**基于美国交通事故数据的分析、建模与可视化**

**摘要**：本文对美国2016-2020年交通事故数据集进行探索性数据分析，以EDA、数值特征处理等特征工程不同处理方式对特征属性进行分析。针对预测未来发生的事故数分别基于Prophet与多层神经网络实现并进行模型效果的对比以及构建随机森林模型来区分事故是否严重。

1. 数据集与数据探索
   1. 数据集介绍

美国交通事故数据集包含1个文件，US\_Accidents\_Dec20\_Updated.csv。

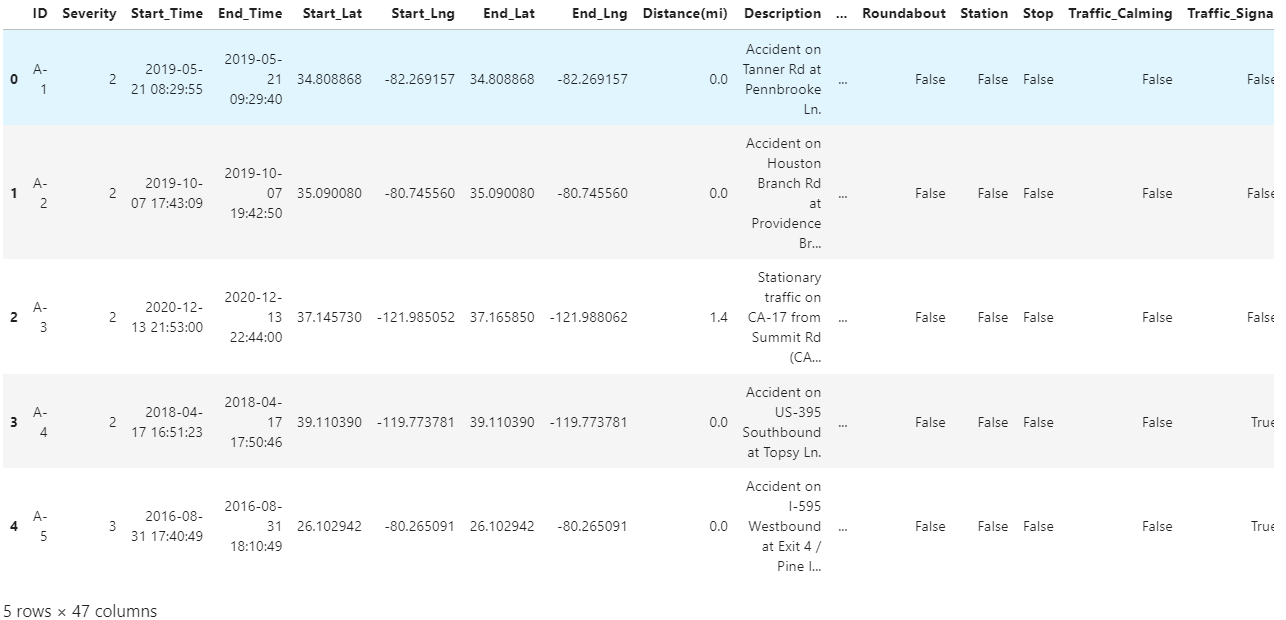


图 1.1：美国2016-2020年交通事故数据集US\_Accidents\_Dec20\_Updated.csv

* 1. 数据探索

先对数据的全貌有一个大致了解，可视化2016年到2020年每年各州发生的交通事故总数，从直观上了解哪些州发生的事故数量较多并且知道对应州的位置以及严重性较高的事故一般分布在哪。

1.2.1 用pyecharts库

首先数据预处理，将原始数据变成相应的字典模式

1. 选取所需要的列，进行缺失值处理

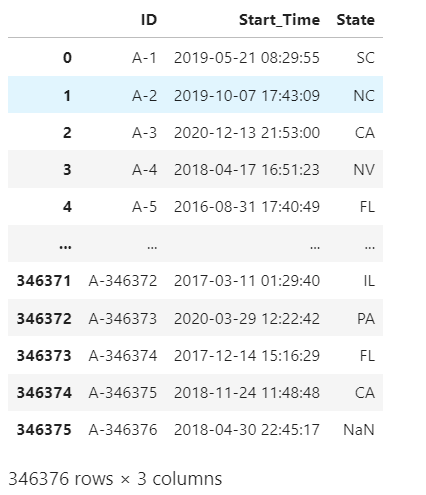


图 1.2

1. 经过groupby得到不同年份不同州发生的交通事故总数

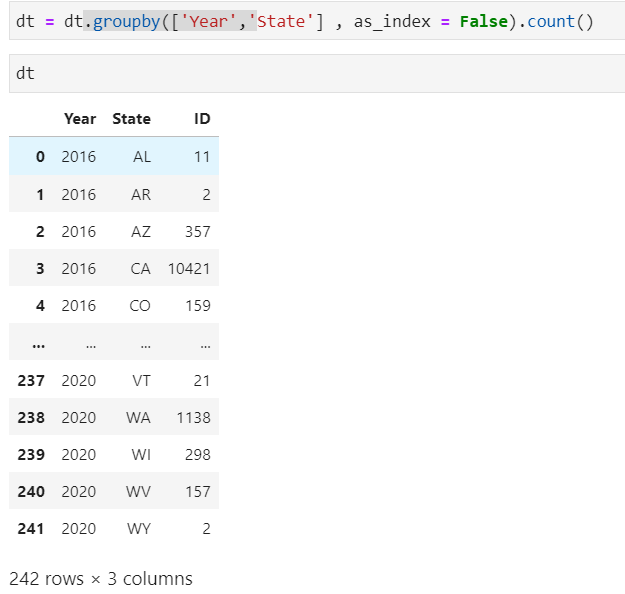


图 1.3

1. 将表变成能够输入的字典模式，将State这一列中的所有州的名称全名，得到map\_data，如这是2016年的数据。



图 1.4

1. 绘制时间轮播图

基于位置信息的可视化，将美国各个位置所发生的事故数进行累加，得到全美各州事故的分布情况，颜色越接近红色，代表事故数量越多，可以清楚的看到，这几年California、Texas、Florida的交通事故数量都比较多，它们都分布在沿海，美国内陆发生的交通事故较少。

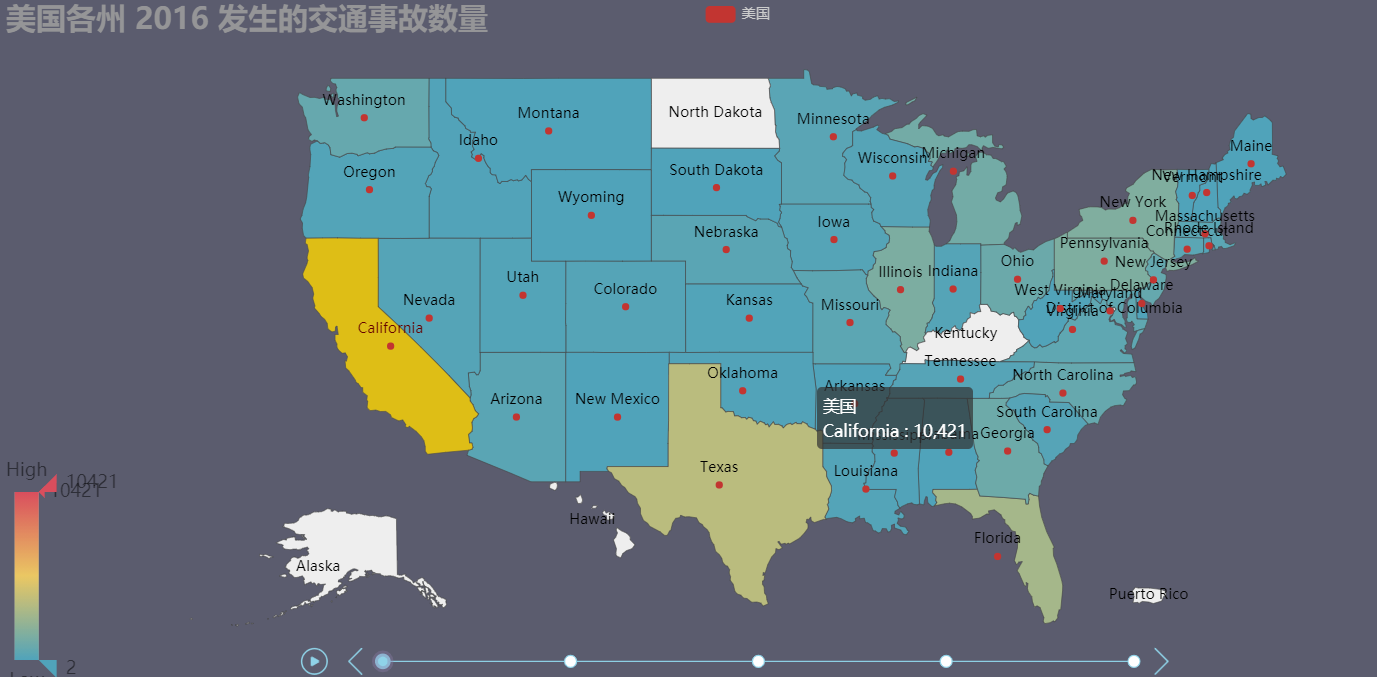


图 1.5 交互图截图

再看柱状图，同样是颜色越接近红色，代表事故数量越多，非常清楚看到California、Texas、Florida的交通事故数量甚至是某些州的数倍，并且从2016-2019年可以看到交通事故数量都有增加的趋势，2019年数量下降，2020年又增加，2019年12月爆发了新冠疫情，数据的变化可能同新冠疫情相关。



