电子科技大学信息与软件工程学院

**项 目 报 告**

课程名称 大数据分析与智能计算

理论教师

实验教师

学生信息：

|  |  |  |
| --- | --- | --- |
| 序号 | 学号 | 姓名 |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**项 目 报 告**

**指导教师： 地点：**

1. **项目名称：航空公司****延误和取消分析项目**
2. **项目时间：2022.12.2—2022.1.2**
3. **项目原理**

# 四、 项目内容

# 五、 需求分析与设计

（需求分析描述、用例图、用例规约、数据流图、顺序图、流程图）

# 六、项目计划

# 七、项目环境配置管理

**7.1** **操作系统**：

**7.2 开发工具**：

**7.3 配置过程**

# 八、项目实践过程

（代码示例格式如下：）

建立特征索引，其作用是XXXXXXX，代码如下所示。

代码1 建立特征索引

|  |
| --- |
| # 建立特征索引  from pyspark.ml.feature import VectorIndexer  featureIndexer = VectorIndexer(inputCol="features", outputCol="indexedFeatures", maxCategories=24).fit(dataFrame) |

1. **问题三**

（1）问题分析

问题三要求我们分别“分析一天中、一周中延误最严重的飞行时间”，首先经过分析，我们理解题意为需要得到的数据结果为数据集内每一天、每一周中的最高延误时间，所以我们选取特征为年Year、月Month、月中的某日DayofMonth、星期几DayOfWeek、抵达延误时间ArrDelay，先算出每一天的最高延误时间，再据此算出每一周的最高延误时间。

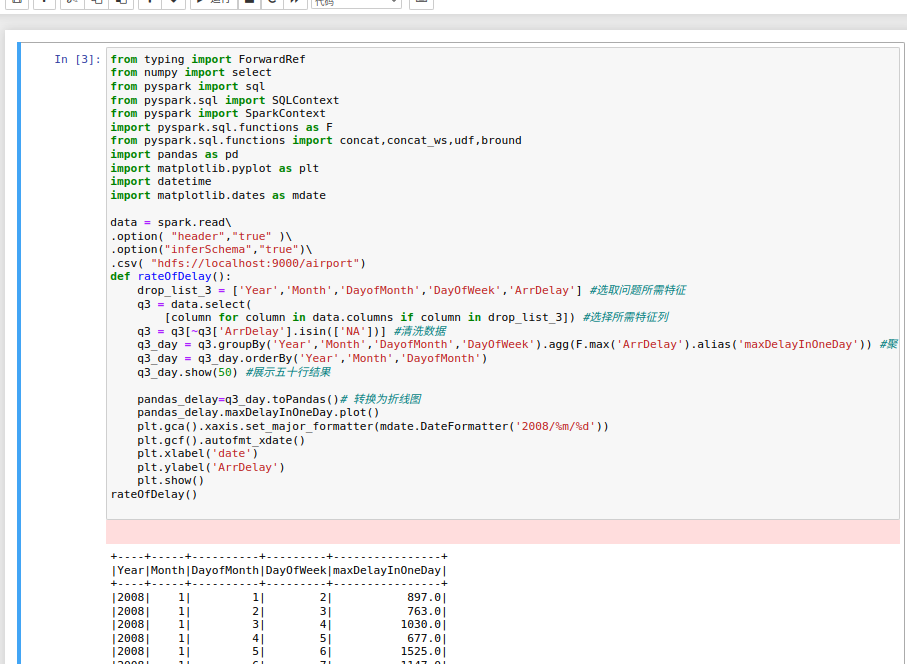
（2）代码

A.每一天的最高延误时间(只显示前50行)

