LR_IBA_Spark_2_IU5-43M

May 8, 2020

Лабораторная работа по Spark часть 1

Задание:

Установить докер (docker.io)

Скачать датасет (https://www.kaggle.com/kazanova/sentiment140#training.1600000.processed.noemoticon.

Распаковать датасет и поместить его в рабочую папку

Запустить докер образ командой docker run -p 8888:8888 -p 4040:4040 --rm -v "(путь до папки где лежат данные):/home/jovyan/work" jupyter/all-spark-notebook

Прочитать данные и зайти в jupyter notebook

Прочитать документацию по spark (https://spark.apache.org/docs/latest/rdd-programming-guide.html

https://spark.apache.org/docs/latest/sql-programming-guide.html

1) Прочитать датасет и вывести первые 20 записей
2) Посчитать количество слов во всех твитах датасета
3) Посчитать количество уникальных слов во всех твитах

Для Dataframe или RDD

4) Найти пользователей с наибольшим числом постов и вывести первые 20 из них
5) Найти пользователей с наибольшим числов слов
6) Найти пользователей с наибольшим количеством упоминаний(упоминанием следует считать вхождение @username в твит)
7) Посчитать наиболее популярные для твитов дни и вывести их в порядке убывания
8) Посчитать наиболее популярные часы для твитов

[94]:
```
# Создаем сессию
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('abc').getOrCreate()
```

[95]:
```
# Считывание данных датасета
data = spark.read.csv("1")
```

[96]:
```
# Задание 1 - вывести первые 20 записей датасета
data.show()
```

```
+---+----------+--------------------+--------+--------------+-----------------
--+
|_c0|       _c1|                 _c2|     _c3|           _c4|
_c5|
+---+----------+--------------------+--------+--------------+-----------------
--+
|  0|1467810369|Mon Apr 06 22:19:…|NO_QUERY|_TheSpecialOne_|@switchfoot
http:…|
|  0|1467810672|Mon Apr 06 22:19:…|NO_QUERY|  scotthamilton|is upset that he
…|
|  0|1467810917|Mon Apr 06 22:19:…|NO_QUERY|       mattycus|@Kenichan I
dived…|
|  0|1467811184|Mon Apr 06 22:19:…|NO_QUERY|        ElleCTF|my whole body
fee…|
|  0|1467811193|Mon Apr 06 22:19:…|NO_QUERY|         Karoli|@nationwideclass
…|
|  0|1467811372|Mon Apr 06 22:20:…|NO_QUERY|       joy_wolf|@Kwesidei not
the…|
|  0|1467811592|Mon Apr 06 22:20:…|NO_QUERY|        mybirch|         Need a
hug |
|  0|1467811594|Mon Apr 06 22:20:…|NO_QUERY|           coZZ|@LOLTrish hey
lo…|
|  0|1467811795|Mon Apr 06 22:20:…|NO_QUERY|2Hood4Hollywood|@Tatiana_K nope
t…|
|  0|1467812025|Mon Apr 06 22:20:…|NO_QUERY|        mimismo|@twittera que me
…|
|  0|1467812416|Mon Apr 06 22:20:…|NO_QUERY| erinx3leannexo|spring break in
p…|
|  0|1467812579|Mon Apr 06 22:20:…|NO_QUERY|   pardonlauren|I just re-
pierced…|
|  0|1467812723|Mon Apr 06 22:20:…|NO_QUERY|           TLeC|@caregiving I
cou…|
|  0|1467812771|Mon Apr 06 22:20:…|NO_QUERY|robrobbierobert|@octolinz16 It
it…|
|  0|1467812784|Mon Apr 06 22:20:…|NO_QUERY|    bayofwolves|@smarrison i
woul…|
|  0|1467812799|Mon Apr 06 22:20:…|NO_QUERY|     HairByJess|@iamjazzyfizzle
I…|
|  0|1467812964|Mon Apr 06 22:20:…|NO_QUERY| lovesongwriter|Hollis' death
sce…|
|  0|1467813137|Mon Apr 06 22:20:…|NO_QUERY|       armotley|about to file
taxes |
|  0|1467813579|Mon Apr 06 22:20:…|NO_QUERY|     starkissed|@LettyA ahh ive
a…|
```

```
|  0|1467813782|Mon Apr 06 22:20:…|NO_QUERY|      gi_gi_bee|@FakerPattyPattz
…|
+---+----------+------------------+--------+--------------+-----------------
--+
only showing top 20 rows
```

[97]:
```
import pyspark.sql.functions as func
```

[98]:
```
# Задание 2 - подсчет слов во всех твитах датасета
 # Подсчет производим по колонке _c5
data_new = data.withColumn('words_count', func.size(func.split(func.col('_c5'),␣
 ↪' ')))
words_count = data_new.select(func.sum('words_count')).collect()
```

[99]:
```
# Итого слов (с учетом повторяющихся)
words_count
```

[99]:
```
[Row(sum(words_count)=23011409)]
```

[100]:
```
# Задание 3 - подсчитать уникальные слова во всех твитах
data_new = data.withColumn('words_splited', func.split(func.col('_c5'), ' '))
data_new.show()
```

```
+---+----------+------------------+--------+--------------+-----------------
--+--------------------+
|_c0|       _c1|               _c2|     _c3|           _c4|
_c5|       words_splited|
+---+----------+------------------+--------+--------------+-----------------
--+--------------------+
|  0|1467810369|Mon Apr 06 22:19:…|NO_QUERY|_TheSpecialOne_|@switchfoot
http:…|[@switchfoot, htt…|
|  0|1467810672|Mon Apr 06 22:19:…|NO_QUERY|  scotthamilton|is upset that he
…|[is, upset, that,…|
|  0|1467810917|Mon Apr 06 22:19:…|NO_QUERY|       mattycus|@Kenichan I
dived…|[@Kenichan, I, di…|
|  0|1467811184|Mon Apr 06 22:19:…|NO_QUERY|        ElleCTF|my whole body
fee…|[my, whole, body,…|
|  0|1467811193|Mon Apr 06 22:19:…|NO_QUERY|         Karoli|@nationwideclass
…|[@nationwideclass…|
|  0|1467811372|Mon Apr 06 22:20:…|NO_QUERY|       joy_wolf|@Kwesidei not
the…|[@Kwesidei, not, …|
|  0|1467811592|Mon Apr 06 22:20:…|NO_QUERY|        mybirch|         Need a
hug |    [Need, a, hug, ]|
|  0|1467811594|Mon Apr 06 22:20:…|NO_QUERY|           coZZ|@LOLTrish hey
lo…|[@LOLTrish, hey, …|
|  0|1467811795|Mon Apr 06 22:20:…|NO_QUERY|2Hood4Hollywood|@Tatiana_K nope
t…|[@Tatiana_K, nope…|
```

```
|  0|1467812025|Mon Apr 06 22:20:…|NO_QUERY|        mimismo|@twittera que me
…|[@twittera, que, …|
|  0|1467812416|Mon Apr 06 22:20:…|NO_QUERY| erinx3leannexo|spring break in
p…|[spring, break, i…|
|  0|1467812579|Mon Apr 06 22:20:…|NO_QUERY|   pardonlauren|I just re-
pierced…|[I, just, re-pier…|
|  0|1467812723|Mon Apr 06 22:20:…|NO_QUERY|           TLeC|@caregiving I
cou…|[@caregiving, I, …|
|  0|1467812771|Mon Apr 06 22:20:…|NO_QUERY|robrobbierobert|@octolinz16 It
it…|[@octolinz16, It,…|
|  0|1467812784|Mon Apr 06 22:20:…|NO_QUERY|    bayofwolves|@smarrison i
woul…|[@smarrison, i, w…|
|  0|1467812799|Mon Apr 06 22:20:…|NO_QUERY|     HairByJess|@iamjazzyfizzle
I…|[@iamjazzyfizzle,…|
|  0|1467812964|Mon Apr 06 22:20:…|NO_QUERY| lovesongwriter|Hollis' death
sce…|[Hollis', death, …|
|  0|1467813137|Mon Apr 06 22:20:…|NO_QUERY|       armotley|about to file
taxes |[about, to, file,…|
|  0|1467813579|Mon Apr 06 22:20:…|NO_QUERY|     starkissed|@LettyA ahh ive
a…|[@LettyA, ahh, iv…|
|  0|1467813782|Mon Apr 06 22:20:…|NO_QUERY|      gi_gi_bee|@FakerPattyPattz
…|[@FakerPattyPattz…|
+---+----------+--------------------+--------+---------------+-----------------
--+--------------------+
only showing top 20 rows
```

[101]:
```
# Воспользуемся функцией explode
data_words = data_new.select("_c4", func.explode("words_splited"))
data_words.show()
```

```
+---------------+--------------------+
|            _c4|                 col|
+---------------+--------------------+
|_TheSpecialOne_|         @switchfoot|
|_TheSpecialOne_|http://twitpic.co…|
|_TheSpecialOne_|                   -|
|_TheSpecialOne_|               Awww,|
|_TheSpecialOne_|              that's|
|_TheSpecialOne_|                   a|
|_TheSpecialOne_|             bummer.|
|_TheSpecialOne_|                    |
|_TheSpecialOne_|                 You|
|_TheSpecialOne_|             shoulda|
|_TheSpecialOne_|                 got|
|_TheSpecialOne_|               David|
|_TheSpecialOne_|                Carr|
|_TheSpecialOne_|                  of|
```

```
|_TheSpecialOne_|               Third|
|_TheSpecialOne_|                 Day|
|_TheSpecialOne_|                  to|
|_TheSpecialOne_|                  do|
|_TheSpecialOne_|                 it.|
|_TheSpecialOne_|                  ;D|
+---------------+--------------------+
only showing top 20 rows
```

[102]:
```
# Итого уникальных слов
data_words.select('col').distinct().count()
```

[102]: 1350484

[103]:
```
# Задание 4 - Найти пользователей с наибольшим числов постов и вывести первые 20 из них
data_users = data.select("_c4", "_c5")
data_users.show()
```

```
+---------------+--------------------+
|            _c4|                 _c5|
+---------------+--------------------+
|_TheSpecialOne_|@switchfoot http:…|
|  scotthamilton|is upset that he …|
|       mattycus|@Kenichan I dived…|
|        ElleCTF|my whole body fee…|
|         Karoli|@nationwideclass …|
|       joy_wolf|@Kwesidei not the…|
|        mybirch|         Need a hug |
|           coZZ|@LOLTrish hey  lo…|
|2Hood4Hollywood|@Tatiana_K nope t…|
|        mimismo|@twittera que me …|
| erinx3leannexo|spring break in p…|
|   pardonlauren|I just re-pierced…|
|           TLeC|@caregiving I cou…|
|robrobbierobert|@octolinz16 It it…|
|    bayofwolves|@smarrison i woul…|
|     HairByJess|@iamjazzyfizzle I…|
| lovesongwriter|Hollis' death sce…|
|       armotley|about to file taxes |
|     starkissed|@LettyA ahh ive a…|
|      gi_gi_bee|@FakerPattyPattz …|
+---------------+--------------------+
only showing top 20 rows
```

