In [32]:

```
# @hidden_cell
# The project token is an authorization token that is used to access project resources like data sources, connections, and used by platform APIs.
from project_lib import Project
project = Project(spark.sparkContext, 'd5baf762-3a3f-4541-ba8f-0bf85b4b3d00', 'p-091a3a7ca5040bbb1a78c7878c43eb8dee03b4bd')
pc = project.project_context
```

```
In [1]: import ibmos2spark
        # @hidden_cell
        credentials = {
            'endpoint': 'https://s3.eu-geo.objectstorage.service.networklayer.com',
            'service_id': 'iam-ServiceId-10604ff5-6186-4e48-bdde-ee9a86142634',
            'iam_service_endpoint': 'https://iam.eu-gb.bluemix.net/oidc/token',
            'api_key': 'TMhjH5iFSO6DhGo1q-wSswMl-8dPadBnzVSCB1tN_aXn'
        }

        configuration_name = 'os_0598830984024571a3ecc109756f7a83_configs'
        cos = ibmos2spark.CloudObjectStorage(sc, credentials, configuration_name, 'bluemix_cos')

        from pyspark.sql import SparkSession
        spark = SparkSession.builder.getOrCreate()
        df = spark.read\
          .format('org.apache.spark.sql.execution.datasources.csv.CSVFileFormat')\
          .option('header', 'true')\
          .load(cos.url('weatherHistory.csv', 'audaz-donotdelete-pr-acosnmalc9mzr6'))
        df.take(5)
```

```
Waiting for a Spark session to start...
Spark Initialization Done! ApplicationId = app-20200612161809-0000
KERNEL_ID = ab91a42d-4433-4542-bbd6-a7432b947cbe
```

```
Out[1]: [Row(Formatted Date='2006-04-01 00:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        ='9.472222222222221', Apparent Temperature (C)='7.388888888888875', Humidity='0.89', Wind Speed (km/h)='14.119
        7', Wind Bearing (degrees)='251.0', Visibility (km)='15.826300000000002', Loud Cover='0.0', Pressure (millibars)
        ='1015.13', Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 01:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        ='9.355555555555558', Apparent Temperature (C)='7.227777777777776', Humidity='0.86', Wind Speed (km/h)='14.2646',
        Wind Bearing (degrees)='259.0', Visibility (km)='15.826300000000002', Loud Cover='0.0', Pressure (millibars)='101
        5.63', Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 02:00:00.000 +0200', Summary='Mostly Cloudy', Precip Type='rain', Temperature (C)
        ='9.377777777777778', Apparent Temperature (C)='9.377777777777778', Humidity='0.89', Wind Speed (km/h)='3.9284000
        000000003', Wind Bearing (degrees)='204.0', Visibility (km)='14.9569', Loud Cover='0.0', Pressure (millibars)='10
        15.94', Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 03:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        ='8.288888888888889', Apparent Temperature (C)='5.944444444444446', Humidity='0.83', Wind Speed (km/h)='14.1036',
        Wind Bearing (degrees)='269.0', Visibility (km)='15.826300000000002', Loud Cover='0.0', Pressure (millibars)='101
        6.41', Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 04:00:00.000 +0200', Summary='Mostly Cloudy', Precip Type='rain', Temperature (C)
        ='8.755555555555553', Apparent Temperature (C)='6.977777777777779', Humidity='0.83', Wind Speed (km/h)='11.0446',
        Wind Bearing (degrees)='259.0', Visibility (km)='15.826300000000002', Loud Cover='0.0', Pressure (millibars)='101
        6.51', Daily Summary='Partly cloudy throughout the day.')]
```

In [4]:

```
#Verificar tipos de dados de acordo com o valores dos dados originais->"Sim" String, 0.1234 -> float, 0,1 - integer
df.dtypes
```

Out[4]:

```
[('Formatted Date', 'string'),
 ('Summary', 'string'),
 ('Precip Type', 'string'),
 ('Temperature (C)', 'string'),
 ('Apparent Temperature (C)', 'string'),
 ('Humidity', 'string'),
 ('Wind Speed (km/h)', 'string'),
 ('Wind Bearing (degrees)', 'string'),
 ('Visibility (km)', 'string'),
 ('Loud Cover', 'string'),
 ('Pressure (millibars)', 'string'),
 ('Daily Summary', 'string')]
```

In [5]:

```
from pyspark.sql.types import IntegerType, FloatType
#Alterar o tipo de dados
df=df.withColumn("Temperature (C)",df["Temperature (C)"].cast(FloatType()))
df=df.withColumn("Apparent Temperature (C)",df["Apparent Temperature (C)"].cast(FloatType()))
df=df.withColumn("Humidity",df["Humidity"].cast(FloatType()))
df=df.withColumn("Wind Speed (km/h)",df["Wind Speed (km/h)"].cast(FloatType()))
df=df.withColumn("Wind Bearing (degrees)",df["Wind Bearing (degrees)"].cast(FloatType()))
df=df.withColumn("Visibility (km)",df["Visibility (km)"].cast(FloatType()))
df=df.withColumn("Loud Cover",df["Loud Cover"].cast(FloatType()))
df=df.withColumn("Pressure (millibars)",df["Pressure (millibars)"].cast(FloatType()))
```