图 1.6 交互图截图

最后将柱状图、地图和饼图绘制在一张图中，看到饼图可以发现这几年基本上California、Texas、Florida总共就差不多占了50%。

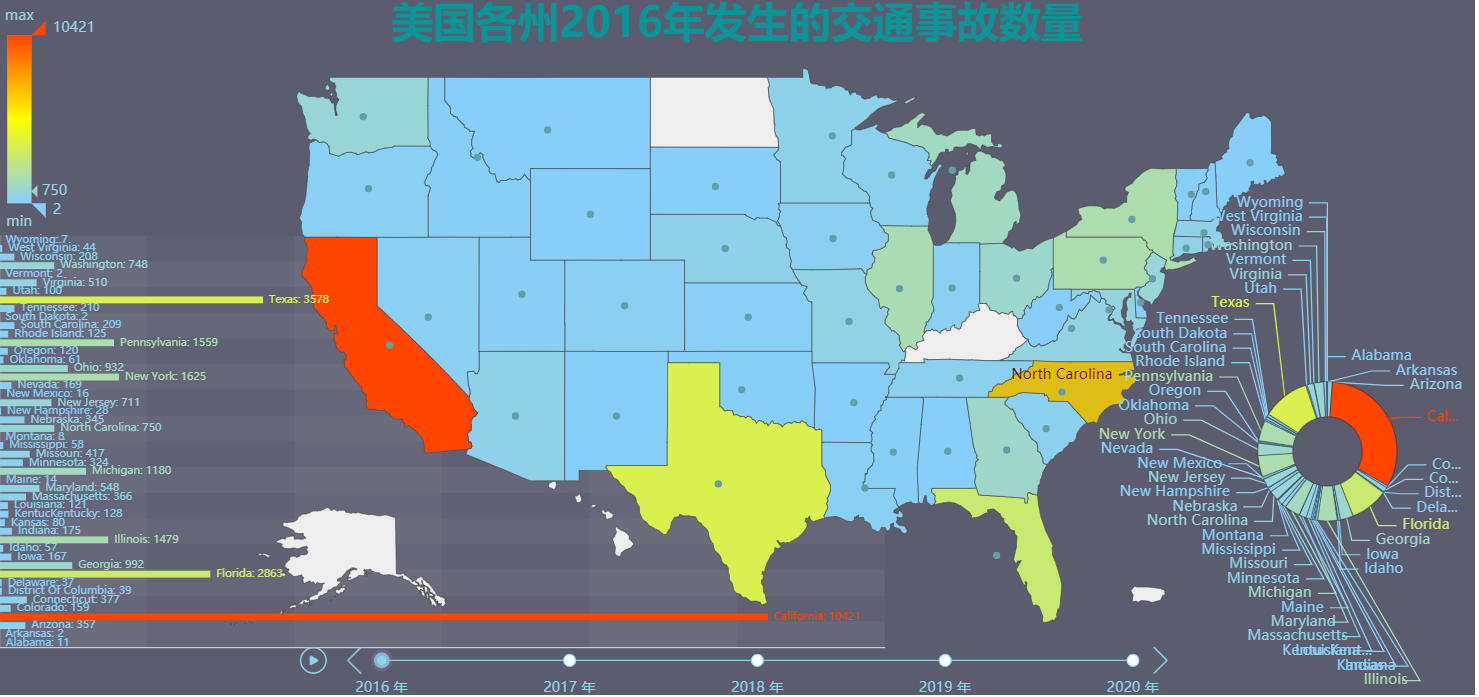


图 1.7交互图截图

1.2.2用folium 库来可视化

上面用pyecharts需要用一个类似于映射表来将数据与图对应，而folium直接可以通过经纬度确定事故发生地点。

1. 数据预处理，获取需要的列，同时进行缺失值处理

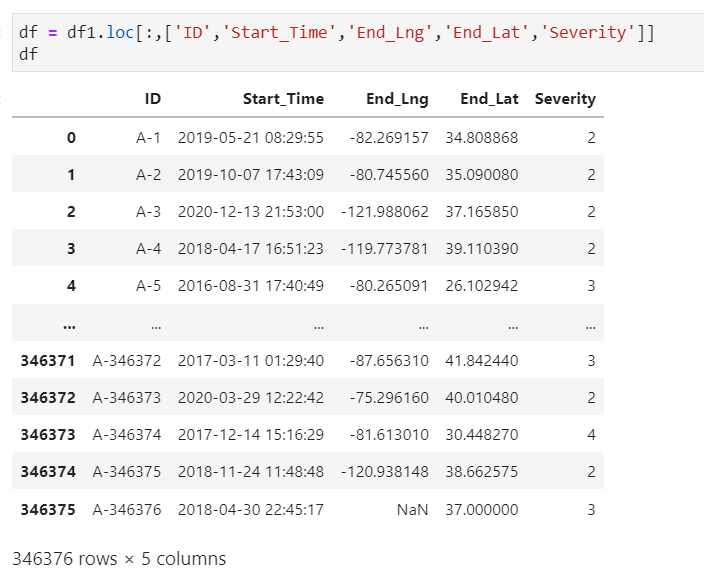


图 1.8

1. 绘图：

泡泡图：如果重叠成一团，就说明事故发生非常密集，通过点的聚集程度可以发现事故一般都是发生在西部沿海以及东部地区。

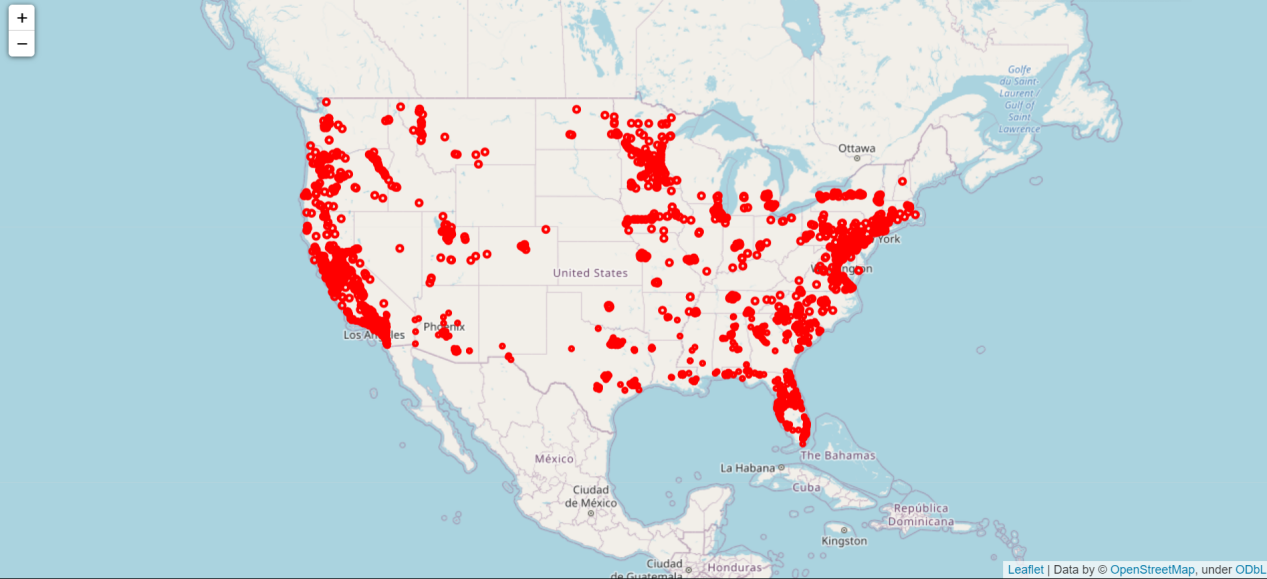


图 1.9交互图截图

热力图：

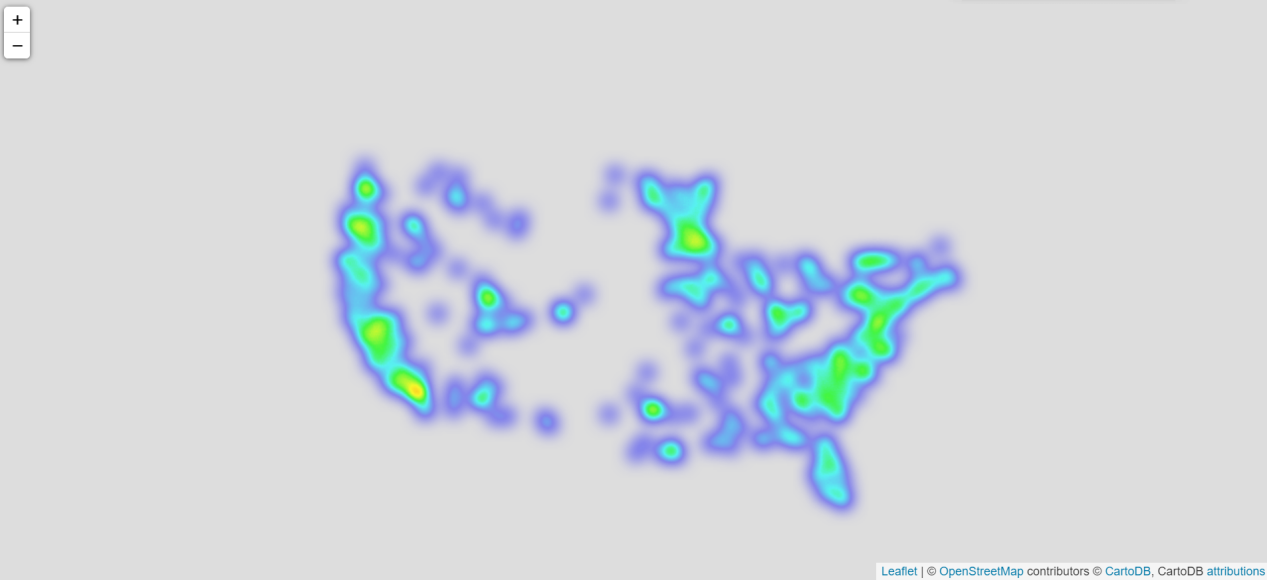


图 1.10交互图截图