[104]:
```
# Сгруппируем по никнейму и находим количество твитов каждого пользователя
data_count = data.groupBy("_c4").count()
data_count.show()
```

```
+---------------+-----+
|            _c4|count|
+---------------+-----+
|     megan_rice|   15|
|        Daniiej|    3|
|         MeghTW|    1|
|   candicebunny|    1|
|stranger_danger|   14|
|  divingkid2001|    1|
|    Lilli_Allen|    1|
|        caaaami|    1|
|       J_Moneyy|    7|
|        SoEdith|    5|
|     convoy3571|   13|
|       kyrabeth|    1|
|      kateblogs|   75|
|    lovelylivxo|   16|
|       irlbinky|    3|
|        Ste1987|   50|
|       squintoo|    1|
|     PhantomV48|    2|
|        sophizz|    2|
|      tink68113|    1|
+---------------+-----+
only showing top 20 rows
```

[105]:
```
# Проводим сортировку по количеству и выводим топ-20 по твитам
data_count.sort(func.desc("count")).show()
```

```
+---------------+-----+
|            _c4|count|
+---------------+-----+
|       lost_dog|  549|
|        webwoke|  345|
|       tweetpet|  310|
|SallytheShizzle|  281|
|    VioletsCRUK|  279|
|    mcraddictal|  276|
|       tsarnick|  248|
|    what_bugs_u|  246|
|    Karen230683|  238|
|      DarkPiano|  236|
|   SongoftheOss|  227|
```

```
|      Jayme1988|  225|
|         keza34|  219|
| ramdomthoughts|  216|
|      shanajaca|  213|
|         wowlew|  212|
|   TraceyHewins|  211|
|     nuttychris|  211|
|   thisgoeshere|  207|
|     Spidersamm|  205|
+---------------+-----+
only showing top 20 rows
```

[106]:
```
# Задание 5 - Найти пользователей с наибольшим числом слов
 # Группируем по никнейму, подсчитываем слова и сортируем в порядке убывания
data_words.groupBy("_c4").count().sort(func.desc("count")).show()
```

```
+---------------+-----+
|            _c4|count|
+---------------+-----+
|       lost_dog| 6588|
|    what_bugs_u| 5177|
|    VioletsCRUK| 5108|
|SallytheShizzle| 4480|
|      shanajaca| 3698|
| felicityfuller| 3626|
|   SongoftheOss| 3436|
|          StDAY| 3341|
|     nuttychris| 3309|
| ramdomthoughts| 3229|
|       tsarnick| 3170|
|    mcraddictal| 3145|
|     Spidersamm| 3028|
|   JessMcFlyxxx| 2989|
|   thisgoeshere| 2977|
|    linnetwoods| 2928|
|    Dutchrudder| 2897|
|MTVnHollyWEST23| 2821|
| JBnVFCLover786| 2785|
|torilovesbradie| 2748|
+---------------+-----+
only showing top 20 rows
```

[107]:
```
# Задание 6 - Найти пользователей с наибольшим кол-вом упоминаний
 # Производим поиск по колонке col и ищем слова которые начинаются с @ и имеют еще символы
```

```
data_words.groupBy("col").count().select('*').where("col LIKE '@%' and col !=␣
↪'@'").sort(func.desc("count")).show()
```

```
+---------------+-----+
|            col|count|
+---------------+-----+
|     @mileycyrus| 4310|
|       @tommcfly| 3837|
|       @ddlovato| 3349|
|  @Jonasbrothers| 1263|
|    @DavidArchie| 1222|
|   @jordanknight| 1105|
| @DonnieWahlberg| 1085|
|@JonathanRKnight| 1053|
|   @mitchelmusso| 1038|
|  @taylorswift13|  973|
|  @jonasbrothers|  954|
|    @selenagomez|  782|
|    @dougiemcfly|  781|
|         @aplusk|  606|
| @peterfacinelli|  602|
|   @joeymcintyre|  562|
|    @gfalcone601|  539|
|     @Dannymcfly|  538|
|   @shaundiviney|  503|
|         @YoungQ|  501|
+---------------+-----+
only showing top 20 rows
```

[108]:
```
# Задание 7 - Посчитать наиболее популярные для твитов дни и вывести их в порядке убывания
data.groupBy("_c2").count().sort(func.desc("count")).show()
```

```
+--------------------+-----+
|                 _c2|count|
+--------------------+-----+
|Mon Jun 15 12:53:…|   20|
|Mon Jun 15 13:39:…|   17|
|Fri May 29 13:40:…|   17|
|Fri May 22 05:10:…|   17|
|Fri Jun 05 11:05:…|   16|
|Fri Jun 05 14:13:…|   16|
|Mon Jun 01 15:19:…|   15|
|Mon Jun 01 12:25:…|   15|
|Fri Jun 05 14:17:…|   15|
|Fri May 22 08:34:…|   15|
|Sat Jun 06 11:59:…|   15|
|Fri May 29 10:09:…|   15|
```

```
|Sat May 30 07:53:…|   15|
|Fri Jun 05 09:45:…|   15|
|Mon Jun 15 10:39:…|   15|
|Fri Jun 05 15:47:…|   14|
|Mon Jun 15 13:40:…|   14|
|Fri May 22 07:45:…|   14|
|Sun May 17 11:20:…|   14|
|Fri May 29 21:11:…|   14|
+--------------------+-----+
only showing top 20 rows
```

[109]:
```
# Берем третий элемент массива (колонки) с датой твитов
data_date = data.withColumn('day', func.split(func.col('_c2'), ' ')[2])
```

[110]:
```
data_date.sort(func.desc('_c4')).show()
```

```
+---+----------+--------------------+--------+---------------+------------------
--+---+
|_c0|       _c1|                 _c2|     _c3|            _c4|
_c5|day|
+---+----------+--------------------+--------+---------------+------------------
--+---+
|  0|2298661604|Tue Jun 23 11:57:…|NO_QUERY|     zzzzeus111|Wishing I were
th…| 23|
|  0|2011498170|Tue Jun 02 19:06:…|NO_QUERY|zzzyourdreamzzz|OMG… Im the
…| 02|
|  4|1980201081|Sun May 31 06:01:…|NO_QUERY|     zzzunzinnn|wat a mild
SURPRI…| 31|
|  4|2064493454|Sun Jun 07 06:25:…|NO_QUERY|     zzzunzinnn|tanoshii
tanoshii…| 07|
|  4|2053334447|Sat Jun 06 04:10:…|NO_QUERY|     zzzunzinnn|I LOVEEE.. YOU,
M…| 06|
|  4|1551520572|Sat Apr 18 09:16:…|NO_QUERY|          zzzum|Oh lord yes! mom
…| 18|
|  4|1825885604|Sun May 17 07:07:…|NO_QUERY|         zzztar|Watching
Lipstick…| 17|
|  4|1826960235|Sun May 17 09:41:…|NO_QUERY|         zzztar|@Heather_Poole I
…| 17|
|  0|2066133668|Sun Jun 07 09:57:…|NO_QUERY|      zzzoeface|my sister comes
h…| 07|
|  0|2236027288|Fri Jun 19 02:40:…|NO_QUERY|     zzzap_BOOM|Is sleep vital?
N…| 19|
|  4|1751920428|Sat May 09 20:36:…|NO_QUERY|        zzzaney|just sending a
tw…| 09|
|  4|1968081803|Fri May 29 20:43:…|NO_QUERY|       zzzandra|@StarChild54 aw,
…| 29|
|  4|2009037610|Tue Jun 02 15:03:…|NO_QUERY|       zzzandra|that redbox
```

```
vendi…| 02|
|  0|1795698676|Thu May 14 08:13:…|NO_QUERY|       zzzandra|officially set
TW…| 14|
|  4|1990023165|Mon Jun 01 03:37:…|NO_QUERY|       zzzainus|check out
rambo's…| 01|
|  4|1990409870|Mon Jun 01 04:47:…|NO_QUERY|       zzzainus|bringing you
jame…| 01|
|  0|2249466470|Fri Jun 19 22:23:…|NO_QUERY|      zzzValzzz|@DalkullanJewel
t…| 19|
|  0|1958069290|Fri May 29 02:33:…|NO_QUERY|      zzzValzzz|uhh I wish
someon…| 29|
|  0|2322412473|Wed Jun 24 22:34:…|NO_QUERY|      zzzValzzz|@ArtByZoe I am
tr…| 24|
|  0|2260003820|Sat Jun 20 18:30:…|NO_QUERY|      zzzValzzz|@HelmStudios hey
…| 20|
+---+----------+--------------------+--------+---------------+-----------------
--+---+
only showing top 20 rows
```

[111]:
```
# Группируем по дню и подсчитываем кол-во твитов в дни в порядке убывания
data_date.groupBy('day').count().sort(func.desc("count")).show()
```

```
+---+------+
|day| count|
+---+------+
| 01|118006|
| 06|115036|
| 07|113661|
| 15|109781|
| 30|103673|
| 18|101227|
| 31| 94588|
| 02| 91626|
| 17| 87250|
| 16| 77126|
| 03| 76921|
| 29| 73827|
| 19| 70605|
| 20| 63824|
| 05| 58757|
| 22| 51083|
| 21| 27079|
| 10| 26029|
| 14| 25732|
| 04| 23323|
+---+------+
only showing top 20 rows
```

[112]:
```
# Задание 8 - посчитать наиболее популярные часы для твитов
data_time = data.withColumn('hours', func.split(func.split(func.col('_c2'), ' ␣
 ↪')[3], ':')[0])\
.groupBy("hours").count().sort(func.desc("count")).show()
```

```
+-----+-----+
|hours|count|
+-----+-----+
|   23|84750|
|   07|83654|
|   00|80865|
|   06|80852|
|   05|78623|
|   22|78328|
|   04|76995|
|   08|76287|
|   01|75268|
|   03|74253|
|   02|73991|
|   21|68964|
|   09|67278|
|   11|61009|
|   10|60689|
|   19|57722|
|   20|57059|
|   16|55720|
|   18|53485|
|   17|51843|
+-----+-----+
only showing top 20 rows
```

SparkMLlib Вторая часть работы со Spark

Задание:

Изучить документацию: https://spark.apache.org/docs/latest/ml-guide.html https://spark.apache.org/docs/latest/ml-statistics.html https://spark.apache.org/docs/latest/ml-datasource https://spark.apache.org/docs/latest/ml-features.html https://spark.apache.org/docs/latest/ml-classification-regression.html