In [6]:
```
df.dtypes
```

Out[6]:
```
[('Formatted Date', 'string'),
 ('Summary', 'string'),
 ('Precip Type', 'string'),
 ('Temperature (C)', 'float'),
 ('Apparent Temperature (C)', 'float'),
 ('Humidity', 'float'),
 ('Wind Speed (km/h)', 'float'),
 ('Wind Bearing (degrees)', 'float'),
 ('Visibility (km)', 'float'),
 ('Loud Cover', 'float'),
 ('Pressure (millibars)', 'float'),
 ('Daily Summary', 'string')]
```

```
In [7]: #Obter os 10 primeiros registos do dataset
        df.head(10)
```

```
Out[7]: [Row(Formatted Date='2006-04-01 00:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =9.472222328186035, Apparent Temperature (C)=7.388888835906982, Humidity=0.8899999856948853, Wind Speed (km/h)=1
        4.11970043182373, Wind Bearing (degrees)=251.0, Visibility (km)=15.826299667358398, Loud Cover=0.0, Pressure (mil
        libars)=1015.1300048828125, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 01:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =9.355555534362793, Apparent Temperature (C)=7.227777957791626, Humidity=0.8600000143051147, Wind Speed (km/h)=14.
        264599800109863, Wind Bearing (degrees)=259.0, Visibility (km)=15.826299667358398, Loud Cover=0.0, Pressure (mill
        ibars)=1015.6300048828125, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 02:00:00.000 +0200', Summary='Mostly Cloudy', Precip Type='rain', Temperature (C)
        =9.377778053283691, Apparent Temperature (C)=9.377778053283691, Humidity=0.8899999856948853, Wind Speed (km/h)=3.
        9284000396728516, Wind Bearing (degrees)=204.0, Visibility (km)=14.956899642944336, Loud Cover=0.0, Pressure (mil
        libars)=1015.9400024414062, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 03:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =8.288888931274414, Apparent Temperature (C)=5.944444656372070, Humidity=0.8299999833106995, Wind Speed (km/h)=14.
        103599548339844, Wind Bearing (degrees)=269.0, Visibility (km)=15.826299667358398, Loud Cover=0.0, Pressure (mill
        ibars)=1016.4099731445312, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 04:00:00.000 +0200', Summary='Mostly Cloudy', Precip Type='rain', Temperature (C)
        =8.755555152893066, Apparent Temperature (C)=6.977777957791626, Humidity=0.8299999833106995, Wind Speed (km/h)=11.
        044599533081055, Wind Bearing (degrees)=259.0, Visibility (km)=15.826299667358398, Loud Cover=0.0, Pressure (mill
        ibars)=1016.510009765625, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 05:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =9.222222328186035, Apparent Temperature (C)=7.111111164093018, Humidity=0.8500000238418579, Wind Speed (km/h)=1
        3.958700180053711, Wind Bearing (degrees)=258.0, Visibility (km)=14.956899642944336, Loud Cover=0.0, Pressure (mi
        llibars)=1016.6599731445312, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 06:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =7.733333110809326, Apparent Temperature (C)=5.52222204208374, Humidity=0.949999988079071, Wind Speed (km/h)=12.3
        64800453186035, Wind Bearing (degrees)=259.0, Visibility (km)=9.982000350952148, Loud Cover=0.0, Pressure (millib
        ars)=1016.719970703125, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 07:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =8.772222518920898, Apparent Temperature (C)=6.527777671813965, Humidity=0.8899999856948853, Wind Speed (km/h)=1
        4.151900291442871, Wind Bearing (degrees)=260.0, Visibility (km)=9.982000350952148, Loud Cover=0.0, Pressure (mil
        libars)=1016.8400268554688, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 08:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =10.822221755981445, Apparent Temperature (C)=10.822221755981445, Humidity=0.8199999928474426, Wind Speed (km/h)=
        11.318300247192383, Wind Bearing (degrees)=259.0, Visibility (km)=9.982000350952148, Loud Cover=0.0, Pressure (mi
        llibars)=1017.3699951171875, Daily Summary='Partly cloudy throughout the day.'),
         Row(Formatted Date='2006-04-01 09:00:00.000 +0200', Summary='Partly Cloudy', Precip Type='rain', Temperature (C)
        =13.772222518920898, Apparent Temperature (C)=13.772222518920898, Humidity=0.7200000286102295, Wind Speed (km/h)=
        12.525799751281738, Wind Bearing (degrees)=279.0, Visibility (km)=9.982000350952148, Loud Cover=0.0, Pressure (mi
        llibars)=1017.219970703125, Daily Summary='Partly cloudy throughout the day.')]
```

