```
!sudo apt update
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q https://dlcdn.apache.org/spark/spark-3.3.1/spark-3.3.1-bin-hadoop3.tgz
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu focal-cran40/ InRelease [3,622 B]
Get:2 https://cloud.r-project.org/bin/linux/ubuntu focal-cran40/ Packages [71.1 kB]
Ign:3 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu2004/x86_64  InRelease
Hit:4 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64  InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal InRelease
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:7 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu focal InRelease [18.1 kB]
Hit:8 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu2004/x86_64  Release
Get:9 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:11 http://ppa.launchpad.net/cran/libgit2/ubuntu focal InRelease
Hit:12 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu focal InRelease
Get:13 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Hit:14 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu focal InRelease
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1,291 kB]
Get:16 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu focal/main Sources [2,381 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2,921 kB]
Get:18 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu focal/main amd64 Packages [1,128 kB]
Fetched 8,150 kB in 6s (1,280 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
29 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
!tar xf spark-3.3.1-bin-hadoop3.tgz
!pip install -q findspark
!pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyspark
  Downloading pyspark-3.3.1.tar.gz (281.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 281.4/281.4 MB 4.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting py4j==0.10.9.5
  Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 199.7/199.7 KB 23.0 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.3.1-py2.py3-none-any.whl size=281845512 sha256=a6afb6da4b93660e4f77288dbc469dc149f5
  Stored in directory: /root/.cache/pip/wheels/43/dc/11/ec201cd671da62fa9c5cc77078235e40722170ceba231d7598
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.5 pyspark-3.3.1
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.3.1-bin-hadoop3"
```

```
from google.colab import files
# upload the Vector Assember model av_model.zip
files.upload()
# upload the ML model rf_model.zip
files.upload()
```

```
# extract the models
import os
import zipfile
dirname = os.getcwd()
print('path=',dirname)

rf_filename = "rf_model.zip"
va_filename = "va_model.zip"

rf_path= os.path.join(dirname, rf_filename)
va_path = os.path.join(dirname, va_filename)

zip_rf = zipfile.ZipFile(rf_path, "r")
zip_rf.extractall(dirname)
zip_va = zipfile.ZipFile(va_path, "r")
zip_va.extractall(dirname)


zip_rf.close()
zip_va.close()
```

```
path= /content
```

```
import findspark
findspark.init()
findspark.find()
```

```
'/content/spark-3.3.1-bin-hadoop3'
```

```
from pyspark.sql import DataFrame, SparkSession
from typing import List
import pyspark.sql.types as T
import pyspark.sql.functions as F
from pyspark.sql.functions import isnull, when, count, col
from pyspark import SparkFiles
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

spark = SparkSession.builder \
        .master('local') \
        .appName("NY Parking Violation") \
        .config("spark.sql.adaptive.enabled","true") \
        .config("spark.executor.memory","10g") \
        .config("spark.driver.memory","10g") \
        .getOrCreate()
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
    v3.3.1
Master
    local
AppName
    NY Parking Violation

```
columns_selected = ["Registration State","Plate Type",\
                "Violation Code", "Vehicle Body Type","Vehicle Make","Issuing Agency", "Street Code1", \
               "Street Code2","Street Code3","Violation Location","Violation Precinct", \
               "Issuer Precinct","Issuer Code","Issuer Command",\
               "Violation County","Law Section","Sub Division","Vehicle Color"]
```

```
# cols_index = [df.toPandas().columns.get_loc(col) for col in columns_selected]
# print(cols_index)
```

```
from pyspark.sql import SparkSession
from pyspark.ml import PipelineModel
from pyspark.ml.feature import VectorAssembler, VectorIndexer, OneHotEncoder, StringIndexer
def prepreocess_data(df):
    df.show(5)
    df.count(), len(df.columns)
    df = df.select(columns_selected)

    # clean up the data as many have incorrect values.
    df = df[(df['Registration State'] != "99") \
        & (df['Plate Type'] != "999") \
        & (df['Violation Code'] != 0)]
    # clean up the data
    # Check if the null value still exist
    df.select([count(when(isnull(c), c)).alias(c) for c in df.columns]).show()

    df = df.na.drop()
    df.dropDuplicates()

    # convert to required type
    cols = [F.col(field[0]).cast('double') if (field[1] == 'int') else F.col(field[0]) for field in df.dtypes]
    df = df.select(cols)

    #use model to transform
    pm = PipelineModel.load("/content/va_model")

    df = pm.transform(df)
    return df
```

```
def predict_data(df):
      #use ml model to predict
    pred_model = PipelineModel.load("/content/rf_model")
    data = df.select(F.col("features_scaled").alias("features"))
    # use the PipelineModel object to perform prediciton on  data.
    prediction = pred_model.transform(data)

    # print the results
    # prediction.select('label','prediction','Violation_Location').show(5)
    prediction.select('prediction','Violation_Location').show(10)
    return prediction
```

```
df = spark.read.csv("/content/PV.00002.csv",inferSchema=True, header= True)
df.count(),len(df.columns)
```

```
(100000, 51)
```

```
vec_df = prepreocess_data(df)
# predict_data(vec_df)
```

```
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+--------------+----
|Summons Number|Plate ID|Registration State|Plate Type|Issue Date|Violation Code|Vehicle Body Type|Vehicle Make|Issuing Agency|Stre
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+--------------+----
|    7021662099| GJH8997|                NY|       PAS|08/27/2013|            37|             SUBN|       SUBAR|             T|
|    7021662105| GJU4876|                NY|       PAS|08/27/2013|            37|             SUBN|       ACURA|             T|
|    7022231175|   5F15B|                NY|       OMT|07/29/2013|            21|             TAXI|        FORD|             T|
|    7022235820| GAC6868|                NY|       PAS|08/20/2013|            20|             SUBN|       HONDA|             T|
|    7022235879| 88696JU|                NY|       COM|08/20/2013|            14|              VAN|       CHEVR|             T|
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+--------------+----
only showing top 5 rows