Загрузить https:/data.cityofchicago.org/Health-Human-Services/Food-Inspections/4ijn-s7e5 в csv формате

Выполнить примеры из https://spark.apache.org/docs/latest/ml-statistics.html https://spark.apache.org/docs/latest/ml-pipeline.html

Построить модель предсказаний:

Изучить пример https://docs.microsoft.com/ru-ru/azure/hdinsight/spark/apache-spark-machine-learning-mllib-ipython

Повторить данный прмер с той лишь разницей что считываются данные с помощью функции spark.read.csv (прочитать какие нужны параметры), чтобы прочитало корректно учесть что %% не работает, (local) и можно заменить %local countResultsdf = spark.sql("SELECT COUNT(results) AS cnt, results as results FROM CountResults GROUP BY results").toPandas()

Второго датасета нет, сделать семплирование датасета на тестовый и тот с помощью к-ого обучается модель регресии (0.8, 0.2) пропорции с помощью функции sample

[113]:
```
from pyspark.ml.linalg import Vectors
from pyspark.ml.stat import Correlation
```

[114]:
```
data = [(Vectors.sparse(4, [(0, 1.0), (3, -2.0)]),),
        (Vectors.dense([4.0, 5.0, 0.0, 3.0]),),
        (Vectors.dense([6.0, 7.0, 0.0, 8.0]),),
        (Vectors.sparse(4, [(0, 9.0), (3, 1.0)]),)]
df = spark.createDataFrame(data, ["features"])

r1 = Correlation.corr(df, "features").head()
print("Pearson correlation matrix:\n" + str(r1[0]))

r2 = Correlation.corr(df, "features", "spearman").head()
print("Spearman correlation matrix:\n" + str(r2[0]))
```

```
Pearson correlation matrix:
DenseMatrix([[1.        , 0.05564149,        nan, 0.40047142],
             [0.05564149, 1.        ,        nan, 0.91359586],
             [       nan,        nan, 1.        ,        nan],
             [0.40047142, 0.91359586,        nan, 1.        ]])
Spearman correlation matrix:
DenseMatrix([[1.        , 0.10540926,        nan, 0.4       ],
             [0.10540926, 1.        ,        nan, 0.9486833 ],
             [       nan,        nan, 1.        ,        nan],
             [0.4       , 0.9486833 ,        nan, 1.        ]])
```

[115]:
```
from pyspark.ml.linalg import Vectors
from pyspark.ml.stat import ChiSquareTest
```

[116]:
```
data = [(0.0, Vectors.dense(0.5, 10.0)),
        (0.0, Vectors.dense(1.5, 20.0)),
        (1.0, Vectors.dense(1.5, 30.0)),
        (0.0, Vectors.dense(3.5, 30.0)),
        (0.0, Vectors.dense(3.5, 40.0)),
        (1.0, Vectors.dense(3.5, 40.0))]
df = spark.createDataFrame(data, ["label", "features"])

r = ChiSquareTest.test(df, "features", "label").head()
print("pValues: " + str(r.pValues))
```

```
print("degreesOfFreedom: " + str(r.degreesOfFreedom))
print("statistics: " + str(r.statistics))
```

```
pValues: [0.6872892787909721,0.6822703303362126]
degreesOfFreedom: [2, 3]
statistics: [0.75,1.5]
```

[117]:
```
from pyspark.ml.stat import Summarizer
from pyspark.sql import Row
from pyspark.ml.linalg import Vectors
sc = spark.sparkContext
```

[118]:
```
df = sc.parallelize([Row(weight=1.0, features=Vectors.dense(1.0, 1.0, 1.0)),
                     Row(weight=0.0, features=Vectors.dense(1.0, 2.0, 3.0))]).
 ↪toDF()

# create summarizer for multiple metrics "mean" and "count"
summarizer = Summarizer.metrics("mean", "count")

# compute statistics for multiple metrics with weight
df.select(summarizer.summary(df.features, df.weight)).show(truncate=False)

# compute statistics for multiple metrics without weight
df.select(summarizer.summary(df.features)).show(truncate=False)

# compute statistics for single metric "mean" with weight
df.select(Summarizer.mean(df.features, df.weight)).show(truncate=False)

# compute statistics for single metric "mean" without weight
df.select(Summarizer.mean(df.features)).show(truncate=False)
```

```
+-----------------------------------+
|aggregate_metrics(features, weight)|
+-----------------------------------+
|[[1.0,1.0,1.0], 1]                 |
+-----------------------------------+

+--------------------------------+
|aggregate_metrics(features, 1.0)|
+--------------------------------+
|[[1.0,1.5,2.0], 2]              |
+--------------------------------+

+--------------+
|mean(features)|
+--------------+
|[1.0,1.0,1.0] |
```

```
+--------------+

+--------------+
|mean(features)|
+--------------+
|[1.0,1.5,2.0] |
+--------------+
```

[119]:
```
from pyspark.ml.linalg import Vectors
from pyspark.ml.classification import LogisticRegression
```

[120]:
```
# Prepare training data from a list of (label, features) tuples.
training = spark.createDataFrame([
    (1.0, Vectors.dense([0.0, 1.1, 0.1])),
    (0.0, Vectors.dense([2.0, 1.0, -1.0])),
    (0.0, Vectors.dense([2.0, 1.3, 1.0])),
    (1.0, Vectors.dense([0.0, 1.2, -0.5]))], ["label", "features"])
```

[121]:
```
# Create a LogisticRegression instance. This instance is an Estimator.
lr = LogisticRegression(maxIter=10, regParam=0.01)
```

[122]:
```
# Print out the parameters, documentation, and any default values.
print("LogisticRegression parameters:\n" + lr.explainParams() + "\n")
```

```
LogisticRegression parameters:
aggregationDepth: suggested depth for treeAggregate (>= 2). (default: 2)
elasticNetParam: the ElasticNet mixing parameter, in range [0, 1]. For alpha =
0, the penalty is an L2 penalty. For alpha = 1, it is an L1 penalty. (default:
0.0)
family: The name of family which is a description of the label distribution to
be used in the model. Supported options: auto, binomial, multinomial (default:
auto)
featuresCol: features column name. (default: features)
fitIntercept: whether to fit an intercept term. (default: True)
labelCol: label column name. (default: label)
lowerBoundsOnCoefficients: The lower bounds on coefficients if fitting under
bound constrained optimization. The bound matrix must be compatible with the
shape (1, number of features) for binomial regression, or (number of classes,
number of features) for multinomial regression. (undefined)
lowerBoundsOnIntercepts: The lower bounds on intercepts if fitting under bound
constrained optimization. The bounds vector size must beequal with 1 for
binomial regression, or the number oflasses for multinomial regression.
(undefined)
maxIter: max number of iterations (>= 0). (default: 100, current: 10)
predictionCol: prediction column name. (default: prediction)
probabilityCol: Column name for predicted class conditional probabilities. Note:
Not all models output well-calibrated probability estimates! These probabilities
```

```
should be treated as confidences, not precise probabilities. (default:
probability)
rawPredictionCol: raw prediction (a.k.a. confidence) column name. (default:
rawPrediction)
regParam: regularization parameter (>= 0). (default: 0.0, current: 0.01)
standardization: whether to standardize the training features before fitting the
model. (default: True)
threshold: Threshold in binary classification prediction, in range [0, 1]. If
threshold and thresholds are both set, they must match.e.g. if threshold is p,
then thresholds must be equal to [1-p, p]. (default: 0.5)
thresholds: Thresholds in multi-class classification to adjust the probability
of predicting each class. Array must have length equal to the number of classes,
with values > 0, excepting that at most one value may be 0. The class with
largest value p/t is predicted, where p is the original probability of that
class and t is the class's threshold. (undefined)
tol: the convergence tolerance for iterative algorithms (>= 0). (default: 1e-06)
upperBoundsOnCoefficients: The upper bounds on coefficients if fitting under
bound constrained optimization. The bound matrix must be compatible with the
shape (1, number of features) for binomial regression, or (number of classes,
number of features) for multinomial regression. (undefined)
upperBoundsOnIntercepts: The upper bounds on intercepts if fitting under bound
constrained optimization. The bound vector size must be equal with 1 for
binomial regression, or the number of classes for multinomial regression.
(undefined)
weightCol: weight column name. If this is not set or empty, we treat all
instance weights as 1.0. (undefined)
```

[123]:
```
# Learn a LogisticRegression model. This uses the parameters stored in lr.
model1 = lr.fit(training)
```

[124]:
```
# Since model1 is a Model (i.e., a transformer produced by an Estimator),
# we can view the parameters it used during fit().
# This prints the parameter (name: value) pairs, where names are unique IDs for␣
 ↪this
# LogisticRegression instance.
print("Model 1 was fit using parameters: ")
print(model1.extractParamMap())
```

```
Model 1 was fit using parameters:
{Param(parent='LogisticRegression_740d4665432d', name='aggregationDepth',
doc='suggested depth for treeAggregate (>= 2)'): 2,
Param(parent='LogisticRegression_740d4665432d', name='elasticNetParam', doc='the
ElasticNet mixing parameter, in range [0, 1]. For alpha = 0, the penalty is an
L2 penalty. For alpha = 1, it is an L1 penalty'): 0.0,
Param(parent='LogisticRegression_740d4665432d', name='family', doc='The name of
family which is a description of the label distribution to be used in the model.
Supported options: auto, binomial, multinomial.'): 'auto',
```

```
Param(parent='LogisticRegression_740d4665432d', name='featuresCol',
doc='features column name'): 'features',
Param(parent='LogisticRegression_740d4665432d', name='fitIntercept',
doc='whether to fit an intercept term'): True,
Param(parent='LogisticRegression_740d4665432d', name='labelCol', doc='label
column name'): 'label', Param(parent='LogisticRegression_740d4665432d',
name='maxIter', doc='maximum number of iterations (>= 0)'): 10,
Param(parent='LogisticRegression_740d4665432d', name='predictionCol',
doc='prediction column name'): 'prediction',
Param(parent='LogisticRegression_740d4665432d', name='probabilityCol',
doc='Column name for predicted class conditional probabilities. Note: Not all
models output well-calibrated probability estimates! These probabilities should
be treated as confidences, not precise probabilities'): 'probability',
Param(parent='LogisticRegression_740d4665432d', name='rawPredictionCol',
doc='raw prediction (a.k.a. confidence) column name'): 'rawPrediction',
Param(parent='LogisticRegression_740d4665432d', name='regParam',
doc='regularization parameter (>= 0)'): 0.01,
Param(parent='LogisticRegression_740d4665432d', name='standardization',
doc='whether to standardize the training features before fitting the model'):
True, Param(parent='LogisticRegression_740d4665432d', name='threshold',
doc='threshold in binary classification prediction, in range [0, 1]'): 0.5,
Param(parent='LogisticRegression_740d4665432d', name='tol', doc='the convergence
tolerance for iterative algorithms (>= 0)'): 1e-06}
```

