep","features")
tavg_octdec_model_df = output.select("tavg_octdec","features")
rf_janfeb_model_df = output.select("rf_janfeb","features")
rf_marmay_model_df = output.select("rf_marmay", "features")
rf_junsep_model_df = output.select("rf_junsep","features")
rf_octdec_model_df = output.select("rf_octdec","features")


#Displaying dataframes created above
tavg_janfeb_model_df.show()
tavg_marmay_model_df.show()
tavg_junsep_model_df.show()
tavg_octdec_model_df.show()
rf_janfeb_model_df.show()
rf_marmay_model_df.show()
rf_junsep_model_df.show()
rf_octdec_model_df.show()
```

```
|        3|[2007.0]|
|        3|[2008.0]|
|        2|[2009.0]|
+---------+--------+
only showing top 20 rows

+---------+--------+
|rf_octdec|features|
+---------+--------+
|        3|[1990.0]|
|        4|[1991.0]|
|        4|[1992.0]|
|        6|[1993.0]|
|        4|[1994.0]|
|        2|[1995.0]|
|        4|[1996.0]|
|        5|[1997.0]|
|        5|[1998.0]|
|        5|[1999.0]|
|        3|[2000.0]|
|        1|[2001.0]|
|        1|[2002.0]|
|        1|[2003.0]|
|        5|[2004.0]|
|        7|[2005.0]|
|        4|[2006.0]|
|        5|[2007.0]|
|        5|[2008.0]|
|        4|[2009.0]|
+---------+--------+
only showing top 20 rows
```

```
#Displaying data type of created training dataframes
tavg_janfeb_model_df.dtypes
tavg_marmay_model_df.dtypes
tavg_junsep_model_df.dtypes
tavg_octdec_model_df.dtypes
rf_janfeb_model_df.dtypes
rf_marmay_model_df.dtypes
rf_junsep_model_df.dtypes
rf_octdec_model_df.dtypes
```

```
[('rf_octdec', 'int'), ('features', 'vector')]
```

```
#Splitting data in each training dataframe into taining and test data
tavg_janfeb_training_df, tavg_janfeb_test_df = tavg_janfeb_model_df.randomSplit([0.7, 0.3])
tavg_marmay_training_df, tavg_marmay_test_df = tavg_marmay_model_df.randomSplit([0.7, 0.3])
tavg_junsep_training_df, tavg_junsep_test_df = tavg_junsep_model_df.randomSplit([0.6, 0.4])
tavg_octdec_training_df, tavg_octdec_test_df = tavg_octdec_model_df.randomSplit([0.6, 0.4])
training_rf_janfeb, test_rf_janfeb = rf_janfeb_model_df.randomSplit([0.7, 0.3])
training_rf_marmay, test_rf_marmay = rf_marmay_model_df.randomSplit([0.7, 0.3])
rf_junsep_training_df, rf_junsep_test_df = rf_junsep_model_df.randomSplit([0.7, 0.3])
rf_octdec_training_df, rf_octdec_test_df = rf_octdec_model_df.randomSplit([0.7, 0.3])
```

```
#Displaying cout of training data
tavg_janfeb_training_df.count()
tavg_marmay_training_df.count()
tavg_junsep_training_df.count()
tavg_octdec_training_df.count()
training_rf_janfeb.count()
training_rf_marmay.count()
rf_junsep_training_df.count()
rf_octdec_training_df.count()
```

```
18
```

```
#displaying count of test data
tavg_janfeb_test_df.count()
tavg_marmay_test_df.count()
tavg_junsep_test_df.count()
tavg_octdec_test_df.count()
test_rf_janfeb.count()
test_rf_marmay.count()
rf_junsep_test_df.count()
rf_octdec_test_df.count()
```

```
8
```

```
#Importing Ml models
from pyspark.ml import Pipeline
```

```
from pyspark.ml.feature import StringIndexer, VectorIndexer, IndexToString
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.evaluation import MulticlassMetrics
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.regression import LinearRegression
```

Linear Regression