代码 3

|  |
| --- |
| from typing import ForwardRef  from numpy import select  from pyspark import sql  from pyspark.sql import SQLContext  from pyspark import SparkContext  import pyspark.sql.functions as F  from pyspark.sql.functions import concat,concat\_ws,udf,bround  import pandas as pd  import matplotlib.pyplot as plt  import datetime  import matplotlib.dates as mdate  data = spark.read\  .option( "header","true" )\  .option("inferSchema","true")\  .csv( "hdfs://localhost:9000/airport")  def rateOfDelay():  drop\_list\_3 = ['Year','Month','DayofMonth','DayOfWeek','ArrDelay'] #选取问题所需特征  q3 = data.select(  [column for column in data.columns if column in drop\_list\_3]) #选择所需特征列  q3 = q3[~q3['ArrDelay'].isin(['NA'])] #清洗数据  q3\_day = q3.groupBy('Year','Month','DayofMonth','DayOfWeek').agg(F.max('ArrDelay').alias('maxDelayInOneDay')) #聚合每天的抵达延误时间进行比较，选出最大值加入“maxDelayInOneDay”列中  q3\_day = q3\_day.orderBy('Year','Month','DayofMonth')  q3\_day.show(50) #展示五十行结果    pandas\_delay=q3\_day.toPandas()# 转换为折线图  pandas\_delay.maxDelayInOneDay.plot()  plt.gca().xaxis.set\_major\_formatter(mdate.DateFormatter('2008/%m/%d'))  plt.gcf().autofmt\_xdate()  plt.xlabel('date')  plt.ylabel('ArrDelay')  plt.show()  rateOfDelay() |