[125]:
```
# We may alternatively specify parameters using a Python dictionary as a␣
↪paramMap
paramMap = {lr.maxIter: 20}
paramMap[lr.maxIter] = 30  # Specify 1 Param, overwriting the original maxIter.
paramMap.update({lr.regParam: 0.1, lr.threshold: 0.55})  # Specify multiple␣
↪Params.
```

[126]:
```
# You can combine paramMaps, which are python dictionaries.
paramMap2 = {lr.probabilityCol: "myProbability"}  # Change output column name
paramMapCombined = paramMap.copy()
paramMapCombined.update(paramMap2)
```

[127]:
```
# Now learn a new model using the paramMapCombined parameters.
# paramMapCombined overrides all parameters set earlier via lr.set* methods.
model2 = lr.fit(training, paramMapCombined)
print("Model 2 was fit using parameters: ")
print(model2.extractParamMap())
```

```
Model 2 was fit using parameters:
{Param(parent='LogisticRegression_740d4665432d', name='aggregationDepth',
doc='suggested depth for treeAggregate (>= 2)'): 2,
Param(parent='LogisticRegression_740d4665432d', name='elasticNetParam', doc='the
ElasticNet mixing parameter, in range [0, 1]. For alpha = 0, the penalty is an
L2 penalty. For alpha = 1, it is an L1 penalty'): 0.0,
```

```
Param(parent='LogisticRegression_740d4665432d', name='family', doc='The name of
family which is a description of the label distribution to be used in the model.
Supported options: auto, binomial, multinomial.'): 'auto',
Param(parent='LogisticRegression_740d4665432d', name='featuresCol',
doc='features column name'): 'features',
Param(parent='LogisticRegression_740d4665432d', name='fitIntercept',
doc='whether to fit an intercept term'): True,
Param(parent='LogisticRegression_740d4665432d', name='labelCol', doc='label
column name'): 'label', Param(parent='LogisticRegression_740d4665432d',
name='maxIter', doc='maximum number of iterations (>= 0)'): 30,
Param(parent='LogisticRegression_740d4665432d', name='predictionCol',
doc='prediction column name'): 'prediction',
Param(parent='LogisticRegression_740d4665432d', name='probabilityCol',
doc='Column name for predicted class conditional probabilities. Note: Not all
models output well-calibrated probability estimates! These probabilities should
be treated as confidences, not precise probabilities'): 'myProbability',
Param(parent='LogisticRegression_740d4665432d', name='rawPredictionCol',
doc='raw prediction (a.k.a. confidence) column name'): 'rawPrediction',
Param(parent='LogisticRegression_740d4665432d', name='regParam',
doc='regularization parameter (>= 0)'): 0.1,
Param(parent='LogisticRegression_740d4665432d', name='standardization',
doc='whether to standardize the training features before fitting the model'):
True, Param(parent='LogisticRegression_740d4665432d', name='threshold',
doc='threshold in binary classification prediction, in range [0, 1]'): 0.55,
Param(parent='LogisticRegression_740d4665432d', name='tol', doc='the convergence
tolerance for iterative algorithms (>= 0)'): 1e-06}
```

[128]:
```
# Prepare test data
test = spark.createDataFrame([
    (1.0, Vectors.dense([-1.0, 1.5, 1.3])),
    (0.0, Vectors.dense([3.0, 2.0, -0.1])),
    (1.0, Vectors.dense([0.0, 2.2, -1.5]))], ["label", "features"])
```

[129]:
```
# Make predictions on test data using the Transformer.transform() method.
# LogisticRegression.transform will only use the 'features' column.
# Note that model2.transform() outputs a "myProbability" column instead of the␣
 ↪usual
# 'probability' column since we renamed the lr.probabilityCol parameter␣
 ↪previously.
prediction = model2.transform(test)
result = prediction.select("features", "label", "myProbability", "prediction") \
    .collect()
```

[130]:
```
for row in result:
    print("features=%s, label=%s -> prob=%s, prediction=%s"
          % (row.features, row.label, row.myProbability, row.prediction))
```

```
features=[-1.0,1.5,1.3], label=1.0 ->
```

```
prob=[0.057073041710340625,0.9429269582896593], prediction=1.0
features=[3.0,2.0,-0.1], label=0.0 ->
prob=[0.9238522311704118,0.07614776882958811], prediction=0.0
features=[0.0,2.2,-1.5], label=1.0 ->
prob=[0.10972776114779748,0.8902722388522026], prediction=1.0
```

[131]:
```
from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import HashingTF, Tokenizer
```

[132]:
```
# Prepare training documents from a list of (id, text, label) tuples.
training = spark.createDataFrame([
    (0, "a b c d e spark", 1.0),
    (1, "b d", 0.0),
    (2, "spark f g h", 1.0),
    (3, "hadoop mapreduce", 0.0)
], ["id", "text", "label"])
```

[133]:
```
# Configure an ML pipeline, which consists of three stages: tokenizer,␣
 ↪hashingTF, and lr.
tokenizer = Tokenizer(inputCol="text", outputCol="words")
hashingTF = HashingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
lr = LogisticRegression(maxIter=10, regParam=0.001)
pipeline = Pipeline(stages=[tokenizer, hashingTF, lr])
```

[134]:
```
# Fit the pipeline to training documents.
model = pipeline.fit(training)
```

[135]:
```
# Prepare test documents, which are unlabeled (id, text) tuples.
test = spark.createDataFrame([
    (4, "spark i j k"),
    (5, "l m n"),
    (6, "spark hadoop spark"),
    (7, "apache hadoop")
], ["id", "text"])
```

[136]:
```
# Make predictions on test documents and print columns of interest.
prediction = model.transform(test)
selected = prediction.select("id", "text", "probability", "prediction")
for row in selected.collect():
    rid, text, prob, prediction = row
    print("(%d, %s) --> prob=%s, prediction=%f" % (rid, text, str(prob),␣
 ↪prediction))
```

```
(4, spark i j k) --> prob=[0.1596407738787475,0.8403592261212525],
prediction=1.000000
(5, l m n) --> prob=[0.8378325685476744,0.16216743145232562],
prediction=0.000000
```

```
(6, spark hadoop spark) --> prob=[0.06926633132976037,0.9307336686702395],
prediction=1.000000
(7, apache hadoop) --> prob=[0.9821575333444218,0.01784246665557808],
prediction=0.000000
```

[137]:
```
#
from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import HashingTF, Tokenizer
from pyspark.sql import Row
from pyspark.sql.functions import UserDefinedFunction
from pyspark.sql.types import *
```

[138]:
```
inspections = spark.read.csv("2", header=True, inferSchema=True)
```

[139]:
```
inspections.take(1)
```

[139]:
```
[Row(Inspection ID=2370170, DBA Name='DAMEN FOOD MARKET', AKA Name='DAMEN FOOD
MARKET', License #=2671810, Facility Type='Grocery Store', Risk='Risk 2
(Medium)', Address='2001 S DAMEN AVE ', City='CHICAGO', State='IL', Zip=60608,
Inspection Date='05/06/2020', Inspection Type='Canvass', Results='Fail',
Violations='36. THERMOMETERS PROVIDED & ACCURATE - Comments: 4-302.12 OBSERVED
NO METAL STEM THERMOMETER ON SITE TO PROPERLY TAKE TEMPERATURES OF COLD AND HOT
FOODS. INSTRUCTED TO PROVIDE. PRIORITY FOUNDATION 7-38-005. | 38. INSECTS,
RODENTS, & ANIMALS NOT PRESENT - Comments: 6-202.15 OBSERVED FRONT ENTRANCE DOOR
NOT RODENT PROOFED WITH A 1/4 INCH GAP. INSTRUCTED TO MAKE SAID DOOR TIGHT
FITTING. | 55. PHYSICAL FACILITIES INSTALLED, MAINTAINED & CLEAN - Comments:
6-501.114 OBSERVED ITEMS ON THE FLOOR IN THE REAR. INSTRUCTED TO ORGANIZE AND
ELEVATE EVERYTHING AT LEAST 6 INCHES.', Latitude=41.85491131910428,
Longitude=-87.67577256587357, Location='(-87.67577256587357,
41.85491131910428)')]
```

[140]:
```
schema = StructType([
StructField("id", IntegerType(), False),
StructField("name", StringType(), False),
StructField("results", StringType(), False),
StructField("violations", StringType(), True)])

df = spark.createDataFrame(inspections.rdd.map(lambda l: (int(l[0]), l[1],␣
 ↪l[12], l[13])) , schema)
df.registerTempTable('CountResults')
```

[142]:
```
df.show()
```

```
+-------+--------------------+-----------------+--------------------+
|     id|                name|          results|          violations|
+-------+--------------------+-----------------+--------------------+
|2370170|   DAMEN FOOD MARKET|             Fail|36. THERMOMETERS …|
```

```
|2370099|JIMMYS BEST MIXED…|Pass w/ Conditions|21. PROPER HOT HO…|
|2370053|      79 FOOD BASKET|              Fail|3. MANAGEMENT, FO…|
|2370044|CHACKO'S MINI FD…|   Out of Business|              null|
|2370040|    VILLA ROSA PIZZA|   Out of Business|              null|
|2370035|HAROLD'S CHICKEN …|          No Entry|              null|
|2370045|     ZAYNA MED GRILL|              Pass|              null|
|2370031|ENERGY AND HEALTH…|   Out of Business|              null|
|2370027|     HOT DOG EXPRESS|   Out of Business|              null|
|2370016|LEAGUE CHILD DEVE…|          No Entry|              null|
|2370015|               CITGO|   Out of Business|              null|
|2370022|                IHOP|   Out of Business|              null|
|2370003|FRIENDLY FOOD & L…|   Out of Business|              null|
|2369994|   WALGREENS # 06297|Pass w/ Conditions|5. PROCEDURES FOR…|
|2369996|GODDESS AND THE B…|              Pass|              null|
|2369973|     WALGREENS #1496|              Pass|              null|
|2369959|SUPER JALAPENO GR…|              Pass|              null|
|2369955|HUMBOLDT HAUS DEL…|              Pass|53. TOILET FACILI…|
|2369930|          MOBILE GAS|              Pass|              null|
|2369937|BIG SHOULDERS COF…|              Pass|              null|
+-------+--------------------+------------------+--------------------+
only showing top 20 rows
```