严重性较高的事故的分布情况，大小代表严重性，越大代表严重性越高，这也说明越容易发生事故的地方其严重性也相对更高。

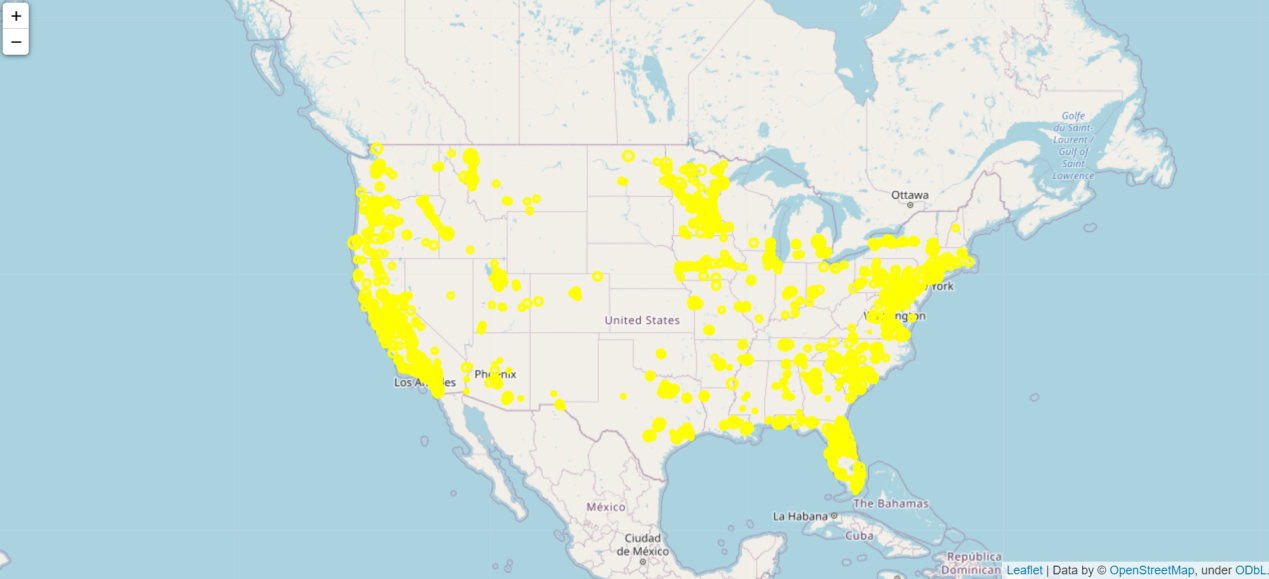


图 1.11交互图截图

1.2.3 利用折线图、箱型图、热力图、小提琴图进行可视化

首先数据预处理，以日期为索引，得到每天全美国发生的事故总数。

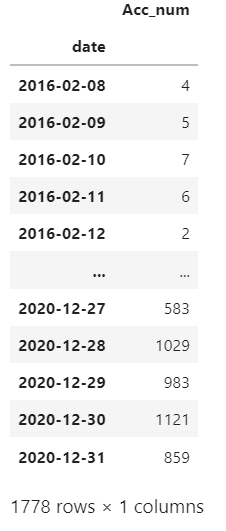


图 1.12

绘制折线图散点图，

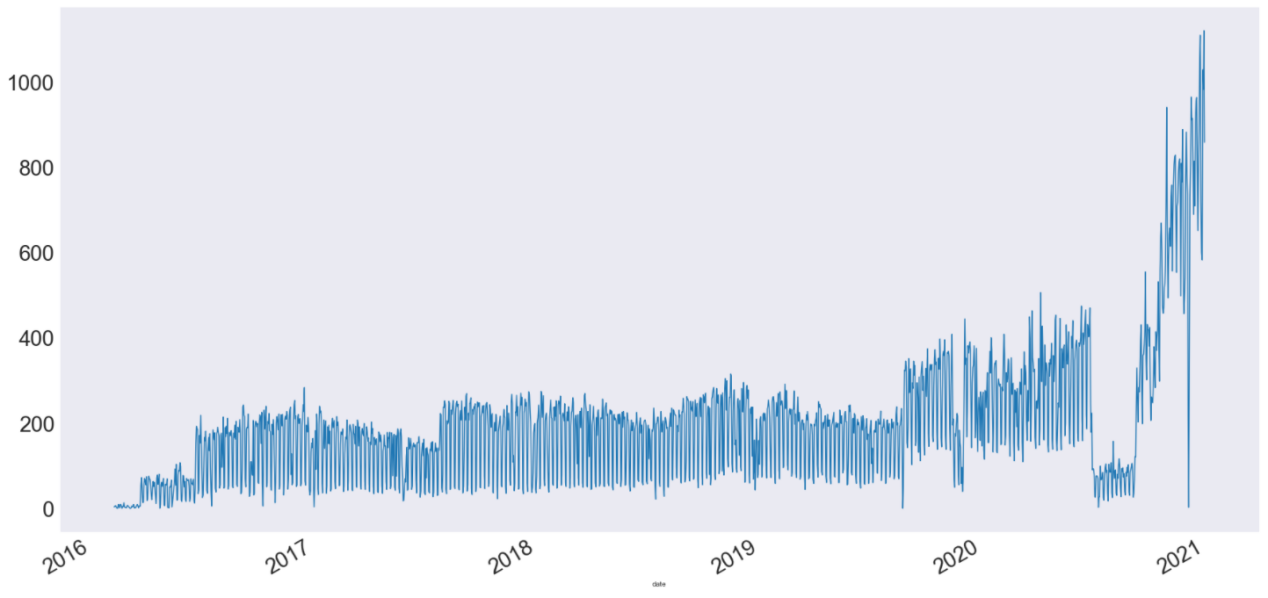


图 1.13

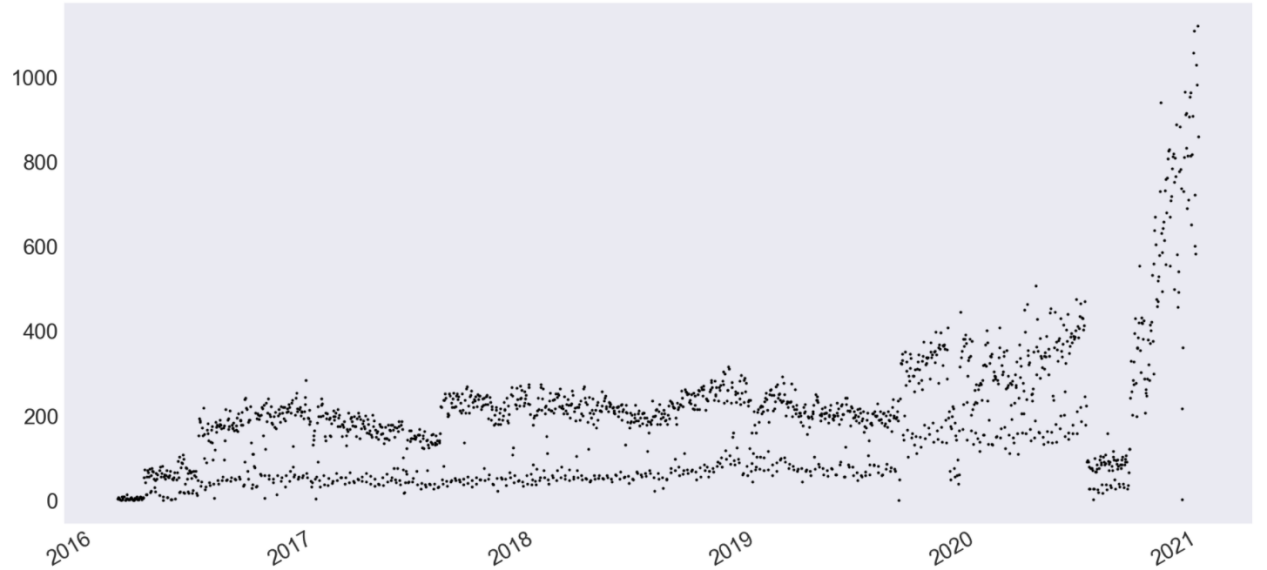


图 1.14

发现由于自2020年疫情的影响，2020-2021年间发生的交通事故数量明显增加，所以先研究2016-2020年间，在不受疫情影响的情况下探寻一般情况下的规律。

再绘制直方图，堆积面积图，概率密度分布图，看看总体的规律

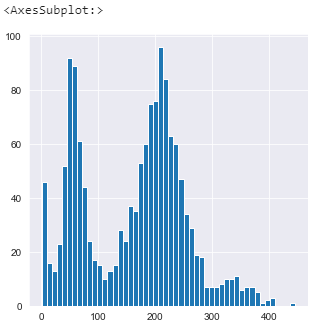
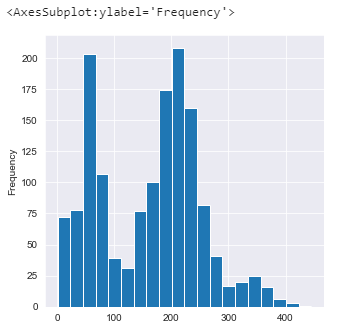