In [38]:

```
#Obter a estatística do dataset
df.describe().show()
```

```
+-------+--------------------+--------------------+-----------+------------------+------------------------+------
------------+------------------+----------------------+------------------+----------+--------------------+-------
-------------+
|summary|      Formatted Date|             Summary|Precip Type|   Temperature (C)|Apparent Temperature (C)|
Humidity| Wind Speed (km/h)|Wind Bearing (degrees)|   Visibility (km)|Loud Cover|Pressure (millibars)|       Dail
y Summary|
+-------+--------------------+--------------------+-----------+------------------+------------------------+------
------------+------------------+----------------------+------------------+----------+--------------------+-------
-------------+
|  count|               96453|               96453|      96453|             96453|                   96453|
96453|             96453|                 96453|             96453|     96453|               96453|
96453|
|   mean|                null|                null|       null|11.932678439246953|      10.855028874886619|0.7348
989658888467|10.810640148965067|    187.50923247592092|10.347324990946753|       0.0|  1003.2359558455405|
null|
| stddev|                null|                null|       null| 9.551546321968077|      10.696847391849245|0.1954
727392558967| 6.913571014225619|    107.38342838070588| 4.192123184996354|       0.0|  116.96990569124763|
null|
|    min|2006-01-01 00:00:...|              Breezy|       null|        -21.822222|              -27.716667|
0.0|               0.0|                   0.0|               0.0|       0.0|                 0.0|Breezy and foggy
...|
|    max|2016-12-31 23:00:...|Windy and Partly ...|       snow|         39.905556|               39.344444|
1.0|           63.8526|                 359.0|              16.1|       0.0|             1046.38|Windy in the aft
e...|
+-------+--------------------+--------------------+-----------+------------------+------------------------+------
------------+------------------+----------------------+------------------+----------+--------------------+-------
-------------+
```

In [8]:

```
#Obter estatística de uma coluna do data set-> count sem missing values
#df.describe(["Apparent Temperature (C)"]).show()
df.describe(["Temperature (C)"]).show()
```

```
+-------+-----------------+
|summary|  Temperature (C)|
+-------+-----------------+
|  count|            96453|
|   mean|11.932678439246953|
| stddev| 9.551546321968077|
|    min|       -21.822222|
|    max|        39.905556|
+-------+-----------------+
```

In [9]:

```
df.count()
```

Out[9]:

```
96453
```

#Uma forma de tratar os Missing values é apagá-los mas podemos perder muita informação valiosa #dfmsvretirados=df.na.drop() #dfmsvretirados.count()#dfmsvretirados.filter(dfmsvretirados["Temperature (C)"]==0).show() dfmsvretirados=df dfmsvretirados.filter(dfmsvretirados["Temperature (C)"]==0).show()

In [11]:

```
#Transformar os zero em NaN (missing values)
import numpy as np
from pyspark.sql.functions import when
#retirar nulos

dfmsvretirados= df
dfmsvretirados.fillna(0)
#cols = dfmsvretirados.columns # list of all columns
#for col in cols:
#    dfmsvretirados= dfmsvretirados.withColumn(col, when(dfmsvretirados[col]==0, np.nan).otherwise(dfmsvretirados
[col]))
```

Out[11]:

```
DataFrame[Formatted Date: string, Summary: string, Precip Type: string, Temperature (C): float, Apparent Temperat
ure (C): float, Humidity: float, Wind Speed (km/h): float, Wind Bearing (degrees): float, Visibility (km): float,
Loud Cover: float, Pressure (millibars): float, Daily Summary: string]
```

In [12]:

```
#Contar os missing values das colunas do dataset
from pyspark.sql.functions import isnan, when, count, col
dfmsvretirados.select([count(when(isnan(c), c)).alias(c) for c in dfmsvretirados.columns]).show()
```

```
+--------------+-------+-----------+---------------+------------------------+--------+-----------------+---------
-------------+---------------+----------+--------------------+-------------+
|Formatted Date|Summary|Precip Type|Temperature (C)|Apparent Temperature (C)|Humidity|Wind Speed (km/h)|Wind Bear
ing (degrees)|Visibility (km)|Loud Cover|Pressure (millibars)|Daily Summary|
+--------------+-------+-----------+---------------+------------------------+--------+-----------------+---------
-------------+---------------+----------+--------------------+-------------+
|             0|      0|          0|              0|                       0|       0|                0|
0|             0|         0|                  0|            0|
+--------------+-------+-----------+---------------+------------------------+--------+-----------------+---------
-------------+---------------+----------+--------------------+-------------+
```

#Retirar todos o missing values #dfmvretirados=dfmvretirados.na.drop()