+------------------+----------+--------------+-----------------+------------+--------------+------------+------------+------------+
|Registration State|Plate Type|Violation Code|Vehicle Body Type|Vehicle Make|Issuing Agency|Street Code1|Street Code2|Street Code3
+------------------+----------+--------------+-----------------+------------+--------------+------------+------------+------------+
|                 0|         0|             0|               52|          51|             0|           0|           0|           0
+------------------+----------+--------------+-----------------+------------+--------------+------------+------------+------------+
```

```
vec_df.show(5,False)
```

```
+------------------+----------+--------------+-----------------+------------+--------------+------------+------------+------------+
|Registration State|Plate Type|Violation Code|Vehicle Body Type|Vehicle Make|Issuing Agency|Street Code1|Street Code2|Street Code3
+------------------+----------+--------------+-----------------+------------+--------------+------------+------------+------------+
|NY                |PAS       |37.0          |SUBN             |SUBAR       |T             |40690.0     |24090.0     |24140.0
|NY                |PAS       |37.0          |SUBN             |ACURA       |T             |40690.0     |24090.0     |24140.0
|NY                |OMT       |21.0          |TAXI             |FORD        |T             |23290.0     |17790.0     |18040.0
|NY                |PAS       |20.0          |SUBN             |HONDA       |T             |65590.0     |28990.0     |52090.0
|NY                |COM       |14.0          |VAN              |CHEVR       |T             |59990.0     |16540.0     |16790.0
+------------------+----------+--------------+-----------------+------------+--------------+------------+------------+------------+
only showing top 5 rows
```

```
prediction = predict_data(vec_df)
```

```
+----------+------------------+
|prediction|Violation_Location|
+----------+------------------+
|       4.0|             109.0|
|       4.0|             109.0|
|      35.0|             103.0|
|      35.0|             103.0|
|      27.0|             104.0|
|      35.0|             103.0|
|      27.0|             104.0|
|      25.0|             108.0|
|      35.0|             103.0|
|      27.0|             104.0|
+----------+------------------+
only showing top 10 rows
```

```
prediction.show(10,True)
```

```
+--------------------+--------------------+--------------------+----------+------------------+
|            features|       rawPrediction|         probability|prediction|Violation_Location|
+--------------------+--------------------+--------------------+----------+------------------+
|(1926,[0,59,100,3...|[0.55610398108441...|[0.01390259952711...|       4.0|             109.0|
|(1926,[0,59,100,3...|[0.63189684659933...|[0.01579742116498...|       4.0|             109.0|
|(1926,[0,62,109,3...|[0.43424074714016...|[0.01085601867850...|      35.0|             103.0|
|(1926,[0,59,100,3...|[0.39215854643541...|[0.00980396366088...|      35.0|             103.0|
|(1926,[0,60,99,30...|[0.30858237046250...|[0.00771455926156...|      27.0|             104.0|
|(1926,[0,63,100,3...|[0.29918287285738...|[0.00747957182143...|      35.0|             103.0|
|(1926,[12,59,102,...|[0.18392480567470...|[0.00459812014186...|      27.0|             104.0|
|(1926,[0,60,101,3...|[0.01163596065937...|[2.90899016484277...|      25.0|             108.0|
|(1926,[6,59,100,3...|[0.70451213965000...|[0.01761280349125...|      35.0|             103.0|
|(1926,[12,59,102,...|[0.18392480567470...|[0.00459812014186...|      27.0|             104.0|
+--------------------+--------------------+--------------------+----------+------------------+
only showing top 10 rows
```

```
df.show(10,True)
```

```
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+--------------+----
|Summons Number|Plate ID|Registration State|Plate Type|Issue Date|Violation Code|Vehicle Body Type|Vehicle Make|Issuing Agency|Stre
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+--------------+----
|    7021662099| GJH8997|                NY|       PAS|08/27/2013|            37|             SUBN|       SUBAR|             T|
|    7021662105| GJU4876|                NY|       PAS|08/27/2013|            37|             SUBN|       ACURA|             T|
|    7022231175|   5F15B|                NY|       OMT|07/29/2013|            21|             TAXI|        FORD|             T|
|    7022235820| GAC6868|                NY|       PAS|08/20/2013|            20|             SUBN|       HONDA|             T|
|    7022235879| 88696JU|                NY|       COM|08/20/2013|            14|              VAN|       CHEVR|             T|
|    7022235892| GAH6321|                NY|       OMS|08/20/2013|            38|             SUBN|       TOYOT|             T|
|    7022235909| PAF8691|                GA|       PAS|08/21/2013|            21|             4DSD|       CHEVR|             T|
|    7022235910| 51976MA|                NY|       COM|08/21/2013|            19|             DELV|       INTER|             T|
|    7022235934|  928CA4|                MA|       PAS|08/21/2013|            21|             SUBN|       ACURA|             T|
|    7022235983| PAF8691|                GA|       PAS|08/21/2013|            38|             4DSD|       CHEVR|             T|
+--------------+--------+------------------+----------+----------+--------------+-----------------+------------+--------------+----
only showing top 10 rows
```

Colab paid products - Cancel contracts here

✓ 0s completed at 7:35 PM