图 1.15

bins分的格数越多，划分得越精确，锯齿现象越明显

堆积面积图

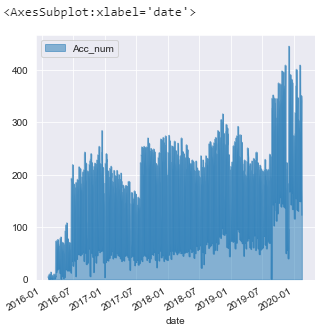


图 1.16

概率密度分布图

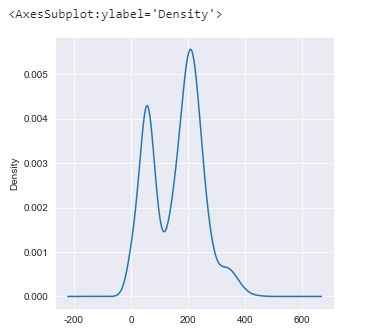


图 1.17

绘制每年的变化折线图、箱型图、热力图

先将数据变成如下矩阵：



图 1.18

折线图

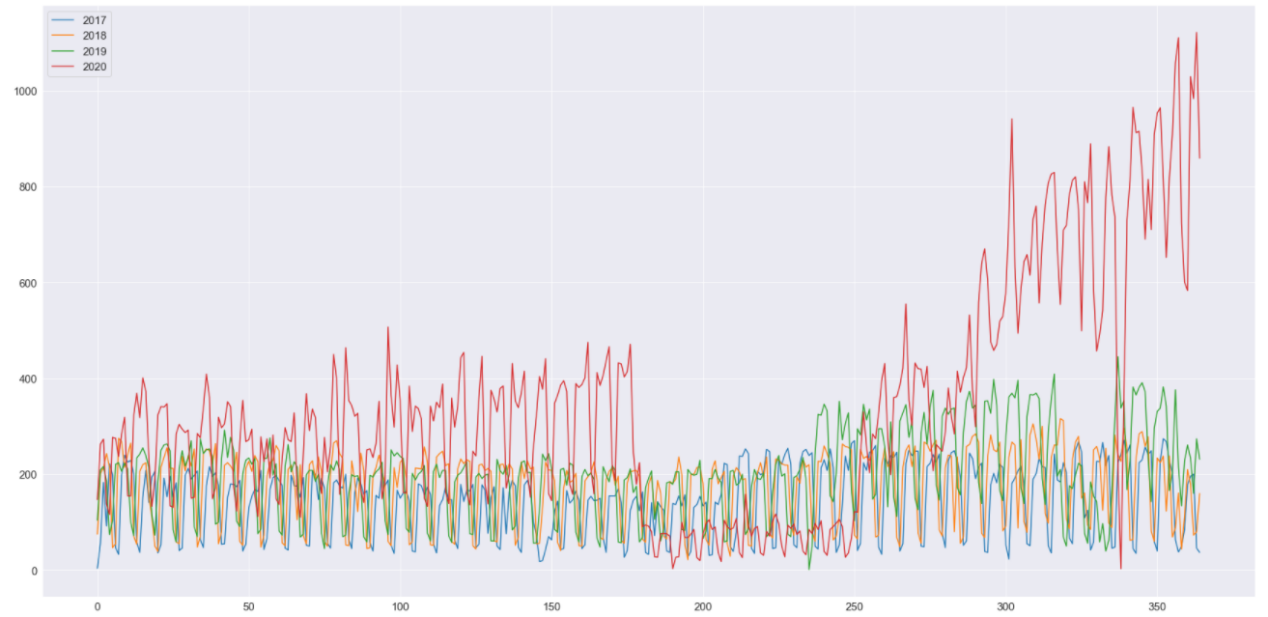


图 1.19

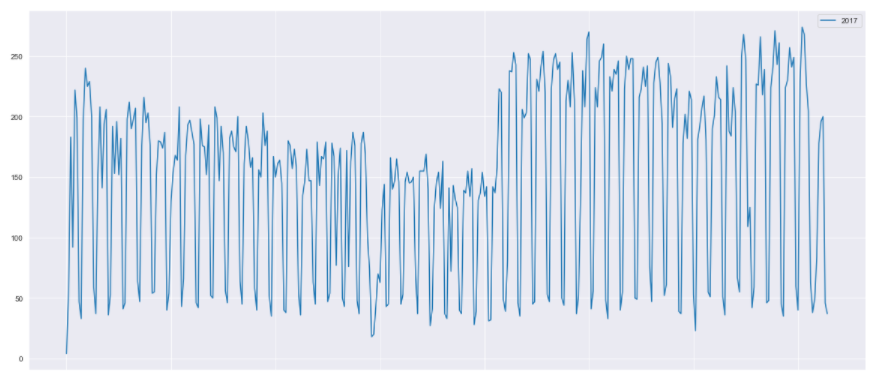


图 1.20

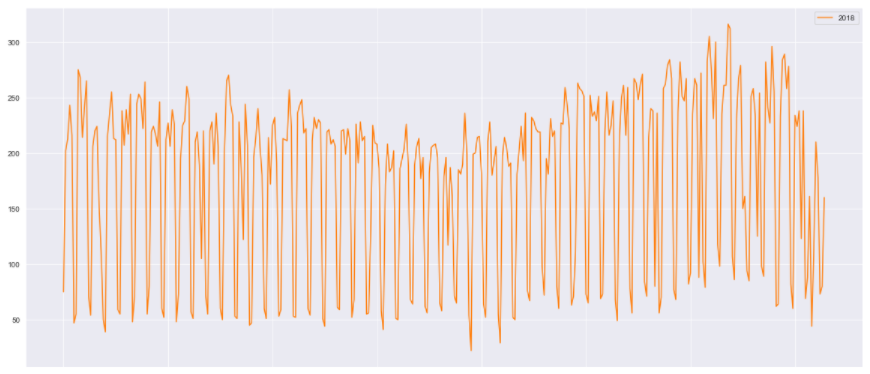


图 1.21

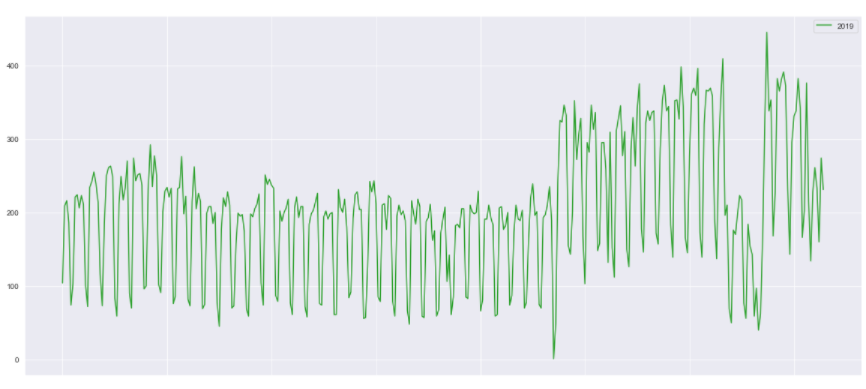


图 1.22

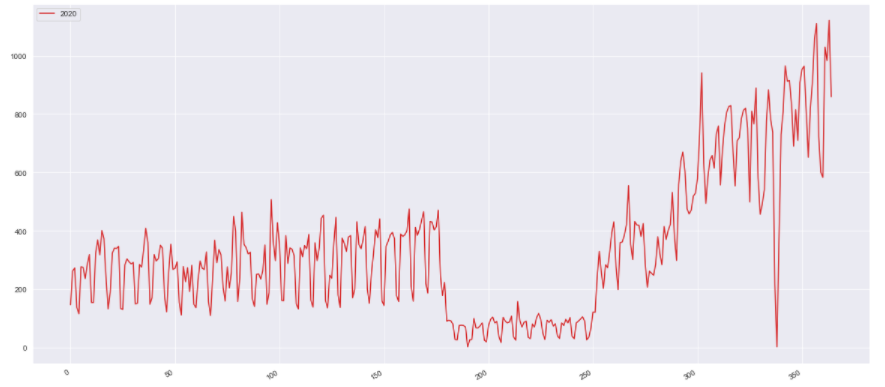


图 1.23

箱型图

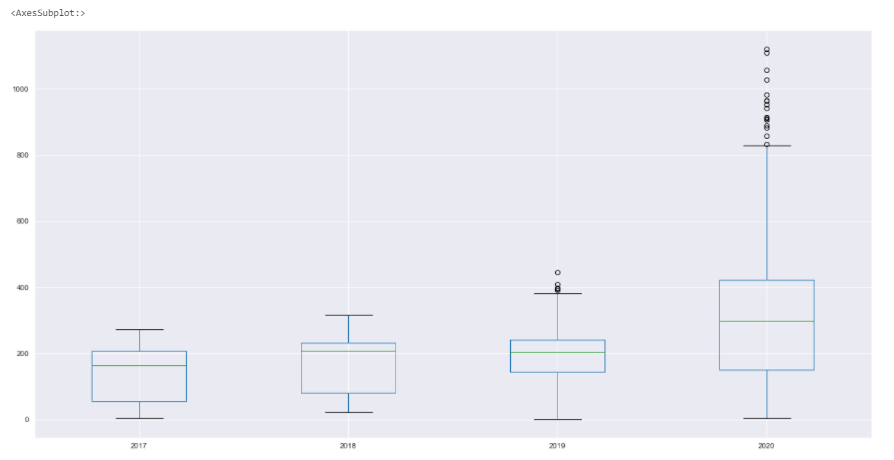


图 1.24

热力图

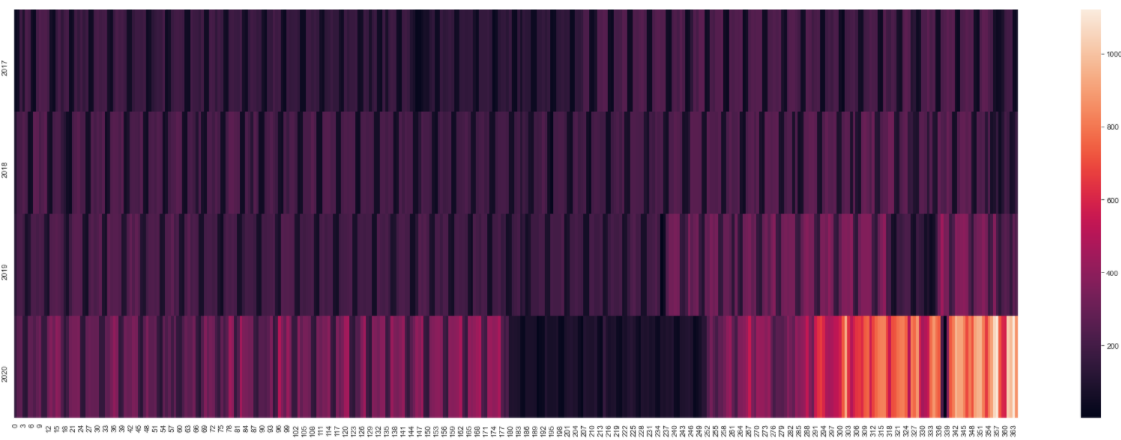


图 1.25

将前面获得的矩阵的各个值可视化出来，每一行是每一年，每一列是每一天，便可以看到1095个数据的全貌，颜色越深，代表发生的数量越少，是可视化方法中的表格显示。

1.3 数值型数据处理

1.3.1特征处理

1.预测未来发生的事故数

1.1Prophet模型需要进行的处理

US\_Accidents\_Dec20\_Updated.csv中的数据，是根据事故编号进行记录的，其记录的还有事故发生的时间，现在需要得到时间序列，以事故数量作为属性，故我们只需提取US\_Accidents\_Dec20\_Updated.csv中ID，Start\_Time对应的列即可，然后计算相应的时间戳得到date，去除不需要的列：Start\_Time，并根据date算得每一天发生的事故数，改变列的名称，得到：

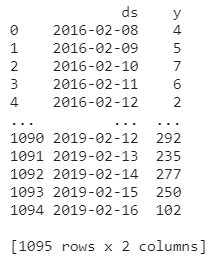


图 1.26：数据预处理过程图

1.2 神经网络模型需要进行的处理：

同1.1Prophet模型需要进行的处理类似，只不过不需要改变列的名称，只需要将 date变为索引得到：

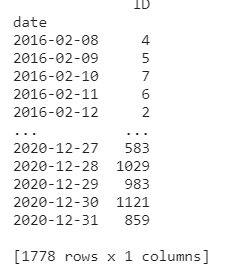


图 1.27：数据预处理过程图

2. 随机森林模型所需的预处理

2.1处理事故严重程度

severity=2以上的事故为严重，让其对应的target=1，之后利用get\_dummies方法对定类数据进行独热向量编码

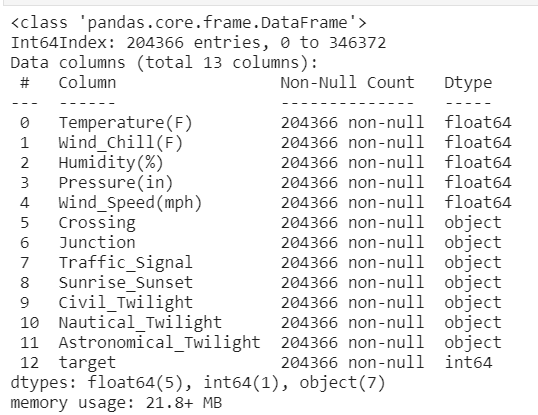


图 1.28：数据预处理过程图

而这个方法就是作用于Dtype为object的属性，例如crossing有两类，ture和false,独热向量编码就会把它处理成两类，crossing\_true 和crossing\_false

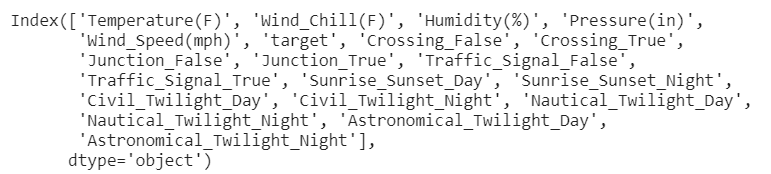


图 1.29：数据预处理过程图

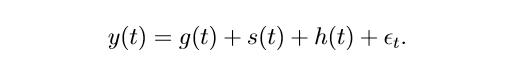
1. 数据建模

本实验需要针对美国2016-2020年交通事故数据集建立有效的预测模型，预测未来发生的事故数是一个时间序列预测问题，区分事故是否严重是一个二分类问题。本实验分别基于Prophet与多层神经网络进行数据建模，并进行模型效果的对比，采用随机森林模型解决二分类问题。

1.预测未来发生的事故数

**1.1 Prophet算法**

Prophet算法是一种基于加法模型预测时间序列数据的方法，采用时间序列趋势分解法的模型，得到三个部分：trend(趋势部分)、seasonality(周期性部分，可以包括daily、weekly、yearly)和holidays(假期影响因子)



其中，公式g(t)代表trend部分，s(t)代表seasonality部分，h(t)代表holidays部分，最后一项代表整个模型不适应的残差部分。Prophet算法也是分别对模型的三个部分进行建模，然后组合生成预测数据。

**1.2 多层神经网络模型**

第一层为32个神经元，激活函数为修正线性单元，第二层为64个神经元，激活函数是修正线性单元，最后输出层一个神经元。

2. 二分类问题

**2.1 随机森林**

随机森林是一种由决策树构成的集成算法。

三、实验结果和分析

3.1 预测未来发生的事故数

在本实验中，训练集为2016-02-08到2019-02-16对应的数据，测试集为2019-02-17到2020-02-16对应的数据。

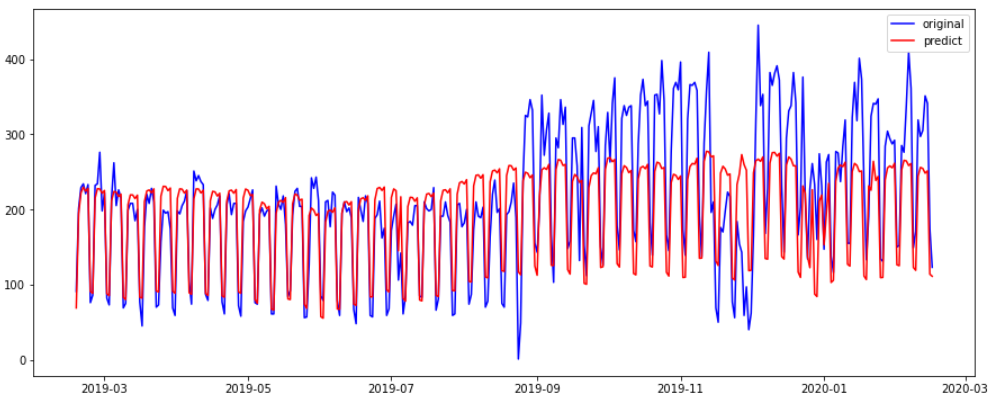


图 1.30：Prophet预测结果与原始值的拟合

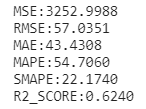


图 1.31：Prophet回归评价指标

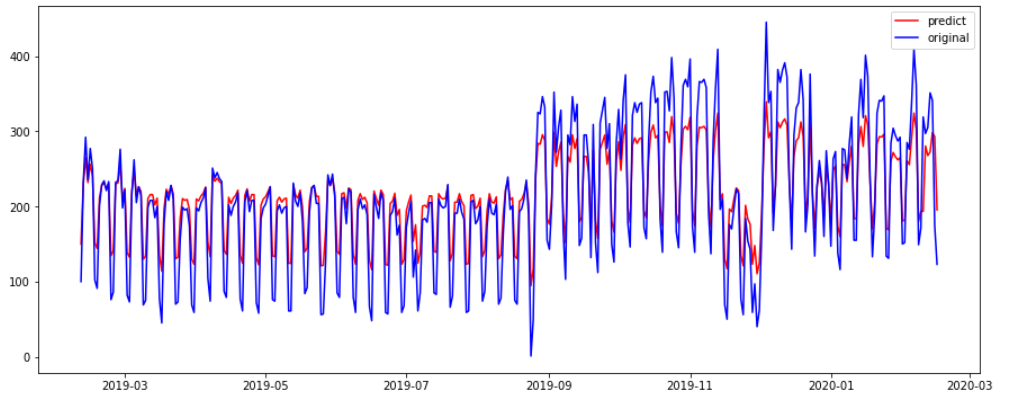


图 1.32： 多层神经网络预测结果与原始值的拟合

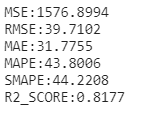


图 1.33： 多层神经网络回归评价指标

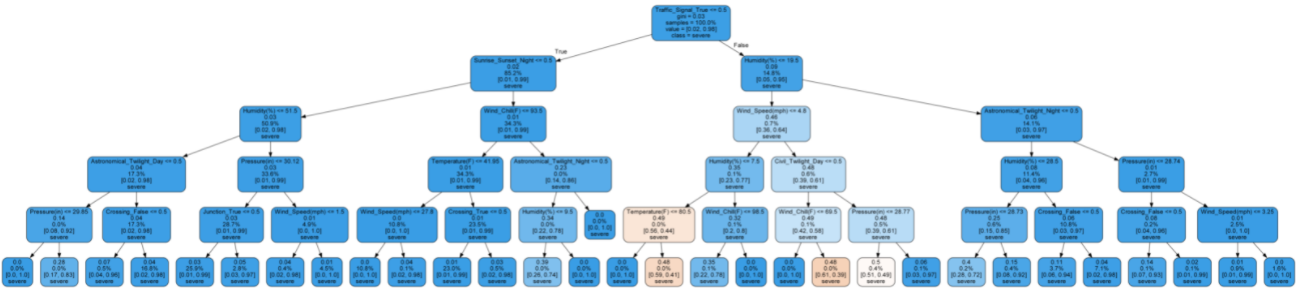
从MAPE来看，在本实验中Prophet的准确程度要优于多层神经网络，从其他指标来看，在本实验中多层神经网络的准确程度要优于Prophet。

那么在本实验中，根据选取评价指标的不同，可以选择不同的模型。

3.2 二分类问题

在本实验中，训练集为2016-02-08到2020-12-31之间80%的数据，测试集为2016-02-08到2020-12-31之间20%的数据。

可视化随机森林中的一颗决策树，属于层次数据可视化中的结点链接法



橙色和蓝色表示严重和不严重，颜色越深表示数据被分得越纯，第一行表示划分指标，百分率表示被划分的比例，越往下分，就越纯，基尼指数越接近0，每一颗决策树所使用的样本都是不一样的，是有放回采样。

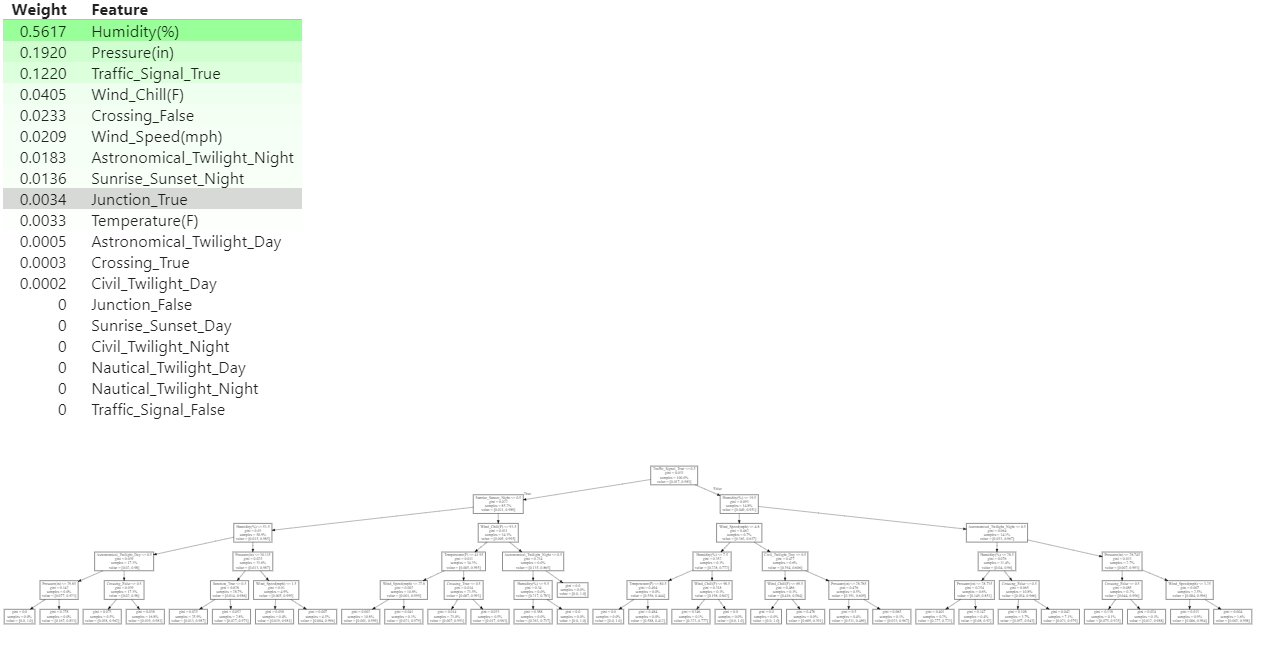


图 1.34

特征排序，依据是权重

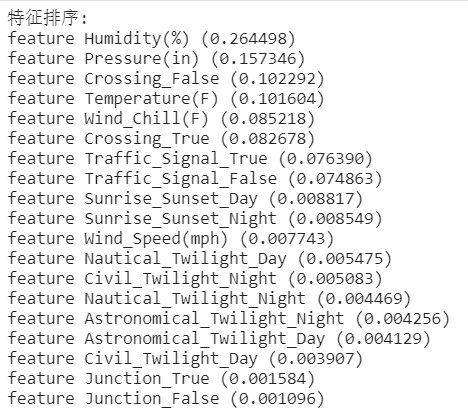


图 1.34

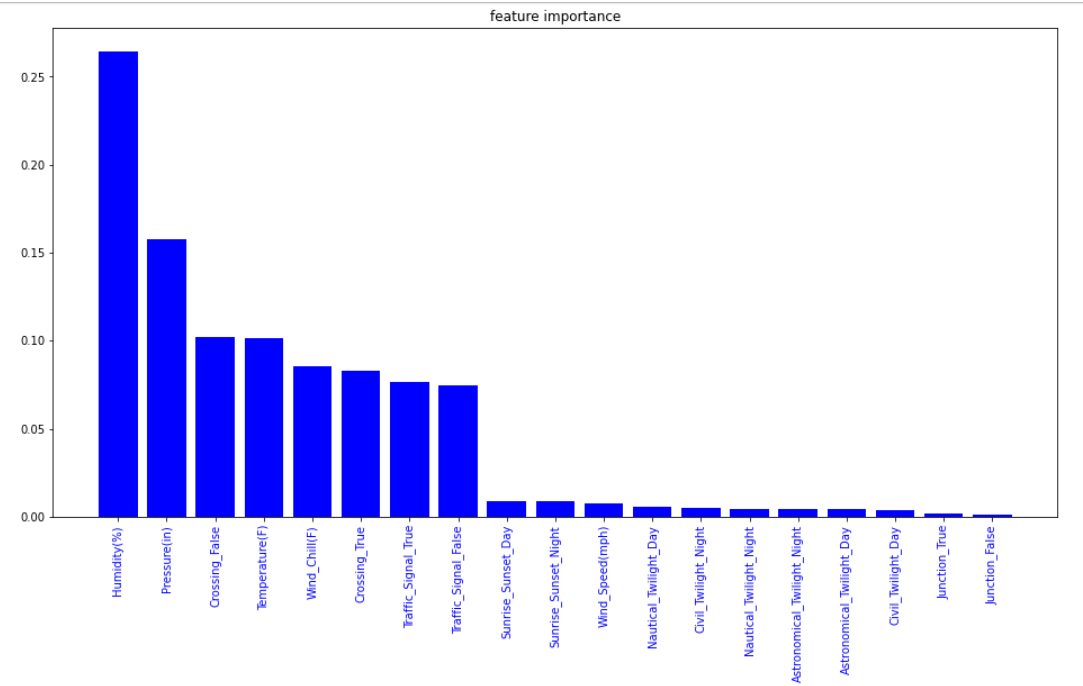


图 1.35

3.2.1定性结果的衡量

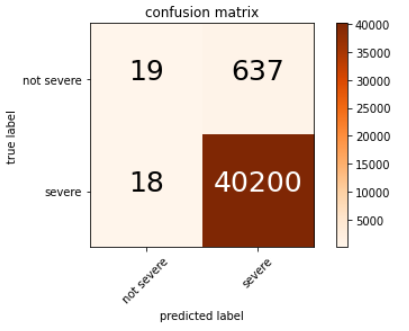


图 1.36

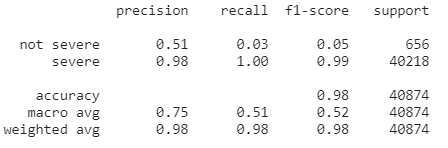


图 1.37

3.2.2定量结果的衡量

受试者工作特性曲线 ROC曲线这条线越靠近左上角，说明分类性能越好

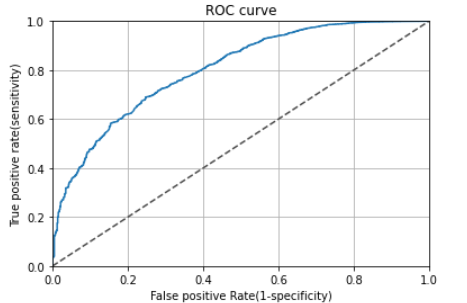


图 1.38

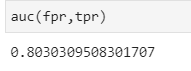


图 1.39

该曲线与虚线围成的面积约为0.80，说明分类效果一般。

3.2.3 对该模型赋予可解释性

使用shap机器学习可解释性分析工具包，对美国2016-2020年交通事故数据集上训练得到的随机森林模型进行可解释性分析。shap值是每个特征使得最后的分类结果反转的次数，可以根据shap值判断特征重要度。

特征重要度图：

对于某个特征，计算测试集每起交通事故的该特征shap值之和，shap值越高，特征越重要

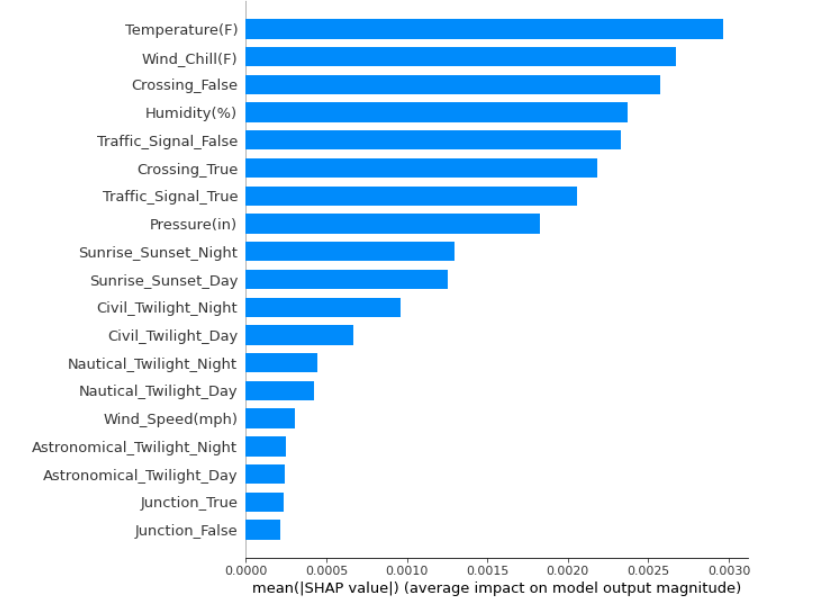


图 1.40

各特征的数值大小与各特征的shap值关系图：

每一行表示一个特征，红色表示该特征的值较高的数据点，蓝色值表示该特征的值较低的数据点，越靠右的点表示shap值越大，对预测为”严重“的正向影响越高，越靠左shap值越小，对预测为”严重“的负向影响越高。如果红和蓝能够区分开，那么就说明特征的数值大小的高和低是有不同影响的，例如Traffic\_Signal\_Flase，往右说明没有交通信号，对预测为事故严重的正向影响就越高。各个特征的大小和它对模型预测的影响非常清楚。

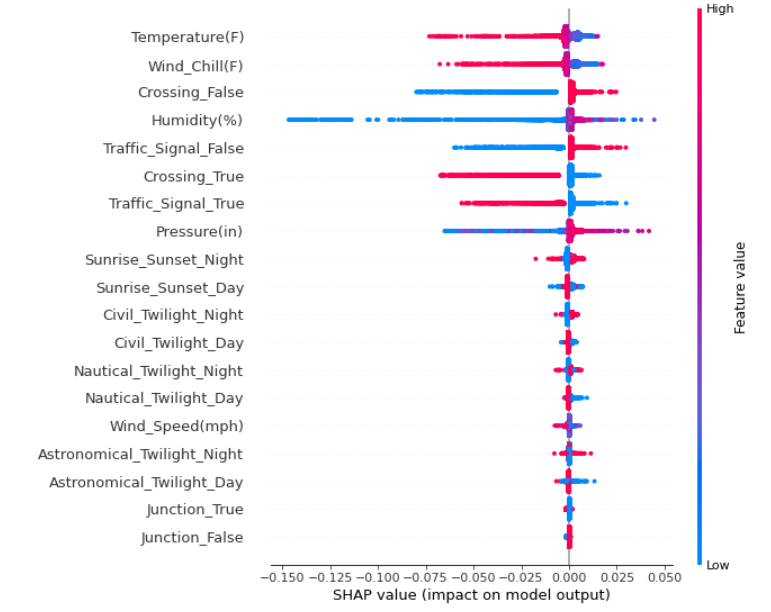


图 1.41

两两关系图：

将特征之间的两两关系考虑进来，主对角线上的图是上幅图所表达的含义，不在主对角线上的图，红点表示这两个特征均为高值，蓝点表示这两个特征均为低值，靠右表示对模型预测为“严重”有正向影响，shap值越高，靠左表示对模型预测为“严重”有负向影响，shap值越低。

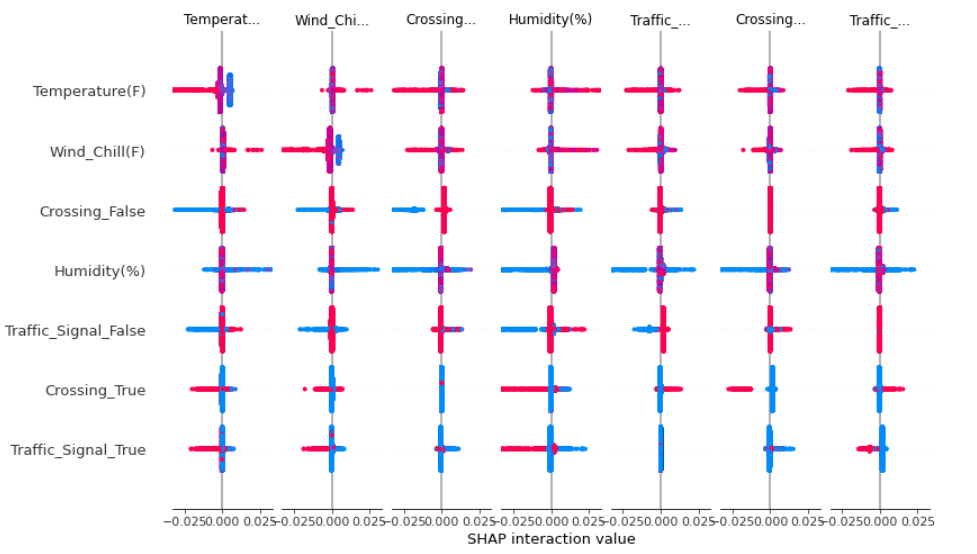


图 1.42

决策图：

展示测试集中部分数据的决策过程，每一行表示一个特征，越靠上的特征其shap值越大，这个图展示了决策路径。属于基于链接方法可视化高维数据

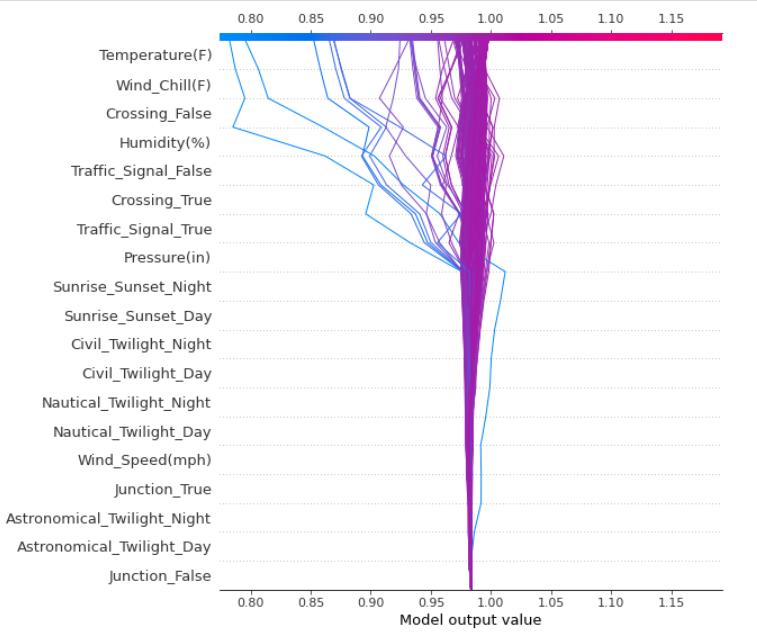


图 1.43：决策图

3.3本实验中采用的可视化方法总结

在数据探索，数据分析，以及实验结果的呈现中采用了不同的可视化方法：

3.3.1时间序列预测问题中可视化方法

1.一维时间序列图

将时间数据作为二维的线图显示，x轴表示时间，y轴表示其他变量

2.多个一维时间序列图

为了反映时间序列的周期性，采用分解的方法用多个一维时间序列图来表达

3.日历可视化

人类社会中时间分为年、月、日等多个等级，采用日历表达时间属性和识别时间的习惯符合，深浅表示事故发生的数量，例如图1.6可视化结果清晰展现了2020年美国新冠疫情爆发前后美国交通事故数量激增的情况。

4.动画显示法

对时序数据最直观的可视化方法就是将数据中的时间变量映射到显示时间上，在本实验中是用户控制的时间条，2016-2020年5年每隔2秒更新一帧的速度在10s内播放，且在动画播放过程中用户可以任意控制播放进度。例如图1.5、1.6、1.7

5.地理空间数据可视化

可视化过程将附着于空间位置的可视化对象的其他属性数据采用适当的视觉通道，在这里主要就是颜色、大小，亮度进行编码并绘制。每幅图上可视化了不同地理区域内发生的交通事故总数，其中一旁的颜色条（从冷色调到暖色调）定义了不同数值对应的色调区间，可以看到随着时间变化色调的变化，例如图1.5、1.6、1.7

1）点形数据的可视化：

点地图：在地图的相应位置摆放标记或者改变该点的颜色，标记的大小和颜色的深浅都代表着特定的含义。例如图1.5、1.7

1. 区域数据的可视化

等值区间地图：该方法假定地图各区域内的数据分布均匀，将区域内相应数据的统计值直接映射为该区域的颜色。例如图1.9、1.10、1.11

3.3.2二分类问题中高维数据的可视化方法

1.基于点的方法：散点图

2.基于线的方法：平行坐标图

3.基于区域的方法

1. 柱状图

堆叠柱状图、直方图

1. 表格显示

多维数据经常以表格的形式存储，对应的可视化方法可以采取表格形式，如热力图，它将规则化数据转化为颜色色调，其中每个规则单元格对应数据的某些属性，属性的值通过颜色映射表转换为不同的色调并填充规则单元。

4.基于链接方法可视化高维数据

决策图

5.层次数据可视化

在构建的随机森林模型中可视化一颗决策树，这里的承接关系为层次关系，问题越复杂，层次数据越复杂，自上而下是细分。采用结点链接法，是树型结构的直观表达，用结点表达数据个体，父结点和子结点之间用边表达层次关系，它的核心是结点和边的位置编码和视觉符号编码，这里是正交布局，结点沿水平或竖直方向排列，所有子结点在父结点的同一侧分布。