[143]:
```
df.select('results').distinct().show()
```

```
+--------------------+
|             results|
+--------------------+
|           Not Ready|
|                Fail|
|            No Entry|
|Business Not Located|
|  Pass w/ Conditions|
|     Out of Business|
|                Pass|
+--------------------+
```

[144]:
```
%local countResultsdf = spark.sql("SELECT COUNT(results) AS cnt, results as␣
↪results FROM CountResults GROUP BY results").toPandas()
```

```
UsageError: Line magic function `%local` not found.
```

[145]:
```
!pip install sparkmagic ipyleaflet geomet
```

```
Requirement already satisfied: sparkmagic in /opt/conda/lib/python3.7/site-
packages (0.15.0)
Requirement already satisfied: ipyleaflet in /opt/conda/lib/python3.7/site-
```

```
packages (0.12.6)
Requirement already satisfied: geomet in /opt/conda/lib/python3.7/site-packages
(0.2.1.post1)
Requirement already satisfied: notebook>=4.2 in /opt/conda/lib/python3.7/site-
packages (from sparkmagic) (6.0.3)
Requirement already satisfied: pandas>=0.17.1 in /opt/conda/lib/python3.7/site-
packages (from sparkmagic) (1.0.2)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from sparkmagic) (1.18.1)
Requirement already satisfied: tornado>=4 in /opt/conda/lib/python3.7/site-
packages (from sparkmagic) (6.0.4)
Requirement already satisfied: ipykernel in /opt/conda/lib/python3.7/site-
packages (from sparkmagic) (5.1.4)
Requirement already satisfied: mock in /opt/conda/lib/python3.7/site-packages
(from sparkmagic) (4.0.2)
Requirement already satisfied: requests in /opt/conda/lib/python3.7/site-
packages (from sparkmagic) (2.23.0)
Requirement already satisfied: ipython>=4.0.2 in /opt/conda/lib/python3.7/site-
packages (from sparkmagic) (7.13.0)
Requirement already satisfied: autovizwidget>=0.6 in
/opt/conda/lib/python3.7/site-packages (from sparkmagic) (0.15.0)
Requirement already satisfied: hdijupyterutils>=0.6 in
/opt/conda/lib/python3.7/site-packages (from sparkmagic) (0.15.0)
Requirement already satisfied: ipywidgets>5.0.0 in
/opt/conda/lib/python3.7/site-packages (from sparkmagic) (7.5.1)
Requirement already satisfied: requests-kerberos>=0.8.0 in
/opt/conda/lib/python3.7/site-packages (from sparkmagic) (0.12.0)
Requirement already satisfied: nose in /opt/conda/lib/python3.7/site-packages
(from sparkmagic) (1.3.7)
Requirement already satisfied: traittypes<3,>=0.2.1 in
/opt/conda/lib/python3.7/site-packages (from ipyleaflet) (0.2.1)
Requirement already satisfied: branca<0.4,>=0.3.1 in
/opt/conda/lib/python3.7/site-packages (from ipyleaflet) (0.3.1)
Requirement already satisfied: click in /opt/conda/lib/python3.7/site-packages
(from geomet) (7.1.1)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from geomet) (1.14.0)
Requirement already satisfied: jupyter-core>=4.6.1 in
/opt/conda/lib/python3.7/site-packages (from notebook>=4.2->sparkmagic) (4.6.3)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.7/site-packages
(from notebook>=4.2->sparkmagic) (2.11.1)
Requirement already satisfied: traitlets>=4.2.1 in
/opt/conda/lib/python3.7/site-packages (from notebook>=4.2->sparkmagic) (4.3.3)
Requirement already satisfied: ipython-genutils in
/opt/conda/lib/python3.7/site-packages (from notebook>=4.2->sparkmagic) (0.2.0)
Requirement already satisfied: nbformat in /opt/conda/lib/python3.7/site-
packages (from notebook>=4.2->sparkmagic) (5.0.4)
Requirement already satisfied: nbconvert in /opt/conda/lib/python3.7/site-
```

```
packages (from notebook>=4.2->sparkmagic) (5.6.1)
Requirement already satisfied: prometheus-client in
/opt/conda/lib/python3.7/site-packages (from notebook>=4.2->sparkmagic) (0.7.1)
Requirement already satisfied: terminado>=0.8.1 in
/opt/conda/lib/python3.7/site-packages (from notebook>=4.2->sparkmagic) (0.8.3)
Requirement already satisfied: jupyter-client>=5.3.4 in
/opt/conda/lib/python3.7/site-packages (from notebook>=4.2->sparkmagic) (6.0.0)
Requirement already satisfied: pyzmq>=17 in /opt/conda/lib/python3.7/site-
packages (from notebook>=4.2->sparkmagic) (19.0.0)
Requirement already satisfied: Send2Trash in /opt/conda/lib/python3.7/site-
packages (from notebook>=4.2->sparkmagic) (1.5.0)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.17.1->sparkmagic) (2019.3)
Requirement already satisfied: python-dateutil>=2.6.1 in
/opt/conda/lib/python3.7/site-packages (from pandas>=0.17.1->sparkmagic) (2.8.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/opt/conda/lib/python3.7/site-packages (from requests->sparkmagic) (1.25.7)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.7/site-packages (from requests->sparkmagic) (2019.11.28)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-
packages (from requests->sparkmagic) (2.9)
Requirement already satisfied: chardet<4,>=3.0.2 in
/opt/conda/lib/python3.7/site-packages (from requests->sparkmagic) (3.0.4)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from ipython>=4.0.2->sparkmagic) (3.0.4)
Requirement already satisfied: pygments in /opt/conda/lib/python3.7/site-
packages (from ipython>=4.0.2->sparkmagic) (2.6.1)
Requirement already satisfied: pexpect; sys_platform != "win32" in
/opt/conda/lib/python3.7/site-packages (from ipython>=4.0.2->sparkmagic) (4.8.0)
Requirement already satisfied: setuptools>=18.5 in
/opt/conda/lib/python3.7/site-packages (from ipython>=4.0.2->sparkmagic)
(46.0.0.post20200311)
Requirement already satisfied: jedi>=0.10 in /opt/conda/lib/python3.7/site-
packages (from ipython>=4.0.2->sparkmagic) (0.16.0)
Requirement already satisfied: backcall in /opt/conda/lib/python3.7/site-
packages (from ipython>=4.0.2->sparkmagic) (0.1.0)
Requirement already satisfied: decorator in /opt/conda/lib/python3.7/site-
packages (from ipython>=4.0.2->sparkmagic) (4.4.2)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.7/site-
packages (from ipython>=4.0.2->sparkmagic) (0.7.5)
Requirement already satisfied: plotly>=3 in /opt/conda/lib/python3.7/site-
packages (from autovizwidget>=0.6->sparkmagic) (4.7.1)
Requirement already satisfied: jupyter>=1 in /opt/conda/lib/python3.7/site-
packages (from hdijupyterutils>=0.6->sparkmagic) (1.0.0)
Requirement already satisfied: widgetsnbextension~=3.5.0 in
/opt/conda/lib/python3.7/site-packages (from ipywidgets>5.0.0->sparkmagic)
(3.5.1)
Requirement already satisfied: cryptography>=1.3; python_version != "3.3" in
```

```
/opt/conda/lib/python3.7/site-packages (from requests-
kerberos>=0.8.0->sparkmagic) (2.8)
Requirement already satisfied: pykerberos<2.0.0,>=1.1.8; sys_platform != "win32"
in /opt/conda/lib/python3.7/site-packages (from requests-
kerberos>=0.8.0->sparkmagic) (1.2.1)
Requirement already satisfied: MarkupSafe>=0.23 in
/opt/conda/lib/python3.7/site-packages (from jinja2->notebook>=4.2->sparkmagic)
(1.1.1)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in
/opt/conda/lib/python3.7/site-packages (from
nbformat->notebook>=4.2->sparkmagic) (3.2.0)
Requirement already satisfied: defusedxml in /opt/conda/lib/python3.7/site-
packages (from nbconvert->notebook>=4.2->sparkmagic) (0.6.0)
Requirement already satisfied: testpath in /opt/conda/lib/python3.7/site-
packages (from nbconvert->notebook>=4.2->sparkmagic) (0.4.4)
Requirement already satisfied: bleach in /opt/conda/lib/python3.7/site-packages
(from nbconvert->notebook>=4.2->sparkmagic) (3.1.3)
Requirement already satisfied: mistune<2,>=0.8.1 in
/opt/conda/lib/python3.7/site-packages (from
nbconvert->notebook>=4.2->sparkmagic) (0.8.4)
Requirement already satisfied: pandocfilters>=1.4.1 in
/opt/conda/lib/python3.7/site-packages (from
nbconvert->notebook>=4.2->sparkmagic) (1.4.2)
Requirement already satisfied: entrypoints>=0.2.2 in
/opt/conda/lib/python3.7/site-packages (from
nbconvert->notebook>=4.2->sparkmagic) (0.3)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.7/site-packages
(from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=4.0.2->sparkmagic)
(0.1.8)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.7/site-
packages (from pexpect; sys_platform != "win32"->ipython>=4.0.2->sparkmagic)
(0.6.0)
Requirement already satisfied: parso>=0.5.2 in /opt/conda/lib/python3.7/site-
packages (from jedi>=0.10->ipython>=4.0.2->sparkmagic) (0.6.2)
Requirement already satisfied: retrying>=1.3.3 in /opt/conda/lib/python3.7/site-
packages (from plotly>=3->autovizwidget>=0.6->sparkmagic) (1.3.3)
Requirement already satisfied: jupyter-console in /opt/conda/lib/python3.7/site-
packages (from jupyter>=1->hdijupyterutils>=0.6->sparkmagic) (6.1.0)
Requirement already satisfied: qtconsole in /opt/conda/lib/python3.7/site-
packages (from jupyter>=1->hdijupyterutils>=0.6->sparkmagic) (4.7.3)
Requirement already satisfied: cffi!=1.11.3,>=1.8 in
/opt/conda/lib/python3.7/site-packages (from cryptography>=1.3; python_version
!= "3.3"->requests-kerberos>=0.8.0->sparkmagic) (1.14.0)
Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.7/site-
packages (from jsonschema!=2.5.0,>=2.4->nbformat->notebook>=4.2->sparkmagic)
(19.3.0)
Requirement already satisfied: pyrsistent>=0.14.0 in
/opt/conda/lib/python3.7/site-packages (from
```

```
jsonschema!=2.5.0,>=2.4->nbformat->notebook>=4.2->sparkmagic) (0.15.7)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
/opt/conda/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat->notebook>=4.2->sparkmagic) (1.5.0)
Requirement already satisfied: webencodings in /opt/conda/lib/python3.7/site-
packages (from bleach->nbconvert->notebook>=4.2->sparkmagic) (0.5.1)
Requirement already satisfied: qtpy in /opt/conda/lib/python3.7/site-packages
(from qtconsole->jupyter>=1->hdijupyterutils>=0.6->sparkmagic) (1.9.0)
Requirement already satisfied: pycparser in /opt/conda/lib/python3.7/site-
packages (from cffi!=1.11.3,>=1.8->cryptography>=1.3; python_version !=
"3.3"->requests-kerberos>=0.8.0->sparkmagic) (2.20)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-
packages (from importlib-metadata; python_version <
"3.8"->jsonschema!=2.5.0,>=2.4->nbformat->notebook>=4.2->sparkmagic) (3.1.0)
```