运行界面如下：

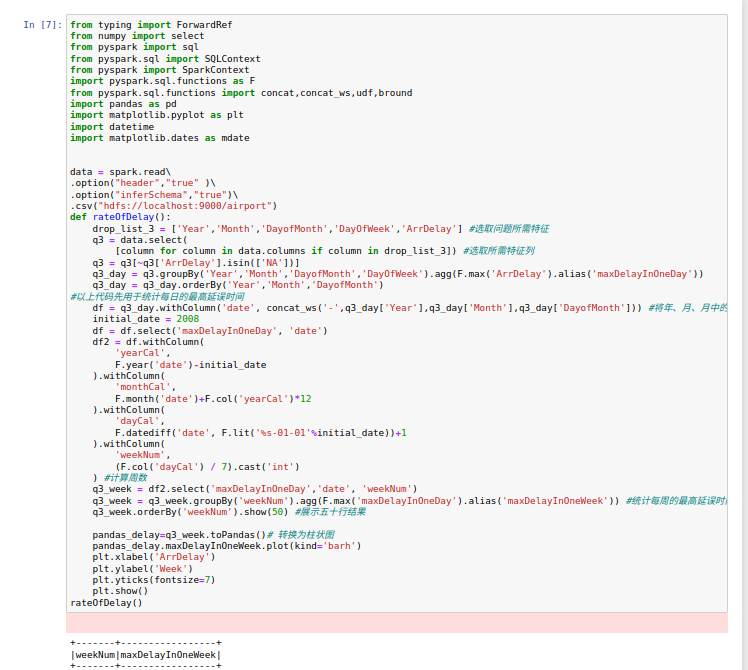


B. 每一周的最高延误时间(只显示前50行)

代码 4

|  |
| --- |
| from typing import ForwardRef  from numpy import select  from pyspark import sql  from pyspark.sql import SQLContext  from pyspark import SparkContext  import pyspark.sql.functions as F  from pyspark.sql.functions import concat,concat\_ws,udf,bround  import pandas as pd  import matplotlib.pyplot as plt  import datetime  import matplotlib.dates as mdate  data = spark.read\  .option("header","true" )\  .option("inferSchema","true")\  .csv("hdfs://localhost:9000/airport")  def rateOfDelay():  drop\_list\_3 = ['Year','Month','DayofMonth','DayOfWeek','ArrDelay'] #选取问题所需特征  q3 = data.select(  [column for column in data.columns if column in drop\_list\_3]) #选取所需特征列  q3 = q3[~q3['ArrDelay'].isin(['NA'])]  q3\_day = q3.groupBy('Year','Month','DayofMonth','DayOfWeek').agg(F.max('ArrDelay').alias('maxDelayInOneDay'))  q3\_day = q3\_day.orderBy('Year','Month','DayofMonth')  #以上代码先用于统计每日的最高延误时间  df = q3\_day.withColumn('date', concat\_ws('-',q3\_day['Year'],q3\_day['Month'],q3\_day['DayofMonth'])) #将年、月、月中的第几日用分隔符“-”连接，与date并列  initial\_date = 2008  df = df.select('maxDelayInOneDay', 'date')  df2 = df.withColumn(  'yearCal',  F.year('date')-initial\_date  ).withColumn(  'monthCal',  F.month('date')+F.col('yearCal')\*12  ).withColumn(  'dayCal',  F.datediff('date', F.lit('%s-01-01'%initial\_date))+1  ).withColumn(  'weekNum',  (F.col('dayCal') / 7).cast('int')  ) #计算周数  q3\_week = df2.select('maxDelayInOneDay','date', 'weekNum')  q3\_week = q3\_week.groupBy('weekNum').agg(F.max('maxDelayInOneDay').alias('maxDelayInOneWeek')) #统计每周的最高延误时间  q3\_week.orderBy('weekNum').show(50) #展示五十行结果    pandas\_delay=q3\_week.toPandas()# 转换为柱状图  pandas\_delay.maxDelayInOneWeek.plot(kind='barh')  plt.xlabel('ArrDelay')  plt.ylabel('Week')  plt.yticks(fontsize=7)  plt.show()  rateOfDelay() |

运行程序界面如下：



1. **问题四**
2. 问题分析

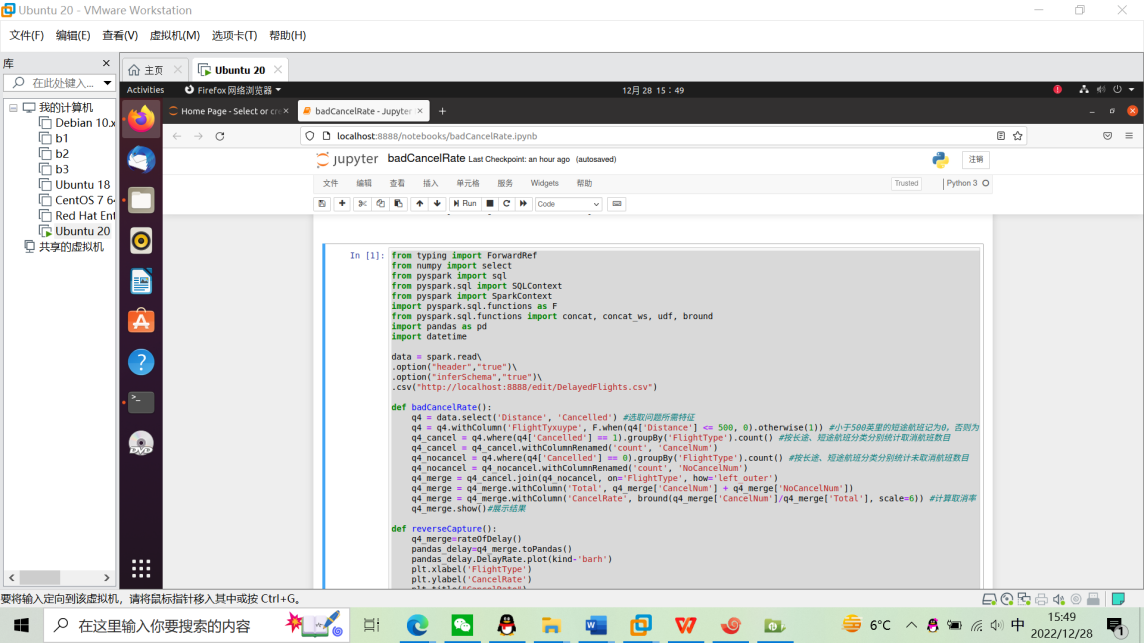
这一问要求判断“短途航班和长途航班，哪种航班取消更严重”，我们对该问题的分析为首先拟定短途和长途航班的距离分割界限为500英里，然后首先对所有航班进行短途和长途的分类，再分别同济两类中的是否取消情况，据此分析，我们需要的特征有**航行距离Distance、航班是否取消Cancelled**。