```
#Creating instance of linear regression model for each target column
tavg_janfeb_lr = LinearRegression(featuresCol= "features", labelCol="tavg_janfeb")
tavg_marmay_lr = LinearRegression(featuresCol= "features", labelCol="tavg_marmay")
tavg_junsep_lr = LinearRegression(featuresCol= "features", labelCol="tavg_junsep")
tavg_octdec_lr = LinearRegression(featuresCol= "features", labelCol="tavg_octdec")
lr_janfeb = LinearRegression(featuresCol= "features", labelCol="rf_janfeb")
lr_marmay = LinearRegression(featuresCol= "features", labelCol="rf_marmay")
rf_junsep_lr = LinearRegression(featuresCol= "features", labelCol="rf_junsep")
rf_octdec_lr = LinearRegression(featuresCol= "features", labelCol="rf_octdec")
```

```
#Fitting training data on models created above
tavg_janfeb_trained_model = tavg_janfeb_lr.fit(tavg_janfeb_training_df)
tavg_marmay_trained_model = tavg_marmay_lr.fit(tavg_marmay_training_df)
tavg_junsep_trained_model = tavg_junsep_lr.fit(tavg_junsep_training_df)
tavg_octdec_trained_model = tavg_octdec_lr.fit(tavg_octdec_training_df)
trained_model_janfeb = lr_janfeb.fit(training_rf_janfeb)
trained_model_marmay = lr_marmay.fit(training_rf_marmay)
rf_junsep_trained_model = rf_junsep_lr.fit(rf_junsep_training_df)
rf_octdec_trained_model = rf_octdec_lr.fit(rf_octdec_training_df)
```

```
#Evaluating linear regression models
tavg_janfeb_results = tavg_janfeb_trained_model.evaluate(tavg_janfeb_training_df)
tavg_marmay_results = tavg_marmay_trained_model.evaluate(tavg_marmay_training_df)
results_tavg_junsep = tavg_junsep_trained_model.evaluate(tavg_junsep_training_df)
results_tavg_octdec = tavg_octdec_trained_model.evaluate(tavg_octdec_training_df)
results_rf_janfeb = trained_model_janfeb.evaluate(training_rf_janfeb)
results_rf_marmay = trained_model_marmay .evaluate(training_rf_marmay)
rf_junsep_results = rf_junsep_trained_model.evaluate(rf_junsep_training_df)
rf_octdec_results = rf_octdec_trained_model.evaluate(rf_octdec_training_df)
```

```
#Displaying R squared value
print("R squared for tavg janfeb",tavg_janfeb_results.r2)
print("R squared for tavg marmay",tavg_marmay_results.r2)
print("R squared for tavg junsep",results_tavg_junsep.r2)
print("R squared for tavg octdec",results_tavg_octdec.r2)
print(rf_junsep_results.r2)
print(rf_octdec_results.r2)
print(results_rf_janfeb.r2)
print(results_rf_marmay.r2)
```

```
R squared for tavg janfeb 0.097355300143393
R squared for tavg marmay 0.28947210988537864
R squared for tavg junsep 0.008325360015568162
R squared for tavg octdec 0.13965231190939598
0.009036388922960326
0.004980823053798655
nan
0.05998737657288544
```

```
#Displaying mean square error for linear regression models created
print("Mean square error for tavg_janfeb: ", tavg_janfeb_results.meanSquaredError)
print("Mean square error for tavg_marmay: ", tavg_marmay_results.meanSquaredError)
print("Mean square error for tavg_junsep: ", results_tavg_junsep.meanSquaredError)
print("Mean square error for tavg_octdec: ", results_tavg_octdec.meanSquaredError)
print("Mean square error of rf_janfeb: ", results_rf_janfeb.meanSquaredError)
print("Mean square error of rf_marmay: ", results_rf_marmay.meanSquaredError)
print("Mean square error for rf_junsep: ", rf_junsep_results.meanSquaredError)
print("Mean square error for rf_octdec: ", rf_octdec_results.meanSquaredError)
```

```
Mean square error for tavg_janfeb:  0.2234045632145097
Mean square error for tavg_marmay:  0.1732034745985774
Mean square error for tavg_junsep:  0.42311451306002407
Mean square error for tavg_octdec:  0.21508692202265076
Mean square error of rf_janfeb:  0.0
Mean square error of rf_marmay:  0.41352029239197324
Mean square error for rf_junsep:  1.1726974221050088
Mean square error for rf_octdec:  2.162016977068289
```