[146]:
```
!jupyter nbextension enable --py widgetsnbextension
```

```
Enabling notebook extension jupyter-js-widgets/extension…
      - Validating: OK
```

[147]:
```
!jupyter nbextension enable --py --sys-prefix ipyleaflet
```

```
Enabling notebook extension jupyter-leaflet/extension…
      - Validating: OK
```

[148]:
```
!jupyter-kernelspec install $(pip show sparkmagic | grep Location | cut -d" "␣
↪-f2)/sparkmagic/kernels/sparkkernel --user
!jupyter-kernelspec install $(pip show sparkmagic | grep Location | cut -d" "␣
↪-f2)/sparkmagic/kernels/pysparkkernel --user
!jupyter-kernelspec install $(pip show sparkmagic | grep Location | cut -d" "␣
↪-f2)/sparkmagic/kernels/pyspark3kernel --user
!jupyter-kernelspec install $(pip show sparkmagic | grep Location | cut -d" "␣
↪-f2)/sparkmagic/kernels/sparkrkernel --user
```

```
[InstallKernelSpec] Removing existing kernelspec in
/home/jovyan/.local/share/jupyter/kernels/sparkkernel
[InstallKernelSpec] Installed kernelspec sparkkernel in
/home/jovyan/.local/share/jupyter/kernels/sparkkernel
[InstallKernelSpec] Removing existing kernelspec in
/home/jovyan/.local/share/jupyter/kernels/pysparkkernel
[InstallKernelSpec] Installed kernelspec pysparkkernel in
/home/jovyan/.local/share/jupyter/kernels/pysparkkernel
Traceback (most recent call last):
  File "/opt/conda/bin/jupyter-kernelspec", line 10, in <module>
    sys.exit(KernelSpecApp.launch_instance())
  File "/opt/conda/lib/python3.7/site-packages/traitlets/config/application.py",
line 664, in launch_instance
    app.start()
```

```
  File "/opt/conda/lib/python3.7/site-packages/jupyter_client/kernelspecapp.py",
line 268, in start
    return self.subapp.start()
  File "/opt/conda/lib/python3.7/site-packages/jupyter_client/kernelspecapp.py",
line 138, in start
    replace=self.replace,
  File "/opt/conda/lib/python3.7/site-packages/jupyter_client/kernelspec.py",
line 345, in install_kernel_spec
    shutil.copytree(source_dir, destination)
  File "/opt/conda/lib/python3.7/shutil.py", line 318, in copytree
    names = os.listdir(src)
FileNotFoundError: [Errno 2] No such file or directory:
'/opt/conda/lib/python3.7/site-packages/sparkmagic/kernels/pyspark3kernel'
[InstallKernelSpec] Removing existing kernelspec in
/home/jovyan/.local/share/jupyter/kernels/sparkrkernel
[InstallKernelSpec] Installed kernelspec sparkrkernel in
/home/jovyan/.local/share/jupyter/kernels/sparkrkernel
```

[149]:
```
!jupyter serverextension enable --py sparkmagic
```

```
Enabling: sparkmagic
- Writing config: /home/jovyan/.jupyter
    - Validating…
      sparkmagic 0.15.0 OK
```

[150]:
```
!sed -i -e 's/return self._pyspark_command(sql_context_variable_name)/return␣
↪self._pyspark_command(sql_context_variable_name, False)/g' $(pip show␣
↪sparkmagic | grep Location | cut -d" " -f2)/sparkmagic/livyclientlib/
↪sqlquery.py

!rm -rf $(pip show sparkmagic | grep Location | cut -d" " -f2)/sparkmagic/
↪livyclientlib/sqlquery.py-e
```

[151]:
```
%local countResultsdf = spark.sql("SELECT COUNT(results) AS cnt, results as␣
↪results FROM CountResults GROUP BY results").toPandas()
```

```
UsageError: Line magic function `%local` not found.
```

[152]:
```
%lsmagic
```

[152]:
```
Available line magics:
%alias  %alias_magic  %autoawait  %autocall  %automagic  %autosave  %bookmark
%cat  %cd  %clear  %colors  %conda  %config  %connect_info  %cp  %debug  %dhist
%dirs  %doctest_mode  %ed  %edit  %env  %gui  %hist  %history  %killbgscripts
%ldir  %less  %lf  %lk  %ll  %load  %load_ext  %loadpy  %logoff  %logon
%logstart  %logstate  %logstop  %ls  %lsmagic  %lx  %macro  %magic  %man
%matplotlib  %mkdir  %more  %mv  %notebook  %page  %pastebin  %pdb  %pdef  %pdoc
%pfile  %pinfo  %pinfo2  %pip  %popd  %pprint  %precision  %prun  %psearch
```

```
%psource  %pushd  %pwd  %pycat  %pylab  %qtconsole  %quickref  %recall  %rehashx
%reload_ext  %rep  %rerun  %reset  %reset_selective  %rm  %rmdir  %run  %save
%sc  %set_env  %store  %sx  %system  %tb  %time  %timeit  %unalias  %unload_ext
%who  %who_ls  %whos  %xdel  %xmode

Available cell magics:
%%!  %%HTML  %%SVG  %%bash  %%capture  %%debug  %%file  %%html  %%javascript
%%js  %%latex  %%markdown  %%perl  %%prun  %%pypy  %%python  %%python2
%%python3  %%ruby  %%script  %%sh  %%svg  %%sx  %%system  %%time  %%timeit
%%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

[153]:
```
# Укажем без magic function %
countResultsdf = spark.sql("SELECT COUNT(results) AS cnt, results as results␣
 ↪FROM CountResults GROUP BY results").toPandas()
```

[154]:
```
countResultsdf
```

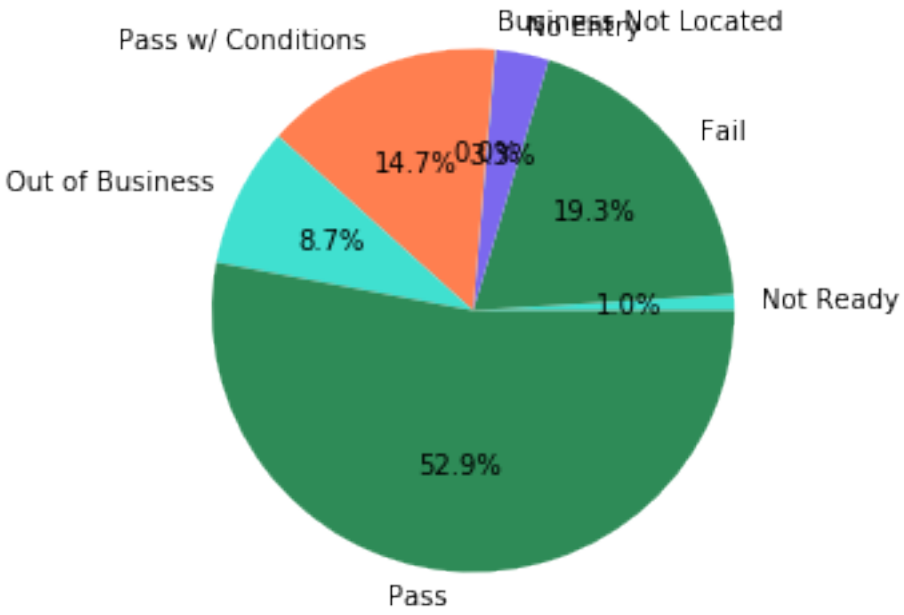
[154]:

	cnt	results
0	2054	Not Ready
1	39699	Fail
2	6812	No Entry
3	72	Business Not Located
4	30264	Pass w/ Conditions
5	17799	Out of Business
6	108797	Pass

[155]:
```
%matplotlib inline
import matplotlib.pyplot as plt

labels = countResultsdf['results']
sizes = countResultsdf['cnt']
colors = ['turquoise', 'seagreen', 'mediumslateblue', 'palegreen', 'coral']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=colors)
plt.axis('equal')
```

[155]:
```
(-1.1089530605705424,
 1.1004263643582002,
 -1.1094731819512265,
 1.1103504043710961)
```



[156]:
```
def labelForResults(s):
    if s == 'Fail':
        return 0.0
    elif s == 'Pass w/ Conditions' or s == 'Pass':
        return 1.0
    else:
        return -1.0
label = UserDefinedFunction(labelForResults, DoubleType())
labeledData = df.select(label(df.results).alias('label'), df.violations).
 ↪where('label >= 0').fillna("null")
```

[158]:
```
labeledDataSplits = labeledData.randomSplit([0.8, 0.2])
trainData = labeledDataSplits[0]
testData = labeledDataSplits[1]
```

[159]:
```
trainData.take(1)
```

[159]:
```
[Row(label=0.0, violations='1. PERSON IN CHARGE PRESENT, DEMONSTRATES KNOWLEDGE,
AND PERFORMS DUTIES - Comments:  2-101.11 PERSON IN CHARGE DOES NOT POSSESS A
CITY OF CHICAGO SANITATION CERTIFICATE. INSTRUCTED ALL PERSONS IN CHARGE MUST
HAVE A CITY SANITATION CERTIFICATE. PRIORITY FOUNDATION VIOLATION. NO CITATION
ISSUED. | 25. CONSUMER ADVISORY PROVIDED FOR RAW/UNDERCOOKED FOOD - Comments:
3-603.11 MENU DOES NOT DISCLOSE AND INFORM CONSUMERS THE SPECIFIC MENU ITEMS
THAT ARE RAW OR UNDER COOKED AND A POTENTIAL HAZARD OF CONSUMING SUCH FOOD. MUST
PROVIDE A CONSUMER ADVISORY AND DISCLOSE SUCH ITEMS. PRIORITY FOUNDATION
VIOLATION. NO CITATION ISSUED. | 37. FOOD PROPERLY LABELED; ORIGINAL CONTAINER -
```