1. 代码

代码 5

|  |
| --- |
| from typing import ForwardRef  from numpy import select  from pyspark import sql  from pyspark.sql import SQLContext  from pyspark import SparkContext  import pyspark.sql.functions as F  from pyspark.sql.functions import concat, concat\_ws, udf, bround  import pandas as pd  import datetim0e  data = spark.read\  .option("header","true")\  .option("inferSchema","true")\  .csv("http://localhost:8888/edit/DelayedFlights.csv")  def badCancelRate():  q4 = data.select('Distance', 'Cancelled') #选取问题所需特征  q4 = q4.withColumn('FlightTyxuype', F.when(q4['Distance'] <= 500, 0).otherwise(1)) #小于500英里的短途航班记为0，否则为长途航班，记为1  q4\_cancel = q4.where(q4['Cancelled'] == 1).groupBy('FlightType').count() #按长途、短途航班分类分别统计取消航班数目  q4\_cancel = q4\_cancel.withColumnRenamed('count', 'CancelNum')  q4\_nocancel = q4.where(q4['Cancelled'] == 0).groupBy('FlightType').count() #按长途、短途航班分类分别统计未取消航班数目  q4\_nocancel = q4\_nocancel.withColumnRenamed('count', 'NoCancelNum')  q4\_merge = q4\_cancel.join(q4\_nocancel, on='FlightType', how='left\_outer')  q4\_merge = q4\_merge.withColumn('Total', q4\_merge['CancelNum'] + q4\_merge['NoCancelNum'])  q4\_merge = q4\_merge.withColumn('CancelRate', bround(q4\_merge['CancelNum']/q4\_merge['Total'], scale=6)) #计算取消率  q4\_merge.show()#展示结果    def reverseCapture():  q4\_merge=rateOfDelay()  pandas\_delay=q4\_merge.toPandas()  pandas\_delay.DelayRate.plot(kind-'barh')  plt.xlabel('FlightType')  plt.ylabel('CancelRate')  plt.title("CancelRate")  plt.show()# |