In [13]:

```
#Verificar contagem de missing values
#from pyspark.sql.functions import isnan, when, count, col
#dfmsvretirados.select([count(when(isnan(c), c)).alias(c) for c in dfmsvretirados.columns]).show()
```

Transformar os Missing values pela média

In [44]:

```
#Transformar os zero em Nan
#from pyspark.sql.functions import when
#cols = df.columns # list of all columns
#for col in cols:
#    dfmsparamedia= df.withColumn(col, when(df[col]==0, np.nan).otherwise(df[col]))
```

In [14]:

```
#Calcular média e atribuir aos missing values
from pyspark.sql.functions import avg
dfmsparamedia=dfmsvretirados
#Percorrer todas as variáveis independentes
    #Todos os que comecem por string
for c in dfmsparamedia.columns:
    if not c[0].startswith("string")==False:
        if c.dType==FloatType:
            media=dfmsparamedia.agg(avg(c)).first()[0]
            print(c,media)
        dfmsparamedia=dfmsparamedia.na.fill(media, c[c])
    elif c=="Precip_Type":
        dfmsparamedia=dfmsparamedia.na.fill("rain")
```

In [15]:

```
#Verificar contagem de missing values
from pyspark.sql.functions import isnan, when, count, col
dfmsparamedia.select([count(when(isnan(c), c)).alias(c) for c in dfmsparamedia.columns]).show()
```

```
+--------------+-------+-----------+---------------+------------------------+--------+-----------------+---------
-------------+---------------+----------+--------------------+-------------+
|Formatted Date|Summary|Precip Type|Temperature (C)|Apparent Temperature (C)|Humidity|Wind Speed (km/h)|Wind Bear
ing (degrees)|Visibility (km)|Loud Cover|Pressure (millibars)|Daily Summary|
+--------------+-------+-----------+---------------+------------------------+--------+-----------------+---------
-------------+---------------+----------+--------------------+-------------+
|             0|      0|          0|              0|                       0|       0|                0|
0|              0|         0|                   0|            0|
+--------------+-------+-----------+---------------+------------------------+--------+-----------------+---------
-------------+---------------+----------+--------------------+-------------+
```

In [17]:

```
df2=dfmsparamedia.withColumnRenamed('Temperature (C)', 'Temp_C')
df2=df2.withColumnRenamed('Apparent Temperature (C)', 'A_Temp_C')
df2=df2.withColumnRenamed('Wind Speed (km/h)', 'WindSpeed')
df2=df2.withColumnRenamed('Wind Bearing (degrees)', 'WindBear')
df2=df2.withColumnRenamed('Visibility (km)', 'Visibility')
df2=df2.withColumnRenamed('Loud Cover', 'LC')
df2=df2.withColumnRenamed('Pressure (millibars)', 'Pressure')
df2=df2.withColumnRenamed('Precip Type', 'Precip_Type')
df2=df2.withColumnRenamed('Formatted Date', 'Formatted_Date')
df2=df2.withColumnRenamed('Daily Summary', 'DSummary')
```

In [18]:

```
df_pd = df2.toPandas()
```

In [19]:

```
df_pd
```

Out[19]:

	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.119700	251.0	15.8263	0.0	1015.130005	Partly cloudy throughout the day.
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.264600	259.0	15.8263	0.0	1015.630005	Partly cloudy throughout the day.
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.928400	204.0	14.9569	0.0	1015.940002	Partly cloudy throughout the day.
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288889	5.944445	0.83	14.103600	269.0	15.8263	0.0	1016.409973	Partly cloudy throughout the day.
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755555	6.977778	0.83	11.044600	259.0	15.8263	0.0	1016.510010	Partly cloudy throughout the day.
5	2006-04-01 05:00:00.000 +0200	Partly Cloudy	rain	9.222222	7.111111	0.85	13.958700	258.0	14.9569	0.0	1016.659973	Partly cloudy throughout the day.
6	2006-04-01 06:00:00.000 +0200	Partly Cloudy	rain	7.733333	5.522222	0.95	12.364800	259.0	9.9820	0.0	1016.719971	Partly cloudy throughout the day.
7	2006-04-01 07:00:00.000 +0200	Partly Cloudy	rain	8.772223	6.527778	0.89	14.151900	260.0	9.9820	0.0	1016.840027	Partly cloudy throughout the day.
8	2006-04-01 08:00:00.000 +0200	Partly Cloudy	rain	10.822222	10.822222	0.82	11.318300	259.0	9.9820	0.0	1017.369995	Partly cloudy throughout the day.