```
Comments:  3-602.11(A; B:1-4,6,7; C; D) ALL PREPACKAGED PREPARED FOODS IN THE
FRONT DISPLAY COOLER SUCH AS: PASTA SALAD, SANDWICHES, GREEN SALADS MUST BE
PROPERLY LABELED WITH BUSINESS NAME, ADDRESS AND LIST OF INGREDIENTS. | 37. FOOD
PROPERLY LABELED; ORIGINAL CONTAINER - Comments:  3-602.11(B:5) ALL PREPACKAGED
GRANOLA ON DISPLAY FOR SALE CONTAINING NUTS MUST BE PROPERLY LABELED FOR
ALLERGENS. PRIORITY FOUNDATION VIOLATION. NO CITATION ISSUED. | 40. PERSONAL
CLEANLINESS - Comments: : 2-402.11 ALL EMPLOYEES PREPARING FOODS/COFFEES MUST
WEAR EFFECTIVE HAIR AND BEARD RESTRAINTS.  | 41. WIPING CLOTHS: PROPERLY USED &
STORED - Comments: : 3-304.14 ALL PREP AREA WET WIPING CLOTHS MUST BE HELD
INSIDE CONTAINERS WITH A SANITIZING SOLUTION BETWEEN USE. | 47. FOOD & NON-FOOD
CONTACT SURFACES CLEANABLE, PROPERLY DESIGNED, CONSTRUCTED & USED - Comments:
4-501.13 NOTED THE BROKEN MICROWAVE OVEN REPAIRED WITH TAPE. MUST PROPERLY
REPAIR OR REPLACE. | 48. WAREWASHING FACILITIES: INSTALLED, MAINTAINED & USED;
TEST STRIPS - Comments:  4-302.14 NO CHLORINE CHEMICAL TEST STRIPS ON SITE FOR
THE CHLORINE SANITIZING DISH WASHING MACHINE. MUST PROVIDE. PRIORITY FOUNDATION
VIOLATION 7-38-005 CITATION ISSUED. | 51. PLUMBING INSTALLED; PROPER BACKFLOW
DEVICES - Comments: : 5-205.15 LEAK ON THE HOT HANDLE OF THE BAKERY PREP HAND
WASHING SINK. MUST REPAIR. | 55. PHYSICAL FACILITIES INSTALLED, MAINTAINED &
CLEAN - Comments: : 6-201.11 FLOORS THROUGHOUT THE FRONT PREP AREA WITH DIRT AND
DEBRIS. FLOOR DRAIN UNDER THE THREE COMPARTMENT SINK DIRTY. MUST CLEAN AND
MAINTAIN ALL. | 60. PREVIOUS CORE VIOLATION CORRECTED - Comments:  8-404.13(B:4)
PREVIOUS CORE VIOLATIONS FROM #2232476 ON 10-30-18 NOT CORRECTED. #57- OBSERVED
FACILITY MISSING REQUIRED FOOD HANDLER TRAINING CERTIFICATION FOR ALL FOOD
HANDLER EMPLOYEES. FACILITY MUST OBTAIN AND MAINTAIN ALL REQUIRED FOOD HANDLER
TRAINING CERTIFICATION FOR ALL FOOD HANDLER EMPLOYEES ON SITE FOR INSPECTION
REVIEW. #58- OBSERVED FACILITY FOOD SERVICE SANITATION MANAGERS WITHOUT REQUIRED
ALLERGEN TRAINING CERTIFICATION. MUST OBTAIN AND MAINTAIN. PRIORITY FOUNDATION
VIOLATION 7-42-090 CITATION ISSUED. ')]
```

[160]:
```
tokenizer = Tokenizer(inputCol="violations", outputCol="words")
hashingTF = HashingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
lr = LogisticRegression(maxIter=10, regParam=0.01)
pipeline = Pipeline(stages=[tokenizer, hashingTF, lr])

model = pipeline.fit(labeledData)
```

[161]:
```
predictionsDf = model.transform(testData)
predictionsDf.registerTempTable('Predictions')
predictionsDf.columns
```

[161]:
```
['label',
 'violations',
 'words',
 'features',
 'rawPrediction',
 'probability',
 'prediction']
```

[162]:
```
predictionsDf.take(1)
```

[162]:
```
[Row(label=0.0, violations="1. PERSON IN CHARGE PRESENT, DEMONSTRATES KNOWLEDGE,
AND PERFORMS DUTIES - Comments:  PERSON IN CHARGE AT THE TIME OF INSPECTION DOES
NOT HAVE A CITY OF CHICAGO SANITATION CERTIFICATE. PRIORITY FOUNDATION
VIOLATION. NO CITATION ISSUED - | 3. MANAGEMENT, FOOD EMPLOYEE AND CONDITIONAL
EMPLOYEE; KNOWLEDGE, RESPONSIBILITIES AND REPORTING - Comments: FOUND NO
EMPLOYEE HEALTH POLICY/TRAINING ON SITE. INSTRUCTED FACILITY TO ESTABLISH AN
APPROPRIATE EMPLOYEE HEALTH POLICY/TRAINING SYSTEM AND MAINTAIN WITH VERIFIABLE
DOCUMENTS ON SITE. PRIORITY FOUNDATION VIOLATION 7-38-010. NO CITATION ISSUED |
5. PROCEDURES FOR RESPONDING TO VOMITING AND DIARRHEAL EVENTS - Comments: FOUND
NO PROCEDURE/PLAN AND KIT FOR RESPONDING TO VOMITING AND DIARRHEAL EVENTS.
INSTRUCTED FACILITY TO DEVELOP AND MAINTAIN A PROCEDURE/PLAN AND KIT TO MAINTAIN
ANY APPROPRIATE SUPPLIES ON SITE. PRIORITY FOUNDATION VIOLATION 7-38-005.NO
CITATION ISSUED   | 10. ADEQUATE HANDWASHING SINKS PROPERLY SUPPLIED AND
ACCESSIBLE - Comments:  OBSERVED HANDWASHING INSTALLED AT FRONT PREP/SERVING
AREA NOT CONVENIENT TO EMPLOYEE(PRESENT HANDWASHING SINK IS LOCATED AT WEST SIDE
OF THE FRONT COUNTER BY THE THREE COMPARTMENT SINK),PREP/SERVING AND CASH
REGISTRAR IS LOCATED AT EAST SIDE OF THE FRONT COUNTER. INSTRUCTED TO INSTALL
ANOTHER HANDWASHING SINK AT OTHER SIDE OF COUNTER BY THE PREP/SERVING AND CASH
REGISTRAR AREA. PRIORITY FOUNDATION VIOLATION:7-38-030(C),NO CITATION ISSUED |
41. WIPING CLOTHS: PROPERLY USED & STORED - Comments: LINENS MUST BE HELD
BETWEEN USE IN A CONTAINER WITH A SANITIZING SOLUTION TO WIPE MULTI USE UTENSILS
/CUTTING BOARDS ETC. | 55. PHYSICAL FACILITIES INSTALLED, MAINTAINED & CLEAN -
Comments: DEBRIS ON FLOOR UNDER SHELVES AND ALONG BASEBOARD WALLS AT SECOND
FLOOR DRY STORAGE,ALSO REMOVE STOCK STORED ON FLOOR, ELEVATE STOCKS 4' ABOVE THE
FLOOR BY PROVIDING RAISED SHELVES | 57. ALL FOOD EMPLOYEES HAVE FOOD HANDLER
TRAINING - Comments: NO FOOD HANDLER TRAINING CERTIFICATE PROVIDED ON SITE
,INSTRUCTED TO PROVIDE   | 59. PREVIOUS PRIORITY FOUNDATION VIOLATION CORRECTED -
Comments: PREVIOUS PRIORITY FOUNDATION VIOLATION NOT CORRECTED FROM
7-25-19,REPORT#2303727 WE OBSERVED BLACK SLIME SUBSTANCE AT INTERIOR OF ICE
MACHINE.SLIME SUBSTANCE DRIPPING ON ICE AT ICE BIN ATTACHED TO UNIT .ICE IS USED
FOR HUMAN CONSUMPTION. INSTRUCTED TO CLEAN,SANITIZE AND MAINTAIN UNIT.  PRIORITY
VIOLATION:7-38-005,CITATION ISSUED -  TODAY 8-2-19 STILL  WE OBSERVED BLACK
SLIME SUBSTANCE INSIDE THE ICE MACHINE. | 61. SUMMARY REPORT DISPLAYED AND
VISIBLE TO THE PUBLIC - Comments: PREVIOUS INSPECTION SUMMARY REPORT FROM
7-25-19,REPORT #2303727 NO DISPLAYED AND VISIBLE TO ALL CUSTOMERS.NEW INSPECTION
SUMMARY REPORT GIVEN AND INSTRUCTED TO DISPLAY PRIORITY FOUNDATION
VIOLATION:7-42-010(B)", words=['1.', 'person', 'in', 'charge', 'present,',
'demonstrates', 'knowledge,', 'and', 'performs', 'duties', '-', 'comments:', '',
'person', 'in', 'charge', 'at', 'the', 'time', 'of', 'inspection', 'does',
'not', 'have', 'a', 'city', 'of', 'chicago', 'sanitation', 'certificate.',
'priority', 'foundation', 'violation.', 'no', 'citation', 'issued', '-', '|',
'3.', 'management,', 'food', 'employee', 'and', 'conditional', 'employee;',
'knowledge,', 'responsibilities', 'and', 'reporting', '-', 'comments:', 'found',
'no', 'employee', 'health', 'policy/training', 'on', 'site.', 'instructed',
'facility', 'to', 'establish', 'an', 'appropriate', 'employee', 'health',
```

'policy/training', 'system', 'and', 'maintain', 'with', 'verifiable',
'documents', 'on', 'site.', 'priority', 'foundation', 'violation', '7-38-010.',
'no', 'citation', 'issued', '|', '5.', 'procedures', 'for', 'responding', 'to',
'vomiting', 'and', 'diarrheal', 'events', '-', 'comments:', 'found', 'no',
'procedure/plan', 'and', 'kit', 'for', 'responding', 'to', 'vomiting', 'and',
'diarrheal', 'events.', 'instructed', 'facility', 'to', 'develop', 'and',
'maintain', 'a', 'procedure/plan', 'and', 'kit', 'to', 'maintain', 'any',
'appropriate', 'supplies', 'on', 'site.', 'priority', 'foundation', 'violation',
'7-38-005.no', 'citation', 'issued', '', '', '|', '10.', 'adequate',
'handwashing', 'sinks', 'properly', 'supplied', 'and', 'accessible', '-',
'comments:', '', 'observed', 'handwashing', 'installed', 'at', 'front',
'prep/serving', 'area', 'not', 'convenient', 'to', 'employee(present',
'handwashing', 'sink', 'is', 'located', 'at', 'west', 'side', 'of', 'the',
'front', 'counter', 'by', 'the', 'three', 'compartment', 'sink),prep/serving',
'and', 'cash', 'registrar', 'is', 'located', 'at', 'east', 'side', 'of', 'the',
'front', 'counter.', 'instructed', 'to', 'install', 'another', 'handwashing',
'sink', 'at', 'other', 'side', 'of', 'counter', 'by', 'the', 'prep/serving',
'and', 'cash', 'registrar', 'area.', 'priority', 'foundation',
'violation:7-38-030(c),no', 'citation', 'issued', '|', '41.', 'wiping',
'cloths:', 'properly', 'used', '&', 'stored', '-', 'comments:', 'linens',
'must', 'be', 'held', 'between', 'use', 'in', 'a', 'container', 'with', 'a',
'sanitizing', 'solution', 'to', 'wipe', 'multi', 'use', 'utensils', '/cutting',
'boards', 'etc.', '|', '55.', 'physical', 'facilities', 'installed,',
'maintained', '&', 'clean', '-', 'comments:', 'debris', 'on', 'floor', 'under',
'shelves', 'and', 'along', 'baseboard', 'walls', 'at', 'second', 'floor', 'dry',
'storage,also', 'remove', 'stock', 'stored', 'on', 'floor,', 'elevate',
'stocks', "4'", 'above', 'the', 'floor', 'by', 'providing', 'raised', 'shelves',
'|', '57.', 'all', 'food', 'employees', 'have', 'food', 'handler', 'training',
'-', 'comments:', 'no', 'food', 'handler', 'training', 'certificate',
'provided', 'on', 'site', ',instructed', 'to', 'provide', '', '|', '59.',
'previous', 'priority', 'foundation', 'violation', 'corrected', '-',
'comments:', 'previous', 'priority', 'foundation', 'violation', 'not',
'corrected', 'from', '7-25-19,report#2303727', 'we', 'observed', 'black',
'slime', 'substance', 'at', 'interior', 'of', 'ice', 'machine.slime',
'substance', 'dripping', 'on', 'ice', 'at', 'ice', 'bin', 'attached', 'to',
'unit', '.ice', 'is', 'used', 'for', 'human', 'consumption.', 'instructed',
'to', 'clean,sanitize', 'and', 'maintain', 'unit.', '', 'priority',
'violation:7-38-005,citation', 'issued', '-', '', 'today', '8-2-19', 'still',
'', 'we', 'observed', 'black', 'slime', 'substance', 'inside', 'the', 'ice',
'machine.', '|', '61.', 'summary', 'report', 'displayed', 'and', 'visible',
'to', 'the', 'public', '-', 'comments:', 'previous', 'inspection', 'summary',
'report', 'from', '7-25-19,report', '#2303727', 'no', 'displayed', 'and',
'visible', 'to', 'all', 'customers.new', 'inspection', 'summary', 'report',
'given', 'and', 'instructed', 'to', 'display', 'priority', 'foundation',
'violation:7-42-010(b)'], features=SparseVector(262144, {2325: 3.0, 4089: 1.0,
5862: 1.0, 6025: 1.0, 6106: 1.0, 7724: 1.0, 8500: 3.0, 9125: 1.0, 9639: 6.0,
10430: 1.0, 11251: 4.0, 11910: 1.0, 12336: 1.0, 12445: 1.0, 13289: 1.0, 14311:

```
9.0, 15889: 3.0, 16181: 1.0, 16256: 1.0, 16332: 3.0, 17584: 1.0, 19212: 1.0,
20495: 1.0, 21028: 1.0, 21316: 1.0, 21471: 1.0, 24145: 1.0, 24973: 1.0, 28282:
1.0, 30844: 2.0, 31735: 1.0, 36200: 1.0, 37728: 2.0, 38765: 3.0, 40268: 2.0,
40830: 1.0, 41170: 1.0, 43298: 1.0, 43583: 1.0, 43815: 1.0, 45531: 11.0, 46498:
2.0, 50217: 1.0, 50223: 4.0, 50849: 2.0, 53943: 1.0, 54205: 1.0, 56667: 1.0,
56804: 1.0, 58141: 1.0, 58162: 1.0, 58241: 2.0, 58262: 1.0, 59755: 1.0, 63050:
1.0, 63091: 5.0, 63241: 1.0, 66250: 1.0, 69718: 1.0, 69998: 1.0, 72516: 3.0,
80333: 3.0, 81631: 1.0, 83300: 1.0, 86752: 1.0, 88105: 1.0, 90757: 2.0, 90859:
1.0, 91677: 17.0, 94851: 1.0, 95122: 1.0, 95866: 1.0, 95906: 1.0, 97260: 1.0,
98373: 1.0, 99346: 1.0, 100258: 7.0, 100952: 1.0, 101169: 2.0, 102675: 1.0,
103382: 8.0, 103838: 8.0, 104153: 1.0, 104659: 4.0, 105063: 1.0, 107950: 1.0,
108647: 1.0, 113458: 2.0, 114381: 3.0, 115062: 1.0, 115218: 1.0, 116873: 2.0,
121133: 4.0, 121356: 1.0, 121424: 1.0, 121517: 1.0, 123069: 3.0, 123938: 1.0,
124643: 2.0, 125011: 1.0, 125353: 2.0, 126466: 2.0, 127370: 2.0, 129461: 3.0,
133143: 3.0, 133156: 1.0, 134024: 2.0, 135499: 1.0, 135560: 2.0, 135568: 1.0,
138751: 1.0, 139098: 3.0, 140737: 1.0, 141488: 1.0, 141683: 2.0, 143894: 1.0,
145542: 1.0, 145624: 1.0, 145697: 1.0, 145838: 1.0, 146227: 2.0, 147024: 1.0,
147489: 2.0, 152685: 2.0, 153032: 1.0, 153581: 1.0, 156250: 6.0, 159390: 1.0,
159775: 2.0, 161061: 2.0, 161088: 1.0, 165823: 3.0, 167152: 1.0, 167656: 1.0,
168425: 2.0, 168592: 1.0, 169961: 1.0, 176964: 8.0, 178880: 1.0, 185591: 1.0,
187337: 1.0, 188373: 1.0, 193224: 1.0, 193313: 2.0, 193347: 1.0, 194536: 1.0,
194821: 2.0, 200018: 1.0, 200400: 4.0, 201846: 3.0, 202732: 1.0, 203018: 7.0,
203609: 1.0, 204253: 2.0, 205044: 14.0, 205305: 2.0, 205324: 1.0, 205340: 1.0,
205349: 1.0, 208279: 2.0, 212952: 5.0, 214482: 1.0, 215647: 1.0, 215944: 2.0,
216058: 1.0, 217720: 1.0, 218380: 1.0, 218391: 1.0, 218825: 3.0, 218829: 1.0,
219136: 8.0, 220960: 1.0, 222453: 3.0, 225305: 4.0, 225574: 2.0, 227410: 4.0,
227642: 1.0, 229407: 1.0, 229543: 1.0, 230443: 2.0, 230962: 1.0, 232128: 1.0,
233971: 2.0, 237325: 2.0, 239774: 1.0, 243536: 1.0, 244670: 2.0, 249180: 8.0,
250521: 1.0, 251195: 2.0, 252167: 1.0, 252272: 1.0, 252378: 1.0, 252417: 1.0,
252637: 1.0, 253475: 2.0, 254742: 1.0, 256070: 1.0, 259167: 2.0, 259928: 2.0}),
rawPrediction=DenseVector([6.0235, -6.0235]), probability=DenseVector([0.9976,
0.0024]), prediction=0.0)]
```

[163]:
```
numSuccesses = predictionsDf.where("""(prediction = 0 AND results = 'Fail') OR
                                      (prediction = 1 AND (results = 'Pass' OR
                                                           results = 'Pass w/␣
 ↪Conditions'))""").count()
numInspections = predictionsDf.count()

print("There were", numInspections, "inspections and there were", numSuccesses,␣
 ↪"successful predictions")
print("This is a", str((float(numSuccesses) / float(numInspections)) * 100) +␣
 ↪"%", "success rate")
```

```
There were 35742 inspections and there were 34643 successful predictions
This is a 96.92518605562084% success rate
```

[164]:
```
true_positive = predictionsDf.where("prediction = 0 AND results = 'Fail'").
↪count()
```

[165]:
```
false_positive = predictionsDf.where("prediction = 0 AND (results = 'Pass' OR␣
↪results = 'Pass w/ Conditions')").count()
```

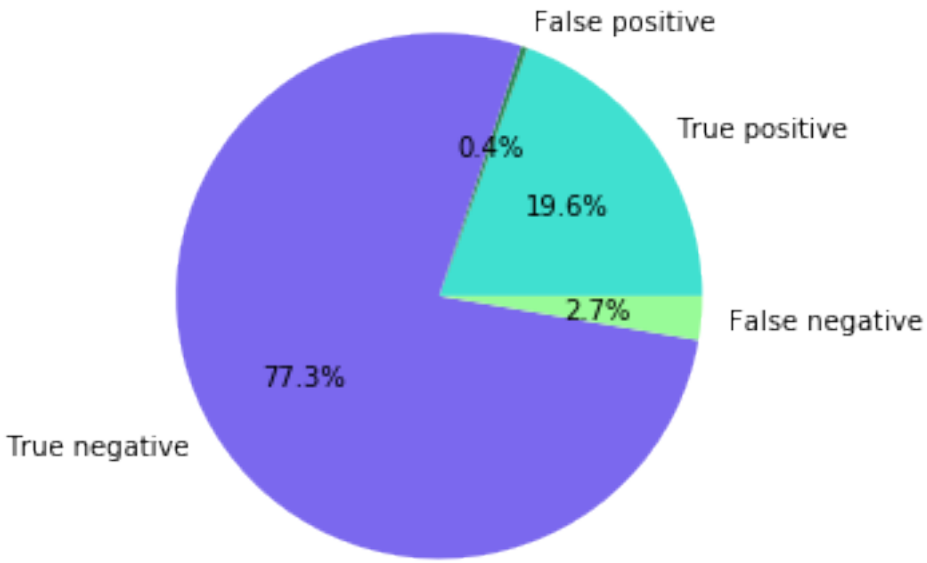
[166]:
```
true_negative = predictionsDf.where("prediction = 1 AND results = 'Fail'").
↪count()
```

[167]:
```
false_negative = predictionsDf.where("prediction = 1 AND (results = 'Pass' OR␣
↪results = 'Pass w/ Conditions')").count()
```

[168]:
```
%matplotlib inline
import matplotlib.pyplot as plt

labels = ['True positive', 'False positive', 'True negative', 'False negative']
sizes = [true_positive, false_positive, false_negative, true_negative]
colors = ['turquoise', 'seagreen', 'mediumslateblue', 'palegreen', 'coral']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=colors)
plt.axis('equal')
```

[168]:
```
(-1.1047212236212636,
 1.1002248201724412,
 -1.1053092053565108,
 1.101589266482251)
```



[ ]: