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： 罗瑜 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**：

Windows 10

适用于Windows的Linux子系统(WSL)

**7.2 开发工具**：

Python 3.12

PySpark

Anaconda3

JupyterNotebook

**7.3 配置过程**

**(1)安装Anaconda**

在终端使用wget命令

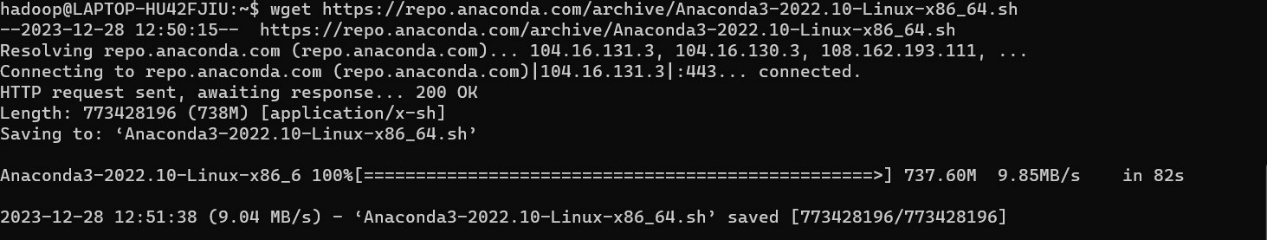
|  |
| --- |
| $ wget https://repo.anaconda.com/archive/Anaconda3-2022.10-Linux-x86\_64.sh |

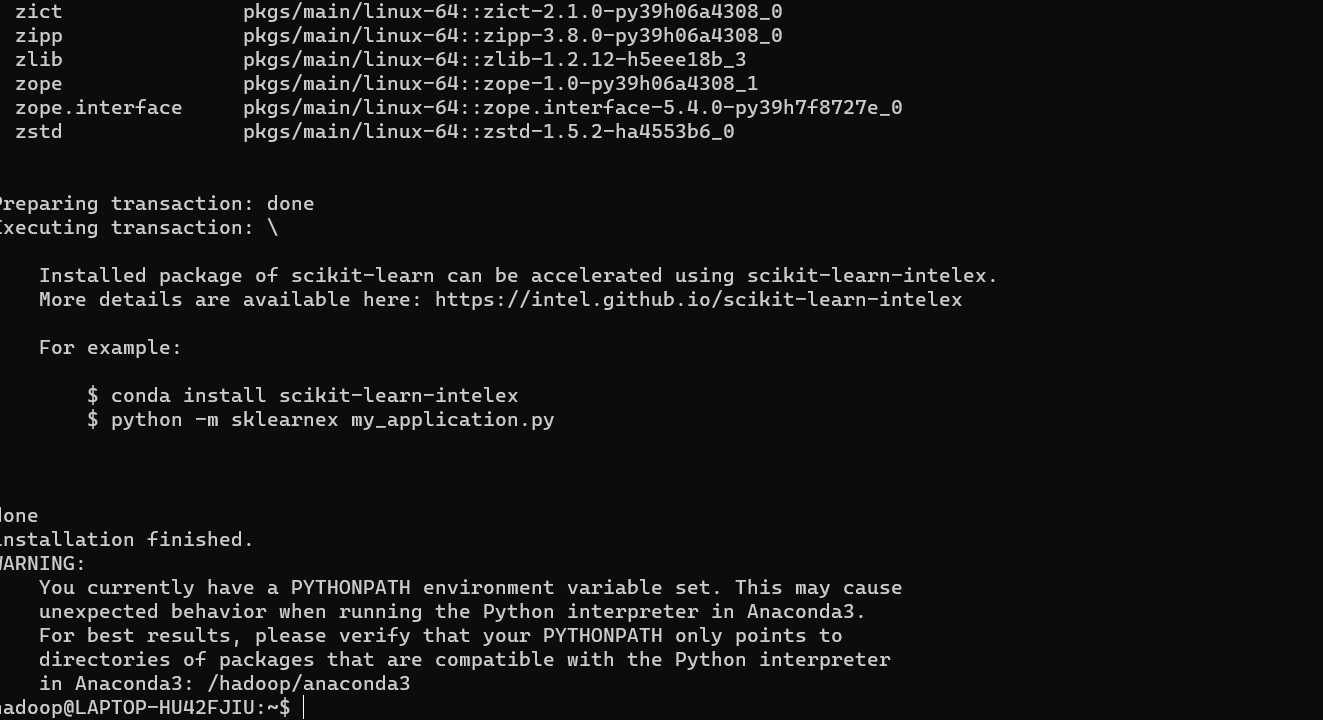
下载完成后，进行安装

|  |
| --- |
| $ bash Anaconda3-2022.10-Linux-x86\_64.sh -b |

-b是批次安装，会忽略Licese条款。

下载过程与安装过程如下图：

****

****

**(2)安装并配置Jupyter**

Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。简而言之，Jupyter Notebook是以网页的形式打开，可以在网页页面中直接编写代码和运行代码，代码的运行结果也会直接在代码块下显示。

配置PySpark driver

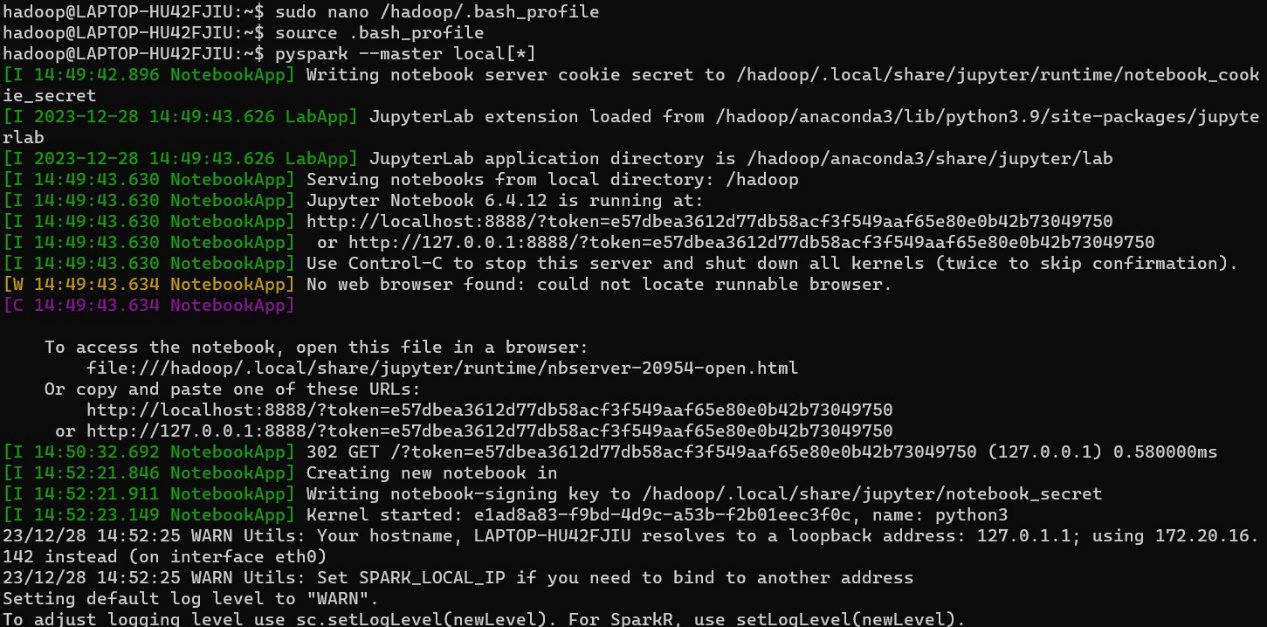
$ sudo nano /hadoop/.bash\_profile

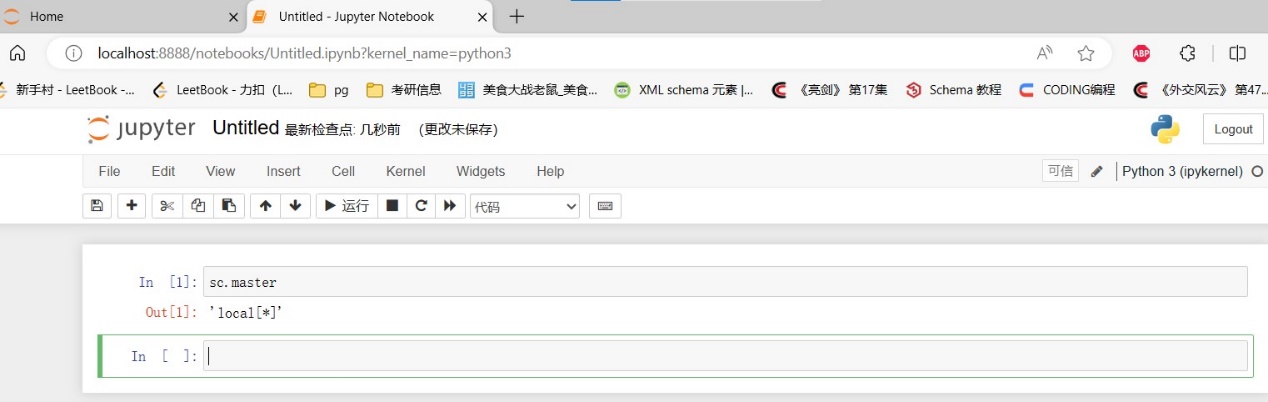
|  |
| --- |
| export PATH=$PATH:$SPARK\_HOME/bin:/hadoop/anaconda3/bin  export ANACONDA\_PATH=/hadoop/anaconda3  export PYSPARK\_DRIVER\_PYTHON=$ANACONDA\_PATH/bin/ipython  export PYSPARK\_PYTHON=$ANACONDA\_PATH/bin/python  export PYSPARK\_DRIVER\_PYTHON=jupyter  export PYSPARK\_DRIVER\_PYTHON\_OPTS='notebook' |

保存关闭后，执行以下命令使得环境变量生效：

$ source /hadoop/.bash\_profile

重新输入pyspark后，在浏览器中就可以使用jupyter了。效果如图：





# 八、项目实践过程

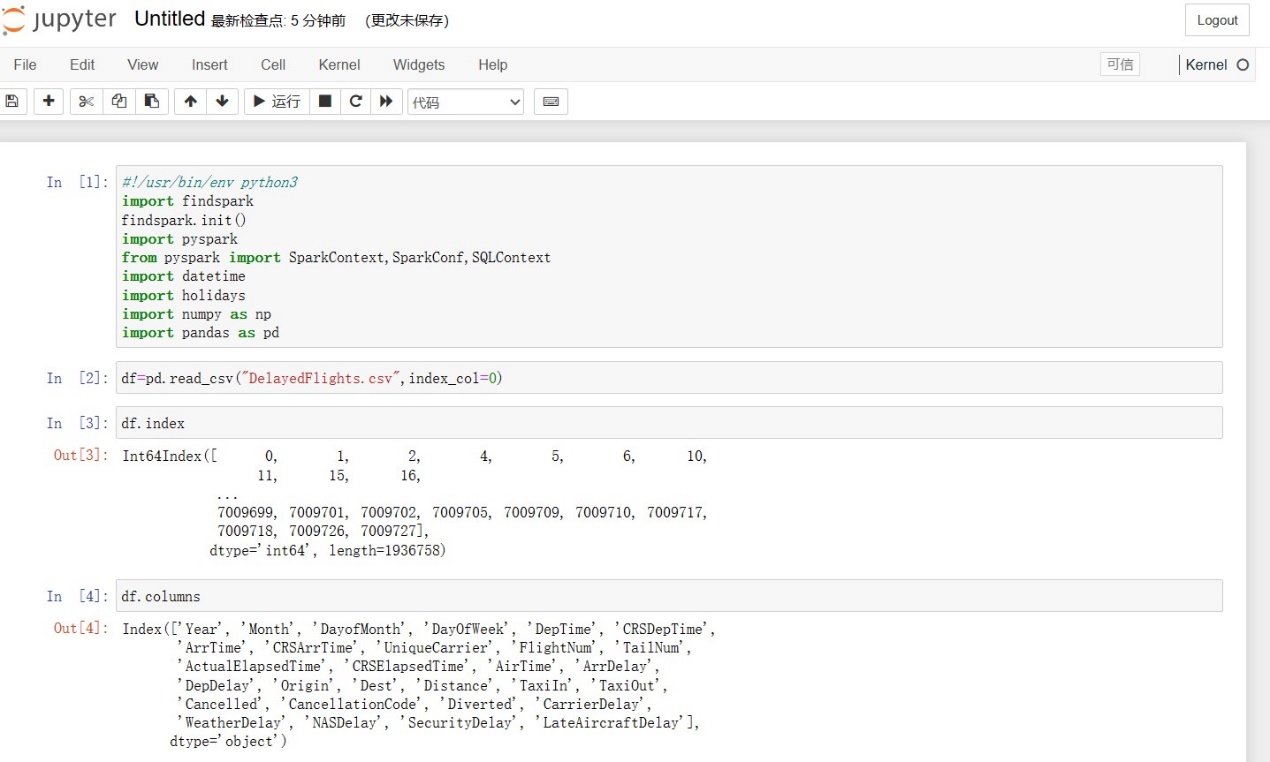
**8.1 读取数据**

从数据集中读取数据并保存到pandas库提供的dataframe对象中。代码如下所示

代码1 读取数据

|  |
| --- |
| # 导入包  #!/usr/bin/env python3  import findspark  findspark.init()  import pyspark  from pyspark import SparkContext,SparkConf,SQLContext  import datetime  import holidays  import numpy as np  import pandas as pd  # 读取csv格式数据集到变量df  df=pd.read\_csv("DelayedFlights.csv",index\_col=0)  # 输出df的行名与列名，以验证是否读取正确  df.index  df.columns |

结果如图：



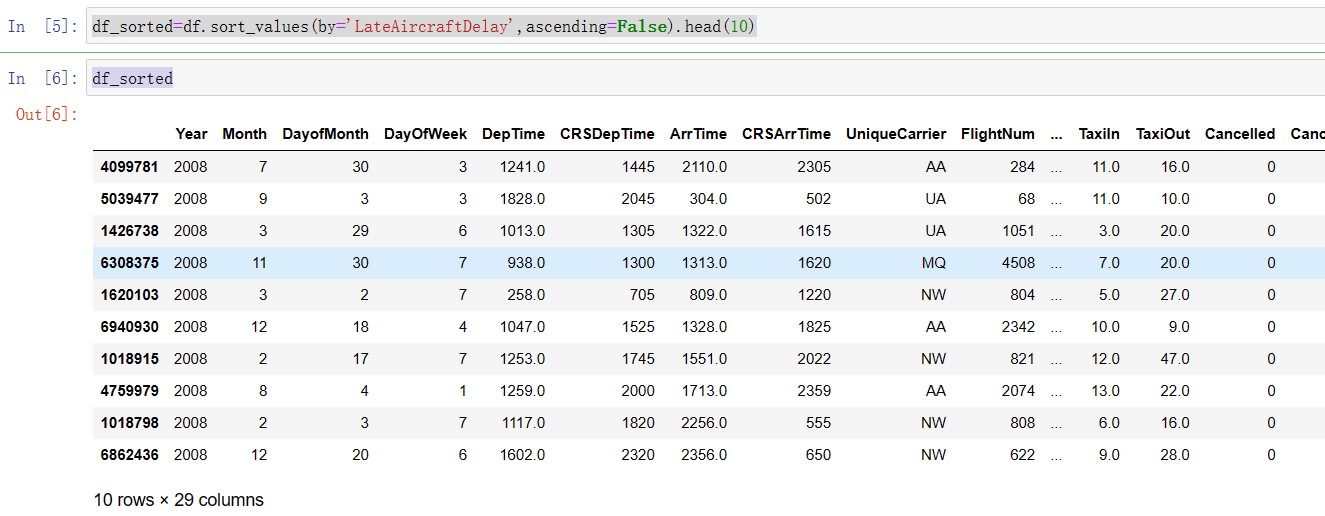
**8.2 完成目标一：查看飞机延误时间最长的前10名航班**

飞机延误时间即'LateAircraftDelay'字段。将数据集按照该字段降序排序并取前10条输出。代码如下所示

代码2 查看飞机延误时间最长的前10名航班

|  |
| --- |
| df\_sorted=df.sort\_values(by='LateAircraftDelay',ascending=False).head(10)  df\_sorted |

结果如图：



**8.3 完成目标二：计算延误的和没有延误的航空公司的比例**

统计出数据集中各个航空公司总的数据条数，以及'LateAircraftDelay'字段等于0的数据条数，即可计算延误的和没有延误的航空公司的比例。代码如下所示

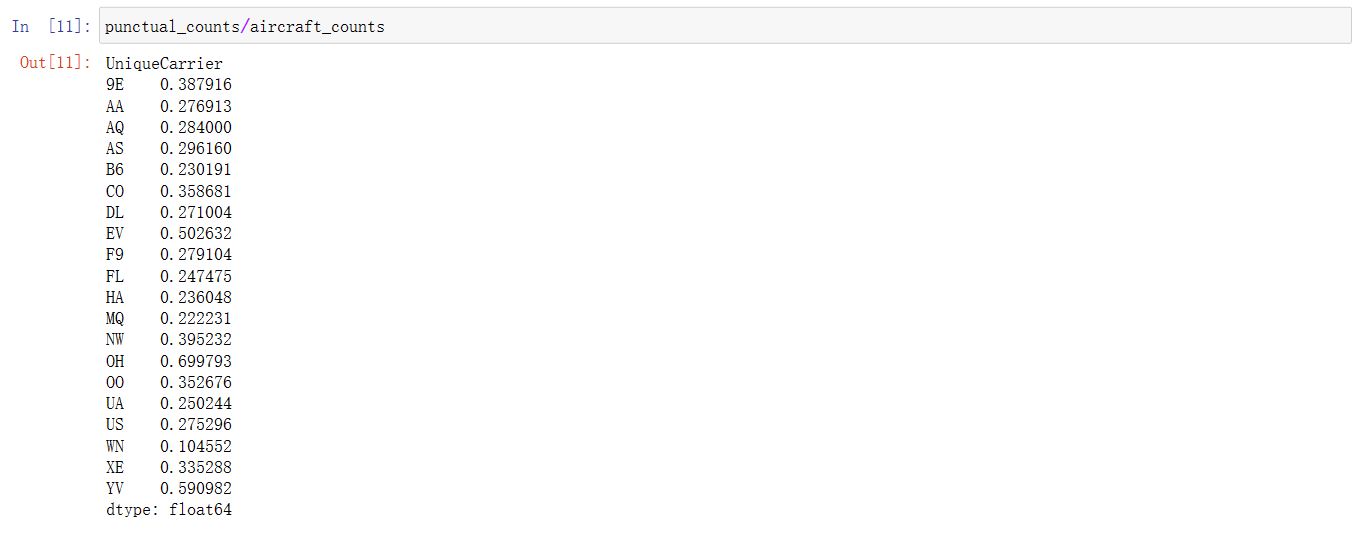
代码3 计算延误的和没有延误的航空公司的比例

|  |
| --- |
| aircraft\_counts = df.groupby('UniqueCarrier').size()  aircraft\_counts  punctual\_counts = df[df.LateAircraftDelay == 0].groupby('UniqueCarrier').size()  punctual\_counts  punctual\_counts/ aircraft\_counts  delay\_counts = df[df.LateAircraftDelay != 0].groupby('UniqueCarrier').size()  delay\_counts  delay\_counts/ aircraft\_counts |

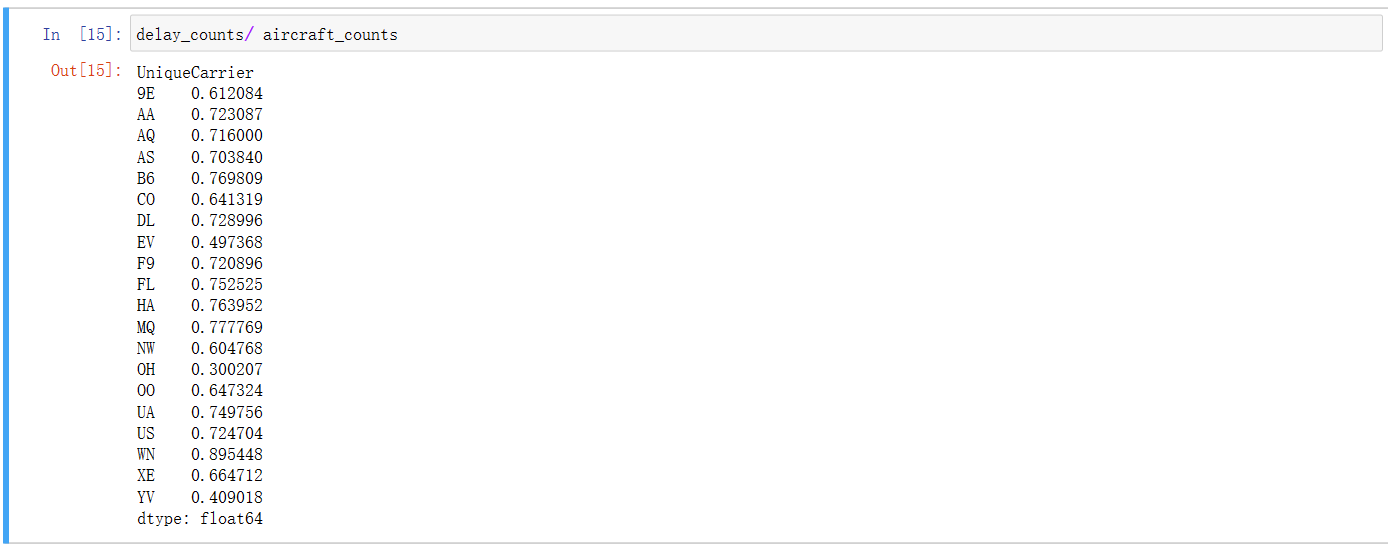
结果如图：











**8.4 完成目标三：分析一天中延误最严重的飞行时间**

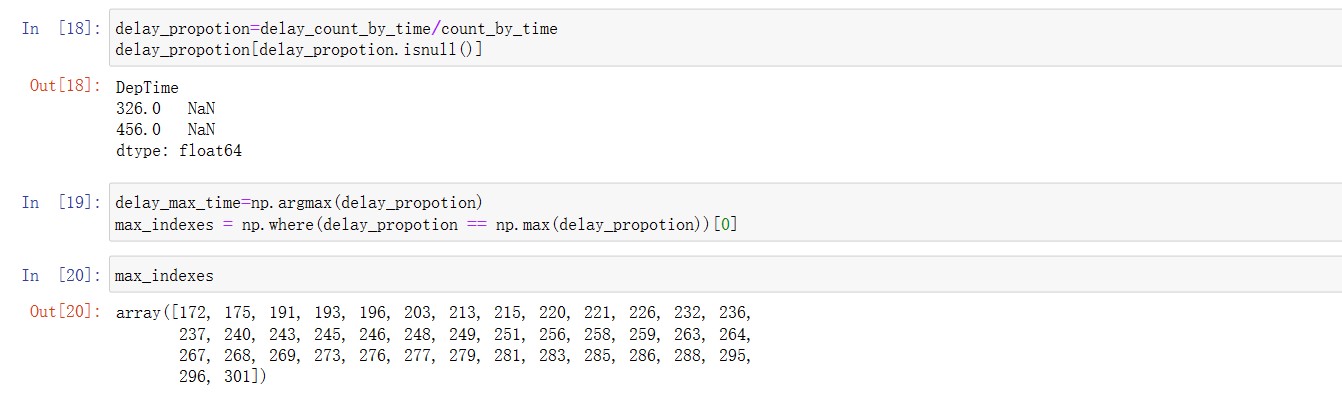
目标三的涵义是：从一天的各个时间点当中延误的航班数占航班总数的占比考虑，占比最大的时间为延误最严重的时间。代码如下所示

代码4 分析一天中延误最严重的飞行时间

|  |
| --- |
| count\_by\_time=df.groupby('DepTime').size()  delay\_count\_by\_time= df[df.LateAircraftDelay != 0].groupby('DepTime').size()  count\_by\_time  delay\_count\_by\_time  delay\_propotion=delay\_count\_by\_time/count\_by\_time  delay\_propotion[delay\_propotion.isnull()]  delay\_max\_time=np.argmax(delay\_propotion)  max\_indexes = np.where(delay\_propotion == np.max(delay\_propotion))[0]  max\_indexes |

结果如图：(注意，max\_indexes中的各项+1表示其对应的索引是一天的第多少分钟)





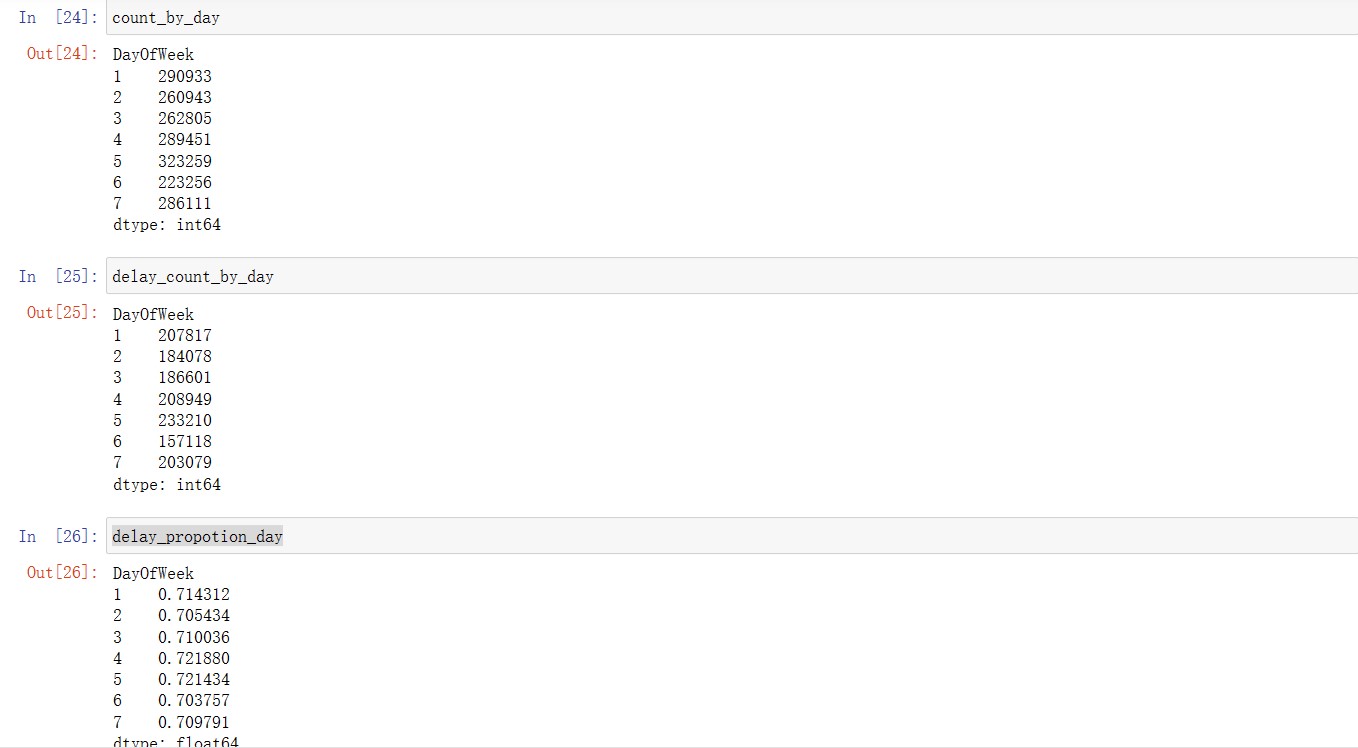
**8.5 完成目标四：分析一周中延误最严重的飞行时间**

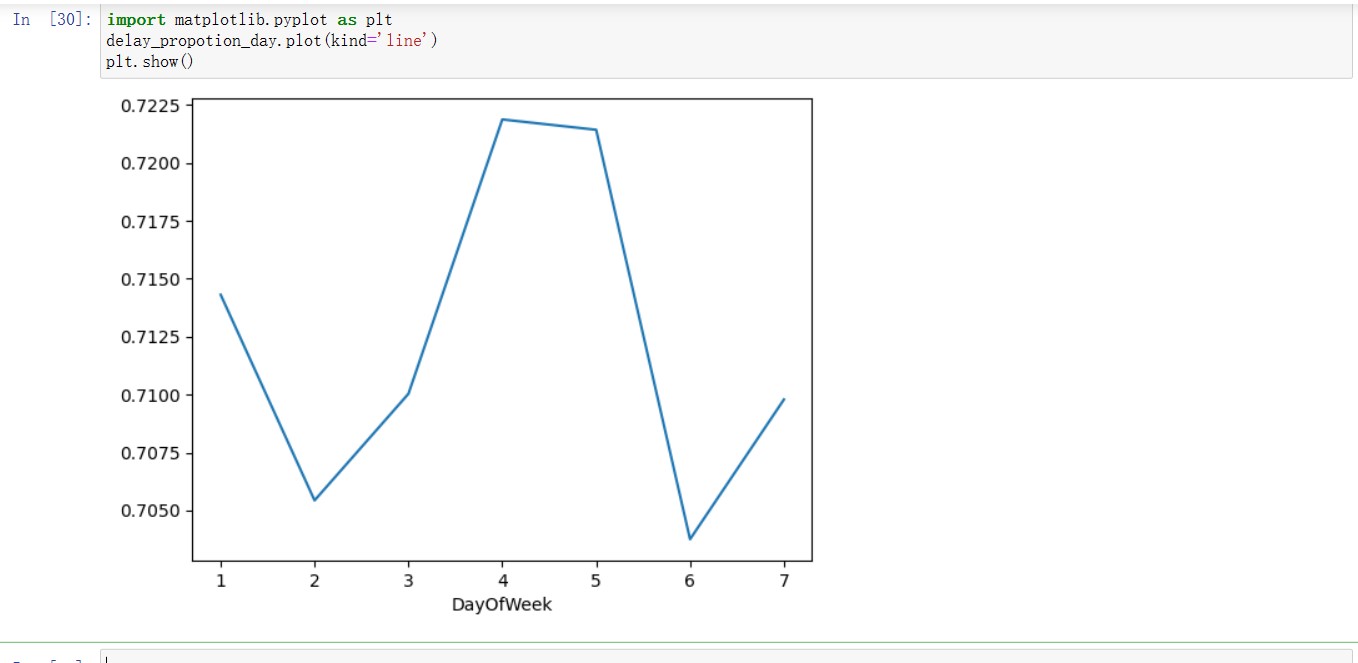
目标四与目标三涵义类似，从一周的各日当中延误的航班数占航班总数的占比考虑，占比最大的一日为延误最严重的一日。代码如下所示