1. 运行界面



**九、项目结果与分析（含重要数据结果分析或核心代码流程分析）**

**1. 查看飞机延误时间最长的前10名航班。**

**2. 计算延误的和没有延误的航空公司的比例。**

**3. 分析****一天中延误最严重的飞行时间。**

**4. 分析一周中延误最严重的飞行时间。**

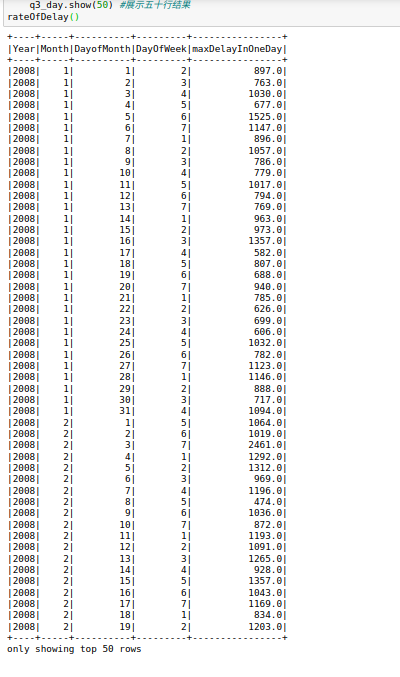
**5. 短途航班和长途航班，哪种航班取消更严重？**

**6. 建立机器学习算法模型，预测未来航班取消情况。**

**3. 分析一天中延误最严重的飞行时间。**

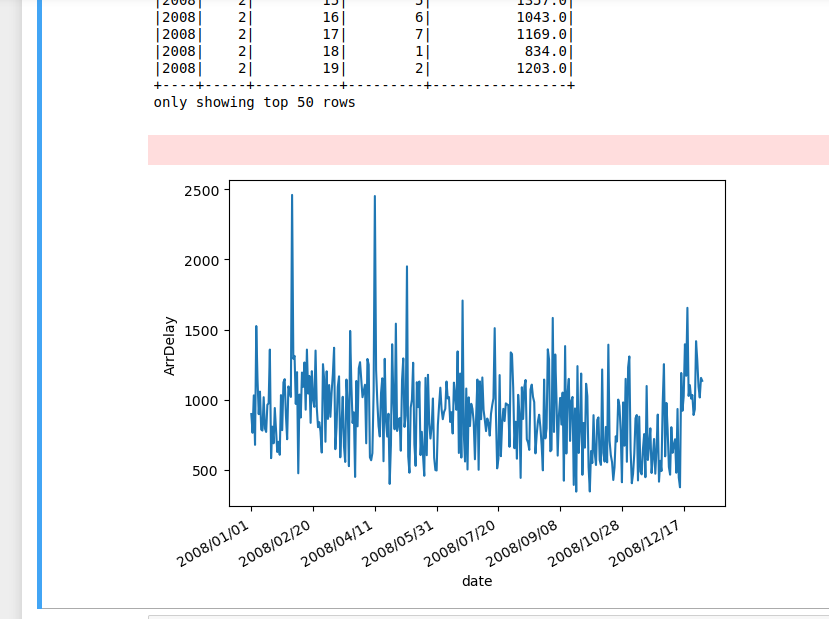
（1）重要数据结果分析

以下列表展示一年中每天最高的抵达延误时间，第一至四列分别为年、月、月中的第几日、星期几，第五列为该天最高的抵达延误时间。由于篇幅有限，此处仅展示部分结果（前50行）。



（2）折线图表示

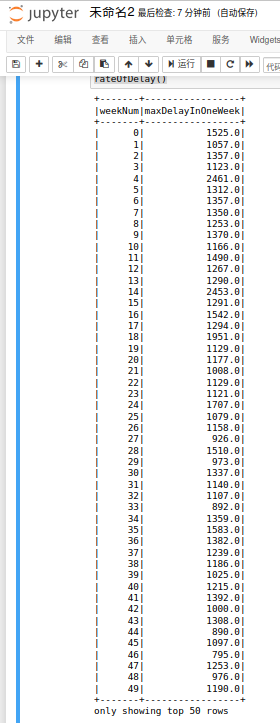
我们将上面的结果转换为可视化表格，调用toPandas()函数将pyspark下的dataframe结构转化为pandas下的dataframe，然后调用plot（）绘制折线图。该折线图用于每日延误的峰值，y轴代表的是延误时间，x轴代表的是日期，图形如下图所示：



**4. 分析一周中延误最严重的飞行时间。**

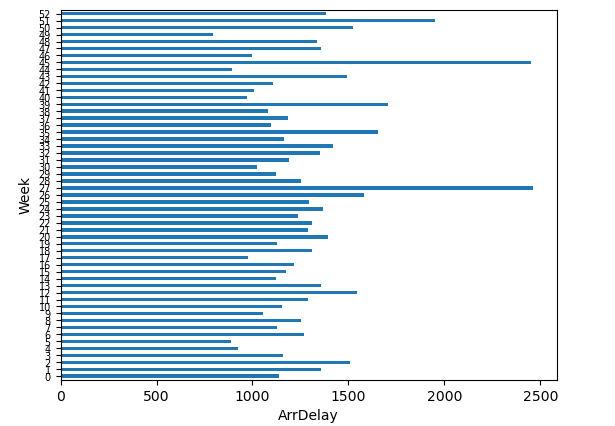
（1）重要数据结果分析

以下列表展示2008年中每一周中的最高抵达延误时间，第一列为周数，第二列为一周中的最高抵达延误时间。由于篇幅有限，此处仅展示部分结果（前50行）。



（2）柱状图表示

我们将上面的结果转换为可视化表格，调用toPandas()函数将pyspark下的dataframe结构转化为pandas下的dataframe，然后调用plot（kind='barh'）绘制柱状图。柱状图描述2008年每周航班延误峰值，y轴代表的是周数，x轴代表的是每周统计延误时间。柱状图如下图所示：



# 十一、总结及心得体会

# 十二、对本项目过程及方法、手段的改进建议