```
#Selecting features of each test data
tavg_janfeb_unlabeled_data = tavg_janfeb_test_df.select("features")
tavg_janfeb_unlabeled_data.show()
tavg_marmay_unlabeled_data = tavg_marmay_test_df.select("features")
tavg_marmay_unlabeled_data.show()
tavg_junsep_unlabeled_data = tavg_junsep_test_df.select("features")
tavg_junsep_unlabeled_data.show()
tavg_octdec_unlabeled_data = tavg_octdec_test_df.select("features")
tavg_octdec_unlabeled_data.show()
rf_janfeb_unlabeled_data = test_rf_janfeb.select("features")
rf_marmay_unlabeled_data = test_rf_marmay.select("features")
rf_janfeb_unlabeled_data.show()
rf_marmay_unlabeled_data.show()
rf_junsep_unlabeled_data = rf_junsep_test_df.select("features")
rf_junsep_unlabeled_data.show()
rf_octdec_unlabeled_data = rf_octdec_test_df.select("features")
rf_octdec_unlabeled_data.show()
```

```
     |[1998.0]|
     |[1999.0]|
     |[2001.0]|
     |[2003.0]|
     |[2012.0]|
     +--------+

     +--------+
     |features|
     +--------+
     |[1990.0]|
     |[1994.0]|
     |[1996.0]|
     |[2001.0]|
     |[2007.0]|
     |[2008.0]|
     |[2010.0]|
     |[2012.0]|
     |[2000.0]|
     +--------+

     +--------+
     |features|
     +--------+
     |[1991.0]|
     |[1993.0]|
     |[1995.0]|
     |[2004.0]|
     |[2009.0]|
     +--------+

     +--------+
     |features|
     +--------+
     |[1997.0]|
     |[1999.0]|
     |[1991.0]|
     |[2005.0]|
     |[2007.0]|
     |[2008.0]|
     |[2010.0]|
     |[2011.0]|
     |[1996.0]|
     +--------+

     +--------+
     |features|
     +--------+
     |[2001.0]|
     |[2002.0]|
     |[2012.0]|
     |[2000.0]|
     |[1994.0]|
     |[2014.0]|
     |[1999.0]|
     |[2015.0]|
     +--------+
```

```
#Trasforming test data on trained models to get predictions
tavg_janfeb_predictions = tavg_janfeb_trained_model.transform(tavg_janfeb_unlabeled_data)
tavg_marmay_predictions = tavg_marmay_trained_model.transform(tavg_marmay_unlabeled_data)
tavg_junsep_predictions = tavg_junsep_trained_model.transform(tavg_junsep_unlabeled_data)
tavg_octdec_predictions = tavg_octdec_trained_model.transform(tavg_octdec_unlabeled_data)
rf_janfeb_predictions = trained_model_janfeb.transform(rf_janfeb_unlabeled_data)
rf_marmay_predictions = trained_model_marmay.transform(rf_marmay_unlabeled_data)
```

```
rf_junsep_predictions = rf_junsep_trained_model.transform(rf_junsep_unlabeled_data)
rf_octdec_predictions = rf_octdec_trained_model.transform(rf_octdec_unlabeled_data)


#Displaying predictions
tavg_janfeb_predictions.show()
tavg_marmay_predictions.show()
tavg_junsep_predictions.show()
tavg_octdec_predictions.show()
rf_janfeb_predictions.show()
rf_marmay_predictions.show()
rf_junsep_predictions.show()
rf_octdec_predictions.show()
```

```
|[1998.0]|25.842891149100772|
|[1999.0]| 25.86828247853903|
|[2001.0]| 25.91906513741555|
|[2003.0]|25.969847796292065|
|[2012.0]|  26.1983697612364|
+--------+------------------+

+--------+----------+
|features|prediction|
+--------+----------+
|[1990.0]|       0.0|
|[1994.0]|       0.0|
|[1996.0]|       0.0|
|[2001.0]|       0.0|
|[2007.0]|       0.0|
|[2008.0]|       0.0|
|[2010.0]|       0.0|
|[2012.0]|       0.0|
|[2000.0]|       0.0|
+--------+----------+

+--------+------------------+
|features|        prediction|
+--------+------------------+
|[1991.0]| 0.531909462215836|
|[1993.0]| 0.576412525524745|
|[1995.0]| 0.620915588833661|
|[2004.0]|0.8211793737237585|
|[2009.0]| 0.932437031996038|
+--------+------------------+

+--------+------------------+
|features|        prediction|
+--------+------------------+
|[1997.0]| 2.482632338253566|
|[1999.0]| 2.456442126290291|
|[1991.0]| 2.561202974143395|
|[2005.0]| 2.377871490400466|
|[2007.0]|2.3516812784371908|
|[2008.0]|2.3385861724555532|
|[2010.0]| 2.312395960492278|
|[2011.0]|2.2993008545106406|
|[1996.0]|2.4957274442352073|
+--------+------------------+

+--------+------------------+
|features|        prediction|
+--------+------------------+
|[2001.0]| 4.214367160775598|
|[2002.0]|4.2285062713795405|
|[2012.0]| 4.369897377418972|
|[2000.0]| 4.200228050171656|
|[1994.0]| 4.115393386547996|
|[2014.0]|  4.39817559862686|
|[1999.0]|4.1860889395677106|
|[2015.0]| 4.412314709230802|
+--------+------------------+
```

Decision Tree

```
#Creating instances of decision tree classifier for targeted columns training data
tavg_janfeb_df_classifier = DecisionTreeClassifier(labelCol="tavg_janfeb").fit(tavg_janfeb_training_df)
tavg_marmay_df_classifier = DecisionTreeClassifier(labelCol="tavg_marmay").fit(tavg_marmay_training_df)
tavg_junsep_df_classifier = DecisionTreeClassifier(labelCol="tavg_junsep").fit(tavg_junsep_training_df)
tavg_octdec_df_classifier = DecisionTreeClassifier(labelCol="tavg_octdec").fit(tavg_octdec_training_df)
rf_janfeb_df_classifier = DecisionTreeClassifier(labelCol="rf_janfeb").fit(training_rf_janfeb)
rf_marmay_df_classifier = DecisionTreeClassifier(labelCol="rf_marmay").fit(training_rf_marmay)
```

```
rf_junsep_classifier = DecisionTreeClassifier(labelCol="rf_junsep").fit(rf_junsep_training_df)
rf_octdec_classifier = DecisionTreeClassifier(labelCol="rf_octdec").fit(rf_octdec_training_df)


#Transforming decision tree models on respective test data
tavg_janfeb_df_predictions = tavg_janfeb_df_classifier.transform(tavg_janfeb_test_df)
tavg_marmay_df_predictions = tavg_marmay_df_classifier.transform(tavg_marmay_test_df)
tavg_junsep_df_predictions = tavg_junsep_df_classifier.transform(tavg_junsep_test_df)
tavg_octdec_df_predictions = tavg_octdec_df_classifier.transform(tavg_octdec_test_df)
df_predictions_janfeb = rf_janfeb_df_classifier.transform(test_rf_janfeb)
df_predictions_marmay = rf_marmay_df_classifier.transform(test_rf_marmay)
rf_junsep_predictions = rf_junsep_classifier.transform(rf_junsep_test_df)
rf_octdec_predictions = rf_octdec_classifier.transform(rf_octdec_test_df)


#Displaying predictions
tavg_janfeb_df_predictions.show()
tavg_marmay_df_predictions.show()
tavg_junsep_df_predictions.show()
tavg_octdec_df_predictions.show()
df_predictions_janfeb.show()
df_predictions_marmay.show()
rf_junsep_predictions.show()
rf_octdec_predictions.show()
```

```
|         26|[1998.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|      26.0|
|         26|[1999.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|      26.0|
|         26|[2001.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|      26.0|
|         26|[2003.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|      26.0|
|         26|[2012.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|      26.0|
+-----------+--------+--------------------+--------------------+----------+

+---------+--------+-------------+-----------+----------+
|rf_janfeb|features|rawPrediction|probability|prediction|
+---------+--------+-------------+-----------+----------+
|        0|[1990.0]|       [17.0]|      [1.0]|       0.0|
|        0|[1994.0]|       [17.0]|      [1.0]|       0.0|
|        0|[1996.0]|       [17.0]|      [1.0]|       0.0|
|        0|[2001.0]|       [17.0]|      [1.0]|       0.0|
|        0|[2007.0]|       [17.0]|      [1.0]|       0.0|
|        0|[2008.0]|       [17.0]|      [1.0]|       0.0|
|        0|[2010.0]|       [17.0]|      [1.0]|       0.0|
|        0|[2012.0]|       [17.0]|      [1.0]|       0.0|
|        1|[2000.0]|       [17.0]|      [1.0]|       0.0|
+---------+--------+-------------+-----------+----------+

+---------+--------+-------------+-------------+----------+
|rf_marmay|features|rawPrediction|  probability|prediction|
+---------+--------+-------------+-------------+----------+
|        0|[1991.0]|[3.0,3.0,0.0]|[0.5,0.5,0.0]|       0.0|
|        0|[1993.0]|[3.0,3.0,0.0]|[0.5,0.5,0.0]|       0.0|
|        1|[1995.0]|[3.0,3.0,0.0]|[0.5,0.5,0.0]|       0.0|
|        1|[2004.0]|[2.0,0.0,0.0]|[1.0,0.0,0.0]|       0.0|
|        1|[2009.0]|[0.0,4.0,1.0]|[0.0,0.8,0.2]|       1.0|
+---------+--------+-------------+-------------+----------+

+---------+--------+--------------------+--------------------+----------+
|rf_junsep|features|       rawPrediction|         probability|prediction|
+---------+--------+--------------------+--------------------+----------+
|        2|[1997.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|       4.0|
|        2|[1999.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|       4.0|
|        3|[1991.0]|[0.0,0.0,1.0,0.0,...|[0.0,0.0,1.0,0.0,...|       2.0|
|        3|[2005.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|       4.0|
|        3|[2007.0]|[0.0,0.0,3.0,0.0,...|[0.0,0.0,1.0,0.0,...|       2.0|
|        3|[2008.0]|[0.0,0.0,3.0,0.0,...|[0.0,0.0,1.0,0.0,...|       2.0|
|        3|[2010.0]|[0.0,0.0,3.0,0.0,...|[0.0,0.0,1.0,0.0,...|       2.0|
|        3|[2011.0]|[0.0,0.0,3.0,0.0,...|[0.0,0.0,1.0,0.0,...|       2.0|
|        4|[1996.0]|[0.0,0.0,0.0,1.0,...|[0.0,0.0,0.0,1.0,...|       3.0|
+---------+--------+--------------------+--------------------+----------+

+---------+--------+--------------------+--------------------+----------+
|rf_octdec|features|       rawPrediction|         probability|prediction|
+---------+--------+--------------------+--------------------+----------+
|        1|[2001.0]|[0.0,1.0,0.0,0.0,...|[0.0,0.125,0.0,0....|       5.0|
|        1|[2002.0]|[0.0,1.0,0.0,0.0,...|[0.0,0.125,0.0,0....|       5.0|
|        2|[2012.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|       4.0|
|        3|[2000.0]|[0.0,1.0,0.0,0.0,...|[0.0,0.125,0.0,0....|       5.0|
|        4|[1994.0]|[0.0,0.0,0.0,0.0,...|[0.0,0.0,0.0,0.0,...|       6.0|
|        4|[2014.0]|[0.0,0.0,1.0,0.0,...|[0.0,0.0,1.0,0.0,...|       2.0|
|        5|[1999.0]|[0.0,1.0,0.0,0.0,...|[0.0,0.125,0.0,0....|       5.0|
|        6|[2015.0]|[0.0,0.0,1.0,0.0,...|[0.0,0.0,1.0,0.0,...|       2.0|
+---------+--------+--------------------+--------------------+----------+
```

```
#Evaluating accuracy of decision tree models
tavg_janfeb_df_accuracy = MulticlassClassificationEvaluator(labelCol="tavg_janfeb", metricName="accuracy").evaluate(tavg_j
```

```
print("tavg_janfeb_df_accuracy: ", tavg_janfeb_df_accuracy)
tavg_marmay_df_accuracy = MulticlassClassificationEvaluator(labelCol="tavg_marmay", metricName="accuracy").evaluate(tavg_m
print("tavg_marmay_df_accuracy: ", tavg_marmay_df_accuracy)
tavg_junsep_df_accuracy = MulticlassClassificationEvaluator(labelCol="tavg_junsep", metricName="accuracy").evaluate(tavg_j
print("tavg_junsep Accuracy: ", tavg_junsep_df_accuracy)
tavg_octdec_df_accuracy = MulticlassClassificationEvaluator(labelCol="tavg_octdec", metricName="accuracy").evaluate(tavg_o
print("tavg_octdec Accuracy: ", tavg_octdec_df_accuracy)
df_accuracy_janfeb = MulticlassClassificationEvaluator(labelCol="rf_janfeb", metricName="accuracy").evaluate(df_prediction
df_accuracy_marmay = MulticlassClassificationEvaluator(labelCol="rf_marmay", metricName="accuracy").evaluate(df_prediction
print("Accuracy: ", df_accuracy_janfeb)
print("Accuracy: ", df_accuracy_marmay)
rf_junsep_accuracy = MulticlassClassificationEvaluator(labelCol="rf_junsep", metricName="accuracy").evaluate(rf_junsep_pre
print("Accuracy: ", rf_junsep_accuracy)
rf_octdec_accuracy = MulticlassClassificationEvaluator(labelCol="rf_octdec", metricName="accuracy").evaluate(rf_octdec_pre
print("Accuracy: ", rf_octdec_accuracy)
```