	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
9	2006-04-01 09:00:00.000 +0200	Partly Cloudy	rain	13.772223	13.772223	0.72	12.525800	279.0	9.9820	0.0	1017.219971	Partly cloudy throughout the day.
10	2006-04-01 10:00:00.000 +0200	Partly Cloudy	rain	16.016666	16.016666	0.67	17.565100	290.0	11.2056	0.0	1017.419983	Partly cloudy throughout the day.
11	2006-04-01 11:00:00.000 +0200	Partly Cloudy	rain	17.144444	17.144444	0.54	19.786900	316.0	11.4471	0.0	1017.739990	Partly cloudy throughout the day.
12	2006-04-01 12:00:00.000 +0200	Partly Cloudy	rain	17.799999	17.799999	0.55	21.944300	281.0	11.2700	0.0	1017.590027	Partly cloudy throughout the day.
13	2006-04-01 13:00:00.000 +0200	Partly Cloudy	rain	17.333334	17.333334	0.51	20.688499	289.0	11.2700	0.0	1017.479980	Partly cloudy throughout the day.
14	2006-04-01 14:00:00.000 +0200	Partly Cloudy	rain	18.877777	18.877777	0.47	15.375500	262.0	11.4471	0.0	1017.169983	Partly cloudy throughout the day.
15	2006-04-01 15:00:00.000 +0200	Partly Cloudy	rain	18.911112	18.911112	0.46	10.400600	288.0	11.2700	0.0	1016.469971	Partly cloudy throughout the day.
16	2006-04-01 16:00:00.000 +0200	Partly Cloudy	rain	15.388889	15.388889	0.60	14.409500	251.0	11.2700	0.0	1016.150024	Partly cloudy throughout the day.
17	2006-04-01 17:00:00.000 +0200	Mostly Cloudy	rain	15.550000	15.550000	0.63	11.157300	230.0	11.4471	0.0	1016.169983	Partly cloudy throughout the day.

	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
18	2006-04-01 18:00:00.000 +0200	Mostly Cloudy	rain	14.255555	14.255555	0.69	8.516900	163.0	11.2056	0.0	1015.820007	Partly cloudy throughout the day.
19	2006-04-01 19:00:00.000 +0200	Mostly Cloudy	rain	13.144444	13.144444	0.70	7.631400	139.0	11.2056	0.0	1015.830017	Partly cloudy throughout the day.
20	2006-04-01 20:00:00.000 +0200	Mostly Cloudy	rain	11.550000	11.550000	0.77	7.389900	147.0	11.0285	0.0	1015.849976	Partly cloudy throughout the day.
21	2006-04-01 21:00:00.000 +0200	Mostly Cloudy	rain	11.183333	11.183333	0.76	4.926600	160.0	9.9820	0.0	1015.770020	Partly cloudy throughout the day.
22	2006-04-01 22:00:00.000 +0200	Partly Cloudy	rain	10.116667	10.116667	0.79	6.649300	163.0	15.8263	0.0	1015.400024	Partly cloudy throughout the day.
23	2006-04-01 23:00:00.000 +0200	Mostly Cloudy	rain	10.200000	10.200000	0.77	3.928400	152.0	14.9569	0.0	1015.510010	Partly cloudy throughout the day.
24	2006-04-10 00:00:00.000 +0200	Partly Cloudy	rain	10.422222	10.422222	0.62	16.985500	150.0	15.8263	0.0	1014.400024	Mostly cloudy throughout the day.
25	2006-04-10 01:00:00.000 +0200	Partly Cloudy	rain	9.911111	7.566667	0.66	17.210899	149.0	15.8263	0.0	1014.200012	Mostly cloudy throughout the day.
26	2006-04-10 02:00:00.000 +0200	Mostly Cloudy	rain	11.183333	11.183333	0.80	10.819200	163.0	14.9569	0.0	1008.710022	Mostly cloudy throughout the day.

	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
27	2006-04-10 03:00:00.000 +0200	Partly Cloudy	rain	7.155556	5.044445	0.79	11.076800	180.0	15.8263	0.0	1014.469971	Mostly cloudy throughout the day.
28	2006-04-10 04:00:00.000 +0200	Partly Cloudy	rain	6.111111	4.816667	0.82	6.649300	161.0	15.8263	0.0	1014.450012	Mostly cloudy throughout the day.
29	2006-04-10 05:00:00.000 +0200	Partly Cloudy	rain	6.788889	4.272222	0.83	13.008800	135.0	14.9569	0.0	1014.489990	Mostly cloudy throughout the day.
...	...	...	...	...	...	...	...	...	...	...	...	...
96423	2016-09-08 18:00:00.000 +0200	Partly Cloudy	rain	27.799999	27.049999	0.32	3.075100	120.0	16.1000	0.0	1014.039978	Partly cloudy starting overnight.
96424	2016-09-08 19:00:00.000 +0200	Partly Cloudy	rain	24.905556	24.905556	0.51	0.000000	0.0	16.1000	0.0	1014.140015	Partly cloudy starting overnight.
96425	2016-09-08 20:00:00.000 +0200	Partly Cloudy	rain	22.366667	22.366667	0.58	3.332700	135.0	15.5526	0.0	1014.340027	Partly cloudy starting overnight.
96426	2016-09-08 21:00:00.000 +0200	Mostly Cloudy	rain	21.016666	21.016666	0.64	3.220000	340.0	16.1000	0.0	1014.729980	Partly cloudy starting overnight.
96427	2016-09-08 22:00:00.000 +0200	Partly Cloudy	rain	19.927778	19.927778	0.71	3.155600	302.0	16.1000	0.0	1014.630005	Partly cloudy starting overnight.
96428	2016-09-08 23:00:00.000 +0200	Partly Cloudy	rain	18.350000	18.350000	0.77	3.220000	53.0	15.5526	0.0	1014.679993	Partly cloudy starting overnight.

	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
96429	2016-09-09 00:00:00.000 +0200	Partly Cloudy	rain	17.755556	17.755556	0.81	2.962400	12.0	16.1000	0.0	1014.650024	Partly cloudy starting in the morning.
96430	2016-09-09 01:00:00.000 +0200	Clear	rain	16.622223	16.622223	0.87	3.429300	349.0	16.1000	0.0	1014.559998	Partly cloudy starting in the morning.
96431	2016-09-09 02:00:00.000 +0200	Clear	rain	16.144444	16.144444	0.87	3.654700	16.0	15.1501	0.0	1014.690002	Partly cloudy starting in the morning.
96432	2016-09-09 03:00:00.000 +0200	Clear	rain	15.594444	15.594444	0.87	3.284400	41.0	15.4399	0.0	1014.520020	Partly cloudy starting in the morning.
96433	2016-09-09 04:00:00.000 +0200	Clear	rain	15.011111	15.011111	0.93	3.203900	341.0	15.8263	0.0	1014.369995	Partly cloudy starting in the morning.
96434	2016-09-09 05:00:00.000 +0200	Clear	rain	15.016666	15.016666	0.90	2.704800	359.0	14.9569	0.0	1014.549988	Partly cloudy starting in the morning.
96435	2016-09-09 06:00:00.000 +0200	Clear	rain	13.872222	13.872222	0.93	4.749500	0.0	15.8263	0.0	1014.659973	Partly cloudy starting in the morning.

	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
96436	2016-09-09 07:00:00.000 +0200	Clear	rain	16.072222	16.072222	0.88	2.785300	12.0	15.7297	0.0	1015.250000	Partly cloudy starting in the morning.
96437	2016-09-09 08:00:00.000 +0200	Partly Cloudy	rain	19.561111	19.561111	0.75	3.719100	12.0	14.9569	0.0	1015.280029	Partly cloudy starting in the morning.
96438	2016-09-09 09:00:00.000 +0200	Partly Cloudy	rain	22.138889	22.138889	0.65	7.776300	30.0	16.1000	0.0	1015.460022	Partly cloudy starting in the morning.
96439	2016-09-09 10:00:00.000 +0200	Partly Cloudy	rain	22.872223	22.872223	0.59	6.423900	49.0	16.1000	0.0	1015.650024	Partly cloudy starting in the morning.
96440	2016-09-09 11:00:00.000 +0200	Partly Cloudy	rain	27.072222	27.022223	0.42	12.010600	49.0	15.5526	0.0	1015.440002	Partly cloudy starting in the morning.
96441	2016-09-09 12:00:00.000 +0200	Partly Cloudy	rain	28.866667	28.216667	0.37	13.926500	61.0	16.1000	0.0	1015.349976	Partly cloudy starting in the morning.
96442	2016-09-09 13:00:00.000 +0200	Partly Cloudy	rain	30.994444	29.972221	0.33	15.617000	70.0	16.1000	0.0	1014.859985	Partly cloudy starting in the morning.

	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
96443	2016-09-09 14:00:00.000 +0200	Partly Cloudy	rain	30.894444	29.450001	0.28	14.779800	43.0	15.5526	0.0	1014.659973	Partly cloudy starting in the morning.
96444	2016-09-09 15:00:00.000 +0200	Partly Cloudy	rain	31.083334	29.616667	0.28	15.504300	40.0	16.1000	0.0	1014.169983	Partly cloudy starting in the morning.
96445	2016-09-09 16:00:00.000 +0200	Partly Cloudy	rain	31.083334	29.611111	0.28	13.894300	40.0	16.1000	0.0	1013.969971	Partly cloudy starting in the morning.
96446	2016-09-09 17:00:00.000 +0200	Partly Cloudy	rain	30.766666	29.311111	0.28	14.216300	24.0	15.5526	0.0	1013.830017	Partly cloudy starting in the morning.
96447	2016-09-09 18:00:00.000 +0200	Partly Cloudy	rain	28.838888	27.850000	0.32	12.203800	21.0	16.1000	0.0	1014.070007	Partly cloudy starting in the morning.
96448	2016-09-09 19:00:00.000 +0200	Partly Cloudy	rain	26.016666	26.016666	0.43	10.996300	31.0	16.1000	0.0	1014.359985	Partly cloudy starting in the morning.
96449	2016-09-09 20:00:00.000 +0200	Partly Cloudy	rain	24.583334	24.583334	0.48	10.094700	20.0	15.5526	0.0	1015.159973	Partly cloudy starting in the morning.

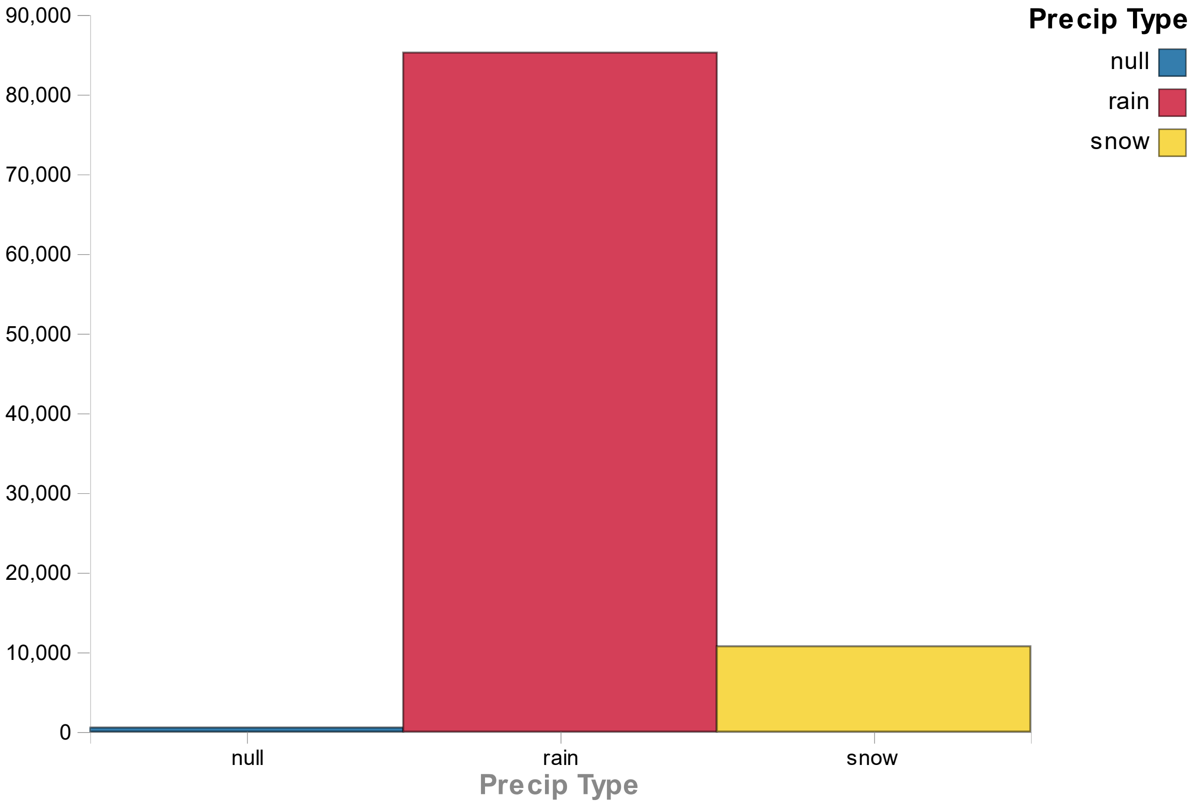
	Formatted_Date	Summary	Precip_Type	Temp_C	A_Temp_C	Humidity	WindSpeed	WindBear	Visibility	LC	Pressure	DSummary
96450	2016-09-09 21:00:00.000 +0200	Partly Cloudy	rain	22.038889	22.038889	0.56	8.983800	30.0	16.1000	0.0	1015.659973	Partly cloudy starting in the morning.
96451	2016-09-09 22:00:00.000 +0200	Partly Cloudy	rain	21.522223	21.522223	0.60	10.529400	20.0	16.1000	0.0	1015.950012	Partly cloudy starting in the morning.
96452	2016-09-09 23:00:00.000 +0200	Partly Cloudy	rain	20.438889	20.438889	0.61	5.876500	39.0	15.5204	0.0	1016.159973	Partly cloudy starting in the morning.

96453 rows × 12 columns

In [20]:

```
import brunel
%brunel data('df_pd') bar x(Precip_Type) y(#count) color(Precip_Type) style('symbol:rect; size:100%;') :: width=6
00, height=400
```

Out[20]:



Criar o Modelo de Machine Learning

In [21]:

```
split_data = df2.randomSplit([0.8, 0.2], 24)
train_data = split_data[0]
test_data = split_data[1]
print('Number of training records: ' + str(train_data.count()))
print('Number of testing records : ' + str(test_data.count()))
```

```
Number of training records: 77090
Number of testing records : 19363
```

In [22]:
```
#Importar as funções da livraria a serem usadas
from pyspark.ml.feature import OneHotEncoder, StringIndexer, IndexToString, VectorAssembler
from pyspark.ml.classification import RandomForestClassifier #Outra função https://spark.apache.org/docs/latest/ml-classification-regression.html
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml import Pipeline, Model
```

In [23]:
```
#Definiçao do Label(dependente)
StringIndexer_label = StringIndexer(inputCol='Precip_Type', outputCol='label').fit(df2) #Output - Label to predict

#Transformar em índices(números) texto
stringIndexer_date = StringIndexer(inputCol='Formatted_Date', outputCol='Data')
stringIndexer_sum = StringIndexer(inputCol='Summary', outputCol='sum')
stringIndexer_dsum = StringIndexer(inputCol='DSummary', outputCol='dsum')
```

In [24]:
```
#Formatted_Date Summary Precip_Type     Temp_C  A_Temp_C        Humidity        WindSpeed       WindBear        Visibility      LC      Pressure        DSummary
vectorAssembler_features = VectorAssembler(inputCols=['Temp_C', 'A_Temp_C', 'Humidity', 'WindSpeed', 'Visibility', 'LC', 'Pressure'], outputCol='features')
```

In [25]:
```
rf = RandomForestClassifier(labelCol='label', featuresCol='features')
```

In [26]:
```
labelConverter = IndexToString(inputCol='prediction', outputCol='predictedLabel', labels=StringIndexer_label.labels)
```

In [27]:
```
#pipeline_rf = Pipeline(stages=[stringIndexer_label, stringIndexer_date, stringIndexer_sum,stringIndexer_dsum, vectorAssembler_features, rf, labelConverter])
pipeline_rf = Pipeline(stages=[StringIndexer_label, stringIndexer_date, stringIndexer_sum,stringIndexer_dsum, vectorAssembler_features, rf])
```

Treino de modelo

In [28]:
```
model_rf = pipeline_rf.fit(train_data)
```

In [29]:

```
predictions = model_rf.transform(test_data)
evaluatorRF = MulticlassClassificationEvaluator(labelCol='label', predictionCol='prediction', metricName='accuracy')
accuracy = evaluatorRF.evaluate(predictions)
print('Accuracy = {:.2f}%'.format(accuracy*100))
print('Test Error = {:.2f}%'.format((1.0 - accuracy)*100))
```

```
Accuracy = 98.08%
Test Error = 1.92%
```

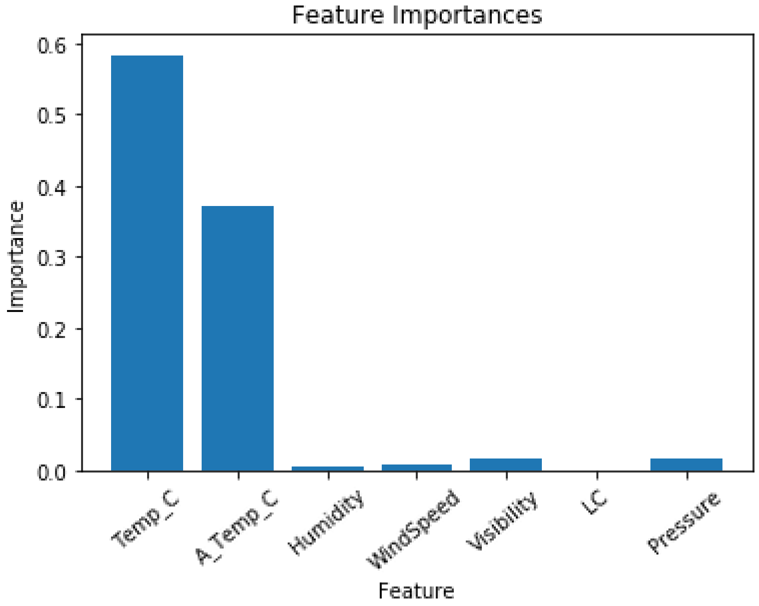
In [31]:

```
import matplotlib.pyplot as plt
importances = model_rf.stages[5].featureImportances
feature_list = ['Temp_C', 'A_Temp_C', 'Humidity', 'WindSpeed', 'Visibility', 'LC', 'Pressure']
x_values = list(range(len(importances)))

plt.bar(x_values, importances, orientation = 'vertical')
plt.xticks(x_values, feature_list, rotation=40)
plt.ylabel('Importance')
plt.xlabel('Feature')
plt.title('Feature Importances')
```

Out[31]:

```
Text(0.5, 1.0, 'Feature Importances')
```



In [32]:

```
rfModel = model_rf.stages[-1]
print(rfModel)  # summary only
```

```
RandomForestClassificationModel (uid=RandomForestClassifier_8eb6ee45a606) with 20 trees
```