代码5 分析一周中延误最严重的飞行时间

|  |
| --- |
| count\_by\_day=df.groupby('DayOfWeek').size()  delay\_count\_by\_day= df[df.LateAircraftDelay != 0].groupby('DayOfWeek').size()  count\_by\_day  delay\_count\_by\_day  delay\_propotion\_day=delay\_count\_by\_day/count\_by\_day  count\_by\_day  delay\_count\_by\_day  delay\_propotion\_day  import matplotlib.pyplot as plt  delay\_propotion\_day.plot(kind='line')  plt.show() |

结果如图：





**8.6 完成目标五：短途航班和长途航班，哪种航班取消更严重？**

根据中国《通用航空短途运输运营服务管理办法》的规定，短途航班指省内飞行的航班或者跨省飞行、航程500公里(310.6英里)以内的航班。为方便起见，本实验不考虑跨省与否，只以航程Distance字段与310.6英里的比较来判断其属于短途或者长途。

目标五的涵义就是分别比较短途航班与长途航班中Cancelled字段为1的航班占对应种类航班总数的比例，其统计与前文的目标三、四类似。代码如下所示

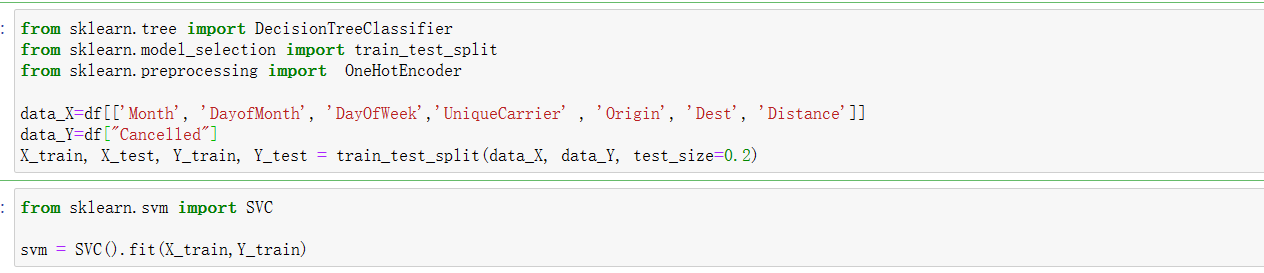
代码6 分析短途航班和长途航班哪种航班取消更严重

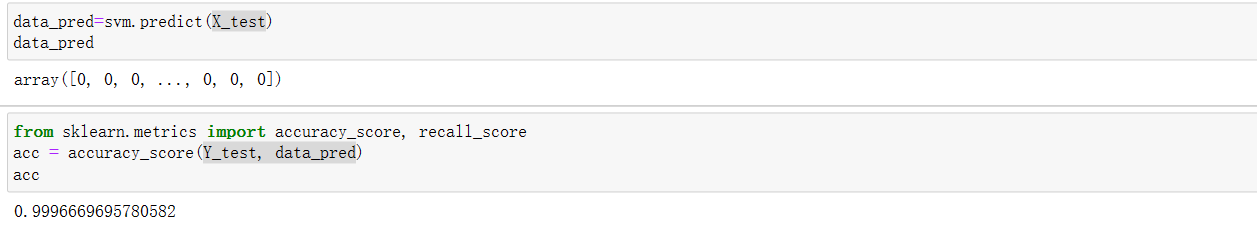
|  |
| --- |
| df\_long = df[df.Distance >= 310.6]  df\_short = df[df.Distance < 310.6]  count\_long= len(df\_long)  count\_cancel\_long= df\_long.groupby('Cancelled').size()  count\_short= len(df\_short)  count\_cancel\_short= df\_short.groupby('Cancelled').size()  long\_cancel\_propotion= count\_cancel\_long[1]/ count\_long  short\_cancel\_propotion= count\_cancel\_short[1]/ count\_short  df\_paint= pd.DataFrame({"long\_cancel\_propotion":[long\_cancel\_propotion], "short\_cancel\_propotion":[short\_cancel\_propotion]})  df\_paint.plot(kind="bar")  plt.show() |

结果如图



**8.7 完成目标六：建立机器学习算法模型，预测未来航班取消情况**





**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。**

# 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