```
tavg_janfeb_df_accuracy:  0.6666666666666666
tavg_marmay_df_accuracy:  0.5714285714285714
tavg_junsep Accuracy:  0.36363636363636365
tavg_octdec Accuracy:  0.8
Accuracy:  0.8888888888888888
Accuracy:  0.6
Accuracy:  0.0
Accuracy:  0.125
```

```
#Displaying precision for decision tree models
tavg_janfeb_dt_precision = MulticlassClassificationEvaluator(labelCol="tavg_janfeb", metricName="weightedPrecision").evalu
print("Precision: ",tavg_janfeb_dt_precision)
tavg_marmay_dt_precision = MulticlassClassificationEvaluator(labelCol="tavg_marmay", metricName="weightedPrecision").evalu
print("Precision: ",tavg_marmay_dt_precision)
tavg_junsep_dt_precision = MulticlassClassificationEvaluator(labelCol="tavg_junsep", metricName="weightedPrecision").evalu
print("Precision: ", tavg_junsep_dt_precision)
tavg_octdec_dt_precision = MulticlassClassificationEvaluator(labelCol="tavg_octdec", metricName="weightedPrecision").evalu
print("Precision: ", tavg_octdec_dt_precision)
rf_junsep_dt_precision = MulticlassClassificationEvaluator(labelCol="rf_junsep", metricName="weightedPrecision").evaluate(
print("Precision: ", rf_junsep_dt_precision)
dt_precision_janfeb = MulticlassClassificationEvaluator(labelCol="rf_janfeb", metricName="weightedPrecision").evaluate(df_
dt_precision_marmay = MulticlassClassificationEvaluator(labelCol="rf_marmay", metricName="weightedPrecision").evaluate(df_
print("Precision: ", dt_precision_janfeb)
print("Precision: ", dt_precision_marmay)
rf_octdec_dt_precision = MulticlassClassificationEvaluator(labelCol="rf_octdec", metricName="weightedPrecision").evaluate(
print("Precision: ", rf_octdec_dt_precision)
```

```
Precision:  0.8333333333333334
Precision:  0.6857142857142857
Precision:  0.14545454545454548
Precision:  0.64
Precision:  0.0
Precision:  0.7901234567901234
Precision:  0.8
Precision:  0.03125
```

Random Forest

```
#Importing Random forest regressor and its evaluator
from pyspark.ml.regression import RandomForestRegressor
from pyspark.ml.evaluation import RegressionEvaluator
```

```
#Creating random forest regressor models for each targeted column and fitting on their respective training data
Tavg_janfeb = RandomForestRegressor(labelCol="tavg_janfeb")
Tavg_janfeb_model = Tavg_janfeb.fit(tavg_janfeb_training_df)
Tavg_marmay = RandomForestRegressor(labelCol="tavg_marmay")
Tavg_marmay_model = Tavg_marmay.fit(tavg_marmay_training_df)
tavg_junsep_rf = RandomForestRegressor(numTrees=5, labelCol="tavg_junsep")
tavg_junsep_model = tavg_junsep_rf.fit(tavg_junsep_training_df)

tavg_octdec_rf = RandomForestRegressor(numTrees=5, labelCol="tavg_octdec")
tavg_octdec_model = tavg_octdec_rf.fit(tavg_octdec_training_df)
rf_janfeb_1 = RandomForestRegressor(numTrees=5, labelCol="rf_janfeb")
rf_marmay_1 = RandomForestRegressor(numTrees=5, labelCol="rf_marmay")

model_janfeb = rf_janfeb_1.fit(training_rf_janfeb)
model_marmay = rf_marmay_1.fit(training_rf_marmay)
rf_junsep = RandomForestRegressor(numTrees=5, labelCol="rf_junsep")
rf_junsep_model = rf_junsep.fit(rf_junsep_training_df)

rf_octdec = RandomForestRegressor(numTrees=5, labelCol="rf_octdec")
rf_junsep_model = rf_octdec.fit(rf_octdec_training_df)


#Applying the test data on trained models
tavg_janfeb_df_predictions = Tavg_janfeb_model.transform(tavg_janfeb_test_df)
tavg_janfeb_df_predictions.show()
tavg_marmay_df_predictions = Tavg_janfeb_model.transform(tavg_marmay_test_df)
tavg_marmay_df_predictions.show()
tavg_junsep_predictions = tavg_junsep_model.transform(tavg_junsep_test_df)
tavg_junsep_predictions.show()
tavg_octdec_predictions = tavg_octdec_model.transform(tavg_octdec_test_df)
tavg_octdec_predictions.show()
predictions_rf_janfeb = model_janfeb.transform(test_rf_janfeb)
predictions_rf_janfeb.show()
predictions_rf_marmay = model_marmay.transform(test_rf_marmay)
predictions_rf_marmay.show()
rf_junsep_predictions = rf_junsep_model.transform(rf_junsep_test_df)
rf_junsep_predictions.show()
rf_octdec_predictions = rf_junsep_model.transform(rf_octdec_test_df)
rf_octdec_predictions.show()
```

```
+---------+--------+----------+
|rf_octdec|features|prediction|
+---------+--------+----------+
|        1|[2001.0]|       4.0|
|        1|[2002.0]|       4.0|
|        2|[2012.0]|       3.2|
|        3|[2000.0]|       5.0|
|        4|[1994.0]|       3.1|
|        4|[2014.0]|       2.8|
|        5|[1999.0]|       5.0|
|        6|[2015.0]|       2.8|
+---------+--------+----------+
```

```
#Displaying RMSE value for each RF model
tavg_janfeb_model_evaluator = RegressionEvaluator(
    labelCol="tavg_janfeb", metricName="rmse")
print("Root Mean Squared Error for tavg_janfeb_model = %g" % tavg_janfeb_model_evaluator.evaluate(tavg_janfeb_df_predictio
tavg_marmay_model_evaluator = RegressionEvaluator(
    labelCol="tavg_marmay", metricName="rmse")
print("Root Mean Squared Error for tavg_marmay_model = %g" % tavg_marmay_model_evaluator.evaluate(tavg_marmay_df_predictio
tavg_junsep_model_evaluator = RegressionEvaluator(
    labelCol="tavg_junsep", metricName="rmse")
print("Root Mean Squared Error for tavg_junsep_model = %g" % tavg_junsep_model_evaluator.evaluate(tavg_junsep_predictions)
tavg_octdec_model_evaluator = RegressionEvaluator(
    labelCol="tavg_octdec", metricName="rmse")
print("Root Mean Squared Error for tavg_octdec_model = %g" % tavg_octdec_model_evaluator.evaluate(tavg_octdec_predictions)
rf_janfeb_evaluator = RegressionEvaluator(labelCol="rf_janfeb", metricName="rmse")
print("Root Mean Squared Error for rf_janfeb_evaluator = %g" % rf_janfeb_evaluator.evaluate(predictions_rf_janfeb))
rf_marmay_evaluator = RegressionEvaluator(labelCol="rf_marmay", metricName="rmse")
print("Root Mean Squared Error for rf_marmay_evaluator = %g" % rf_marmay_evaluator.evaluate(predictions_rf_marmay))
rf_junsep_model_evaluator = RegressionEvaluator(
    labelCol="rf_junsep", metricName="rmse")
print("Root Mean Squared Error (RMSE) on rf_junsep_model_evaluator= %g" %rf_junsep_model_evaluator.evaluate(rf_junsep_pred
rf_octdec_model_evaluator = RegressionEvaluator(
    labelCol="rf_octdec", metricName="rmse")
print("Root Mean Squared Error (RMSE) on rf_octdec_model_evaluator= %g" %rf_octdec_model_evaluator.evaluate(rf_octdec_pred
```

```
Root Mean Squared Error for tavg_janfeb_model = 0.70297
Root Mean Squared Error for tavg_marmay_model = 4.38504
Root Mean Squared Error for tavg_junsep_model = 0.811284
Root Mean Squared Error for tavg_octdec_model = 0.340588
Root Mean Squared Error for rf_janfeb_evaluator = 0.333333
Root Mean Squared Error for rf_marmay_evaluator = 0.418197
```