海关大数据项目

实训报告

姓名	夏小鱼
学号	1810800127
专业	计算机科学与技术
班级	18AI创新班
日期	2021年7月4日

目录

1 项目背景与目标

1.1 项目背景

中国每年都会从国外进口大量的动物，这些动物不仅要满足畜牧业的需要，也要满足日益增长的餐饮需求。但

这些进口动物可能会携带一些疾病，这些疾病会对社会公众的健康产生非常严重的影响。因此，相关部门对每一批进口动物都会进行严格的动物检疫。近些年来，随着大数据和人工智能技术的快速发展，可以借助海量数据对动物检疫流程进行更加精准的分析，使得动物疫病的检测更加精确，让工作人员清晰地掌握检疫各个环节的信息和情况，从而有效地辅助工作人员做好动物检疫的工作，进一步提高工作效率。

1.2 项目目标

本项目通过对真实的动物检疫数据进行处理、分析和可视化，了解检疫过程中的数据，并针对特定疫病建立检测模型，从而实现对进口动物疫病的智能检测。首先，针对每个检疫环节的数据，进行数据预处理和分析；其次，通过数据库的表连接操作，将需要分析的数据进行连接，更深入地对检疫流程进行分析；然后，从数据中筛选出疫病率最高的动物品种，并从该动物品种中挑选出患病数量最高的两种疫病作为建模目标，再分别对这两种疫病建立不同的检测模型；最后，对两种疫病的检测模型进行评估。图1是本项目的流程图：

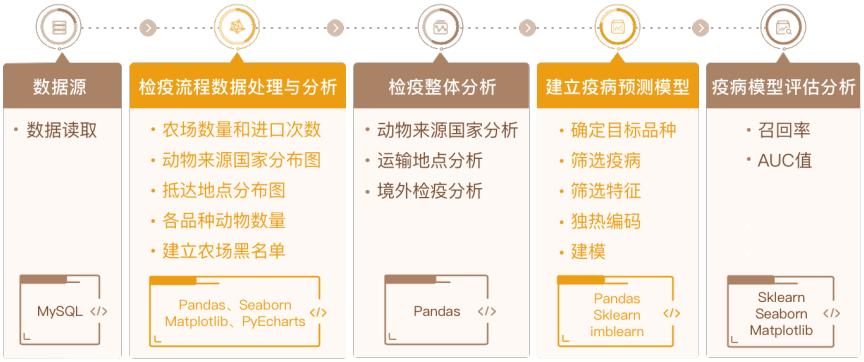

图1：项目流程图

2 检疫环节数据处理与分析

2.1 动物检疫流程介绍

通常来说，动物以批次为单位完成进口，一只动物从境外进口到中国需要经过相关的检疫流程，该流程包括7个环节，具体如图2所示：

图2：动物检疫流程

每个环节都存储着对应的数据，并存放在各自的数据库中，具体信息如表1所示：

表1：各检疫环节数据说明

检疫环节	数据库名	数据表
境外农场	OVERSEAS_FARM	import_farm、import_nation
境外检疫	OVERSEAS_QUARANTINE	overseas_quarantine_record、overseas_lab_record
境外起运	OVERSEAS_START	overseas_start_record
运输	TRANSPORT	transport_record
境内抵达	DOMESTIC_ARRIVE	domestic_arrive_record
境内检疫	DOMESTIC_QUARANTINE	domestic_quarantine_record
入境	FINAL_RESULT	animal_quarantine_result

2.2 境外农场环节分析

首先对境外农场进行分析，境外农场数据存放在数据库 OVERSEAS_FARM 中，境外农场数据库有两个数据表，分别是 import_farm 和 import_nation 。

import_farm 表存储的是各批次进口动物分别来自哪些农场，在进口动物时一批动物来自多个农场。其字段说明如表2所示：

表2： import_farm 表字段说明

列名	类型	说明
import_id	int	进口编号
farm_id	varchar	农场编号

import_nation 表存储的是各批次进口动物分别来自哪个国家，在进口动物时一批动物只来自一个国家，字段说明如表3所示：

表3： import_nation 表字段说明

列名	类型	说明
import_id	int	进口编号
nation	varchar	进口动物来源国家

原始数据存放在MySQL数据库中，首先需要从数据库中读取数据，在完成数据读取之后，进行数据格式转换。

由于上一节中从数据库导出的数据是嵌套元组，为了便于后续分析，需要使用Python中的Pandas库将数据格式转换为DataFrame。 DataFrame是Pandas中常用的二维数据结构。由于嵌套元组无法直接转换为DataFrame，所以先使用list()函数将其转换为列表，然后再转换为DataFrame，且在转换时需要设置数据的列名，具体方法为: data = pd.DataFrame(list(data_original) ,columns=columns_name) data_original : 需要转换的数据。 columns_name :每一列的名称，列表形式。

DataFrame为二维表结构，每一行代表一条记录，具体如表4所示：

表4： import_nation 表数据概览

	import_id	nation
0	184	澳大利亚
1	186	新西兰
2	187	澳大利亚
3	188	澳大利亚
4	189	澳大利亚

使用Pandas提供的函数，对数据进行预处理和数据探索性分析。首先对数据 import_farm 分析后发现本项目中总共有7758个农场、105批进口动物。

Pandas中的Series对象的 unique() 能够提取Series对象中数据的不同取值，去掉重复数据，返回所有不同取值组成的数组Python的内置函数 len() 可以返回一个对象（数组、列表等）的元素个数， 比如 num = len(list)， num 为 list 列表中元素的个数。结合 unique() 和 len() 可以统计一个DataFrame对象中某一列的不同取值。 统计农场数量 num_farm = len(import_farm['farm_id'].unique()) print(num_farm)
统计进口次数 num_import = len(import_farm['import_id'].unique()) print(num_import)

接着分析数据 import_nation ，统计每个国家向中国出口动物的次数，这些国家都包括是澳大利亚、新西兰、乌拉圭、加拿大、法国、美国、爱尔兰、阿根廷、丹麦、智利，最后将统计结果展示在地图上，如图3所示：

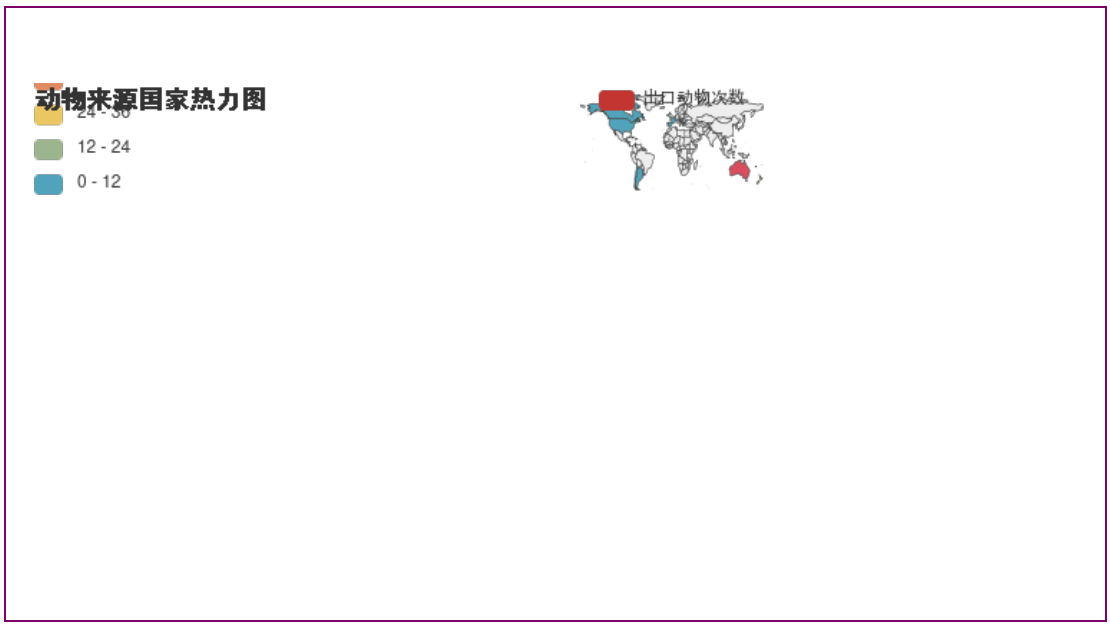

图3：出口国家分布

pyecharts模块中的子模块charts的 Map()类可以绘制热力图，具体语法:map = Map() map.add(name,data,maptype,is_map_symbol_show)map.set_series_opts() map.set_global_opts()
由热力图可以看到出口次数最多的两个动物来源国是澳大利亚和新西兰。

2.3 境外检疫环节分析

然后动物进行境外检疫，这部分数据存放在数据库 OVERSEAS_QUARANTINE 中，其中有两个数据表，分别是 overseas_quarantine_record 和 overseas_lab_certification 。

第一张表是境外检疫工作记录表 overseas_quarantine_record ，该表存储的是每一批进口动物在境外检疫场进行检疫的工作记录，具体字段如表5所示：

表5： overseas_quarantine_record 表字段说明

列名	类型	说明
import_id	int	进口编号
overseas_field_id	varchar	境外检疫场
prevention_disease	varchar	预防疫病名称
prevention_medicine	varchar	预防用药名称
prevention_method	varchar	预防用药方法
desinsectization_medicine	varchar	驱虫用药名称
desinsectization_dosage	varchar	驱虫用药剂量
desinsectization_method	varchar	驱虫用药方法
immunization_category	varchar	免疫接种类型
immunization_disease	varchar	免疫接种疫病名称
immunization_method	varchar	免疫接种方法

第二张表是境外检疫实验室工作记录表 overseas_lab_record ，该表存储的是每一批进口动物对应的境外检疫实验室的工作记录，如表6所示：

表6： overseas_lab_record 表字段说明

列名	类型	说明
import_id	int	进口编号
overseas_lab	varchar	境外检疫实验室
retention_method	varchar	样品保存方法

境外检疫工作在农场所属国的检疫场进行，并由该国家检疫实验室负责检疫，检疫实验室这一列部分名称过长，需要对其进行替换，替换完成后得出实验室名单。

2.4 境外起运环节分析

动物经过境外检疫后送至起运地点。这部分数据存放在数据库 OVERSEAS_START 中，它只有一个数据表 overseas_start_record ，该表存储的是运输工具在运输动物前进行防疫处理的相关数据，其字段如表7所示：

表7： overseas_start_record 表字段说明

列名	类型	说明
import_id	int	进口编号
start_medicine	varchar	境外起运用药名称
start_method	varchar	境外起运用药方法

由于数据 overseas_start_record 中存在缺失值，需要对其进行缺失值填充。

为了考察DataFrame对象overseas_start_record的缺失值情况，我们使用isnul1()方法来检测缺失值，该方法会返回一个DataFrame对象，表明数据是否存在缺失值。如果存在缺失值，那么，overseas_start_record对应位置的取值为True ;不是，则为False 。isnul1()方法常与sum()方法结合使用，可以统计DataFrame对象中每一列的缺失值数量，具体方法为：data.isnul1( ).sum()，其中data是一个DataFrame类型数据。 缺失值检测完毕后，需要对缺失值进行处理。DataFrame对象的fillna()方法可以对缺失值进行填充。

2.5 运输环节分析

运输环节的数据存放在数据库 TRANSPORT 中，它只有一个数据表 transport_record 。 transport_record 存放着每批次进口动物的运输过程的信息，其字段如表8所示：

表8： transport_record 表字段说明

列名	类型	说明
import_id	int	进口编号
fodder	varchar	饲料
fodder_country	varchar	饲料来源国家
bedding	varchar	垫草
bedding_country	varchar	垫草来源国家
vehicle	varchar	运输工具
start_location	varchar	起运地点
arrive_location	varchar	抵达地点

读取数据后，对运输过程进行分析，首先分析动物运输工具，结果如图4所示：

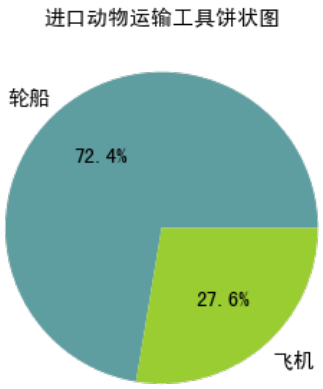

图4：动物进口运输工具

运输工具信息存放在运输记录表中的vehicle字段中，首先计算不同运输工具的帧数，然后绘制饼图展示计算结果.　使用Series类中的value_counts()方法统计不同取值的频数，然后使用plot.pie()方法绘制饼状图，其具体方法如下:
fig = plt.figure(figsize=(7,5))#设置图像大小 data['列名'].value_counts( ).plot.pie()# 绘制饼状图plt.title("title_name")#设置图标题 plt.show()#绘图

接着分析国内各城市接收进口动物的次数如图5所示：

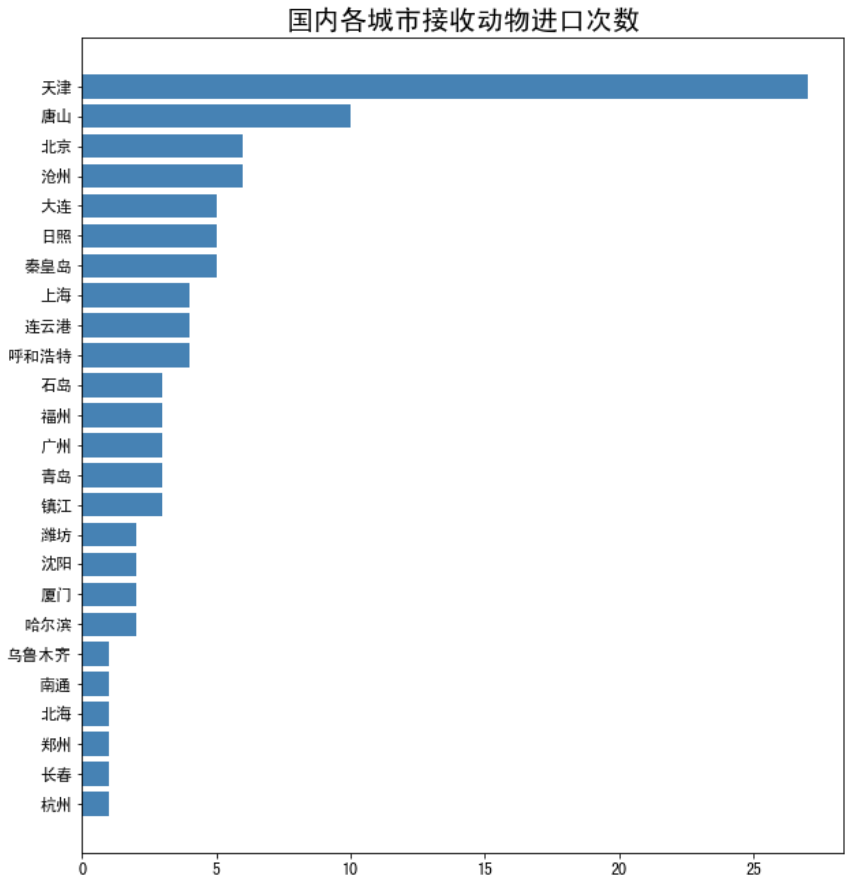

图5：国内各城市接收动物进口次数

由条形图可以看出，接收动物批次最多的四个城市依次是天津、唐山、沧州和北京。

为了更清楚的观察国内港口和国内机场的地理位置，绘制国内港口和国内机场的分布图，结果发现在不论是机场还是港口，抵达城市大多分布在沿岸城市。

图6：国内港口和机场位置分布

最后为了更直观的了解不同城市接收进口动物次数的差异，我们将绘制国内城市接收进口动物次数热力图，这样既能观察地理位置分布又能了解各城市接收进口动物次数。

图7：国内城市接收进口次数热点

2.6 境内抵达环节分析

动物经过运输后抵达中国，这部分数据存放在数据库 DOMESTIC_ARRIVE 中，它只有一个数据表 domestic_arrive_record ，该表记录每批动物在抵达时的数据，其字段如表9所示：

表9： domestic_arrive_record 表字段说明

列名	类型	说明
import_id	int	进口编号
death	varchar	运输死亡数量
entry_time	date	入境时间

由于动物在运输过程中的颠簸、碰撞、踩踏以及每个动物自身的健康状况不同，运输过程会使动物死亡，因此在动物抵达国内时查看每批次动物的死亡情况如何。由于运输死亡数量 death 中存在异常值，因此先对异常值进行替换，然后分析各批次进口动物的死亡数量，结果发现绝大部分批次的进口动物运输死亡数量都为0。

2.7 境内检疫环节分析

接下来动物接受境内检疫，境内检疫的数据存放在数据库 DOMESTIC_QUARANTINE 中，其只有一个数据表 domestic_quarantine_record ，该表记录的是境内检疫场在接收动物前所做的准备工作以及负责检疫的实验室等数据，具体字段如表10所示：

表10： domestic_quarantine_record 表字段说明

列名	类型	说明
import_id	int	进口编号
domestic_filed_id	varchar	境内检疫场编号
medicine_first	varchar	境内检疫场第一次用药
medicine_second	varchar	境内检疫场第二次用药
medicine_third	varchar	境内检疫场第三次用药
domestic_method	varchar	境内检疫用药处理方法
domestic_lab	varchar	境内检疫实验室
unqualified_handle	varchar	不合格动物处理方法

统计后发现共有61个境内检疫场负责105批进口动物，每个境内检疫场负责过的进口动物批次如表11所示：

表11：境外检疫场负责的进口批次

境内检疫场	进口编号
FID001	[184, 263, 295, 315]
FID002	[186, 359]
FID003	[187, 284]
FID004	[188, 216, 262]
FID005	[189, 275]
……	……

境内检疫记录表中记录了动物在境内检疫场进行检疫的相关数据，其中domestic_field_id这一列是检疫场的名称。统计每个检疫场处理过的进口批次，需要用到groupby()方法进行分组聚合，分组与聚合的过程分为三步：拆分:将数据按照某一列拆分为若干组 应用:对每个分组执行同一个方法 合并:将计算的结果合并。
请使用groupby()方法按照字段domestic_field_id对境内检疫记录表进行分组，并保存在field_group中。
请使用len()函数计算field_group的长度，得到境内检疫场的个数，并保存在n 中。 请使用unique()方法提取field_group 中import_id字段的唯一值，得到各检疫场处理的进口批次，并保存在count中。

2.8 入境结果环节分析

入境后的每只动物的数据存放在数据库 FINAL_RESULT 中，其只有一个数据表 animal_record ，具体字段描述如表12所示：

表12： animal_record 表字段说明

列名	类型	说明
animal_id	int	动物编号
import_id	int	进口编号
farm_id	varchar	农场编号
status	int	是否患有疫病
disease_name	varchar	疫病名称
gender	varchar	性别
category	varchar	动物品种
year	year	年份

本项目中总共有动物451170只，包含四个品种分别是猪、马、牛、羊，各品种数量如图8所示：

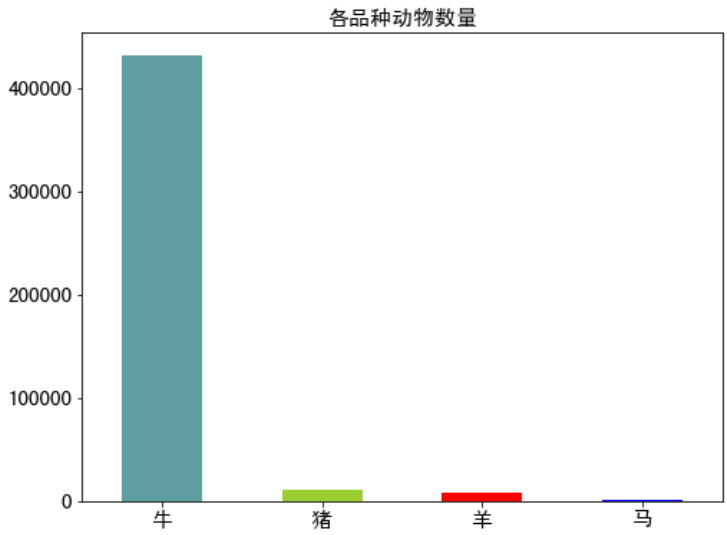

图8：各品种动物数量

接着对每批进口动物品种分析，结果发现每批动物进口只包含一个动物品种。最后，为了减少疫病风险，需要统计疫病率超过50%的农场列入农场黑名单。

如果某些农场出口的动物有50%以上带有疫病，则说明该农场产出的动物患病的可能性很高，从这些农场进口动物的风险很大，因此需要建立一份农场黑名单来获取这些农场的信息，本小节将从上一节计算的农场疫病率farm_disease_rate中筛选出疫病率高于0.5的农场。 farm_disease_rate是一个Series对象，它的索引是农场编号，因此我们需要将疫病率高于0.5的索引筛选出来，然后再使用tolist()方法将索引转为列表，即可得到农场黑名单，对Series对象按照某个条件筛选索引并转为列表的方法如下:
list_1=data_series[data_series>0.5].index.tolist()，其中data_series是一个Series对象。
建立农场黑名单列表 farm_blacklist = farm_disease_rate[farm_disease_rate>0.5].index.tolist()

3 整体分析

前边已经对单个的检疫环节有所了解，由于单独对一个检疫环节进行分析能够获得的信息有限，所以需要结合多个检疫环节的数据进行深入的分析，本环节将进行表连接，连接数据后再进行分析。

3.1 动物来源国分析

首先在之前的分析中了解到进口动物来自多个国家，在进口动物时需要对每个动物来源国的情况有一定的了解，但由于各批次动物来源国的数据和动物检疫的数据存放在不同的数据库中，因此在分析前需要进行表连接。每批进口动物来源国家的数据存放在数据库 OVERSEAS_FARM_PROCESSED 的 import_nation 表中，动物检疫结果数据存放在数据库 FINAL_RESULT_PROCESSED 的 animal_quarantine_result 表中，首先在数据库中连接这两张表，然后转为DataFrame格式命名为 animal_nation ，基于连接的数据分析各国家的疫病率和各动物品种的来源国家。

为了分析各国家疫病率和动物品种的来源国家，需要从动物来源国家表中提取字段nation，从动物检疫结果表中提取字段animal_id 、import_id .status和category 。因为两张表有共同字段import_id，所以可以通过import_id连接两张表。使用MySQL左连接语句LEFT JoIN完成表连接。 LEFT30IN示例： 数据库oVERSEAS_FARM_PROCESSED中存放着表import_nation，数据库 FINAL_RESULT_PROCESSED中存放着表animal_quarantine_result，通过共同字段import_id连接这两张表，并按照顺序提取字段

animal_id，import_id，category，nation。注意:由于两张表中都有字段import_id ， 所以在提取字段时需要指明是要提取哪张表中的字段import_id。

首先计算各国家疫病率，结果如表13所示：

表13：各国疫病率

国家	疫病率
乌拉圭	0.200955
加拿大	0.021596
新西兰	0.059712
智利	0.026284
法国	0.015370
澳大利亚	0.119474
爱尔兰	0.090909
美国	0.014968
阿根廷	0.000000

首先根据国家对animal_nation进行分组，这样可以得到每个国家动物的疫病数据。然后根据分组结果的疫病列计算动物来源国家的疫病率，使用count()和sum()统计总数与患病数，然后两者相除得到动物来源国家的疫病率。
如果一个国家的出口动物疫病率超过50%，那么该国家出口的动物患病风险很高，应尽量避免从这个国家进口动物。 基于DataFrame对象animal_nation，统计进口动物来源国家的疫病率。 使用groupby()方法按照nation列对animal_nation进行分组，分组结果保存在变量animal_nation_group 中。 根据animal_nation_group的status列计算动物来源国家的疫病率，首先使用count()方法和sum()方法统计各来源国家的动物总量和患病动物的数量，然后将两者相除得到动物来源国家的疫病率，保存为Series对象nation_disease_rate(小数点后保留2位)。 可以看出数据中没有疫病率超过50%的国家。 nation丹麦 0.00 乌拉圭 0.20 加拿大 0.02 新西兰 0.06 智利 0.03 法国 0.02 澳大利亚 0.12 爱尔兰 0.09 美国 0.01 阿根廷 0.00 Name: status，dtype: float64

然后分析每个动物品种的来源国家，结果如表14所示：

表14：各动物品种来源国

动物品种	来源地
牛	[澳大利亚, 新西兰, 乌拉圭, 阿根廷, 智利]
猪	[加拿大, 美国, 丹麦, 法国]
羊	[澳大利亚]
马	[新西兰, 爱尔兰, 澳大利亚, 阿根廷, 法国]

有些动物来自于南北半球，有些动物只来自于南半球。有些动物横跨多个大洲，有些只来自于一个州。羊和牛只来自于南半球。

3.2 动物起运地点分析和抵达地点分析

下面将分析动物起运地点和抵达地点的疫病率，起运地点和抵达地点存储在数据库 TRANSPORT_PROCESSED 中的 transport_record 表，动物检疫结果数据存放在数据库 FINAL_RESULT_PROCESSED 的 animal_quarantine_result 表中。为了分析各起运地点和抵达地点的疫病率，需要在数据库中连接表 transport_record 和 animal_quarantine_result ，连接后的数据转为DataFrame格式并命名为 animal_transport 。

使用连接后的数据 animal_transport 分析每个起运地点的疫病率，结果如表15所示：

表15：起运地点疫病率

起运地点	疫病率
卢森堡	0.09589
吉朗	0.079903
哥本哈根	0.000000
基督城	0.033272
墨尔本	0.188039
奥克兰	0.000000
巴黎	0.009298
布宜诺斯艾	0.000000
悉尼	0.000000
提马鲁	0.078548
斯波茨伍德	0.085213
波特兰	0.121428
珀斯	0.038663
纳皮尔	0.045797
芝加哥	0.018302
蒙特维的亚	0.178524
阿姆斯特丹	0.090909

大多数起运地点的疫病率都在0.1以下。其中哥本哈根，奥克兰，布宜诺斯艾和悉尼的疫病率为0，而蒙特维的亚的疫病率最高。

然后分析每个抵达地点的疫病率，结果如表16所示：

表16：抵达地点疫病率

抵达地点	疫病率
上海	0.000000
乌鲁木齐	0.017526
北京	0.03871
北海	0.052657
南通	0.024698
厦门	0.109124
呼和浩特	0.095198
哈尔滨	0.270042
唐山	0.092337
大连	0.165639
天津	0.111327
广州	0.117275
日照	0.053455
杭州	0.013441
沈阳	0.000000
沧州	0.138754
潍坊	0.056226
石岛	0.091943
福州	0.179362
秦皇岛	0.071934
连云港	0.074949
郑州	0.240397
镇江	0.091105
长春	0.000000
青岛	0.163721
黄埔	0.109621

最后分析各抵达地点接收的动物品种，结果如表17所示：

表17：地点接收品种

抵达地点	品种
上海	['马']
乌鲁木齐	['猪']
北京	['马' '牛' '猪']
北海	['牛']
南通	['牛']
厦门	['牛']
呼和浩特	['牛' '羊']
哈尔滨	['牛' '猪']
唐山	['牛']
大连	['牛' '猪']
天津	['牛' '猪']
广州	['牛']
日照	['牛']
杭州	['猪']
沈阳	['马']
沧州	['牛']
潍坊	['牛']
石岛	['牛']
福州	['牛' '猪']
秦皇岛	['牛']
连云港	['牛']
郑州	['牛']
镇江	['牛']
长春	['猪']
青岛	['牛' '猪']
黄埔	['牛']

结果发现，不同抵达地点的疫病率也存在较大差异，整体疫病率范围是0%-27%，例如上海、长春、沈阳三个城市的疫病率为零，而哈尔滨的疫病率高达27%。疫病率高的抵达地点最容易爆发疫病，因此需要对疫病率高的抵达地点采取更严格的防疫措施。　结果发现，每种动物的抵达地点不同，牛会运送到绝大部分城市。结合前一节的各抵达地点的疫病率可以发现，疫病率为零的几个抵达地点都没有牛运送过去，所以总体中牛的疫病率可能比较高。

3.3 境外检疫分析

接下来将分析境外检疫场和样品保存方法的疫病率，境外检疫场和样品保存方法的数据分别存放在数据库 OVERSEAS_QUARANTINE_PROCESSED 中的表 overseas_quarantine_record 和表 overseas_lab_record ，动物检疫结果数据存放在数据库 FINAL_RESULT_PROCESSED 的 animal_quarantine_result 表中，在分析前需要对这三张表进行多表连接，连接后转为DataFrame并命名为 animal_overseas_quarantine 。

> 多表连接语句:SQL对一条SELECT语句中可以连接的表的数目没有限制，创建连接的基本规则也相同，首先列出所有表，然后定义表之间的关系，具体语法为: SELECT target FROM table_1,table_2,table_3 WHERE table_1.col_1=table_2.col_2AND table_2.col_2=table_3.col_3AND table_1.col_1=table_3.col_3 target是从被连接的几张表中提取出的字段。 table_1 ,table_2和table_3是需要连接的表。, col_1.col_2和col_3为多表连接时依赖的字段。

使用连接后的数据 animal_overseas_quarantine 计算各境外检疫场的疫病率，结果发现各境外检疫场的疫病率跨度为0%到30%之间，结果如表18所示：

表18：境外检疫场疫病率

境外检疫场编号	疫病率
OVFD007	0.305294
OVFD011	0.29774
OVFD022	0.262296
OVFD053	0.240397
OVFD055	0.219281
……	……

最后分析了各实验室不同的样品保存方法的疫病率，发现疫病率并没有太大差异，结果如表19所示：

表19：不同样品保存方法的疫病率

样品保存方法	疫病率
冷冻保存	0.124591
冷藏保存	0.11027
常温保存	0.095743

4 构建疫病预测模型

本环节构建疫病预测模型，数据中有四个品种分别是猪、马、牛、羊，本项目选出疫病率最高的动物品种作为目标品种，然后在目标品种的所有疫病中选出患病数量最高的两种疫病作为目标疫病构建预测模型。

4.1 筛选动物品种

首先计算各动物品种的疫病率，筛选出疫病率最高的品种。

牛，猪，羊，马的疫病率分别为0.111201，0.015257，0.097184，0.025316。 最终筛选出的动物品种是牛。

4.2 确定目标疫病

确定目标动物品种为牛之后，从 animal_quarantine_result 中取出牛的数据保存为 cow ，每只动物的疫病情况记录在 disease_name 这一列中，根据 disease_name 建立牛的疫病标签数据 cow_disease_data 。 cow_disease_data 中的每一列是某一种病的患病情况，每一行代表一只动物的患病情况，如果某只动物患有某种病，那么该只动物在这一列取值为1，不患病则取值为0。因此对 cow_disease_data 每一列求和即可得到每一种疫病的患病数量，选择数量较高的前两个作为目标疫病。

在确定了目标品种后，我们选取患病数量最多的两种疫病作为目标疫病建立检测模型。DataFrame对象cow中的disease_name列存放着每只牛患病的数据，如果患病则存放疫病名称，不患病则为“无”。具体处理办法是生成一个DataFrame保存疫病的标签数据，命名为cow_disease_data，每一列存放牛的一种疫病标签数据，如果某只牛患有该病，那么这只牛在这一列的取值就为1，如果不患该病，那么取值为0，通过cow_disease_data计算每种疫病的患病数量。得到cow_disease_data后，首先使用sum()方法对cow_disease_data进行按列求和，然后使用sort_values()进行排序，设置参数ascending为False进行降序排序，从而得到各疫病患病数量的降序排序，查看前两种患病数量最多的疫病。
结果发现，患病数量最多的两种疫病为牛传染性鼻气管炎和牛地方流行性白血病，所以选取这两种病作为建模的目标疫病。

4.3 疫病标签数据保存

现在已经筛选出牛患病数量最高的两种疫病分别为牛传染性鼻气管炎和牛地方流行性白血病，从 cow_disease_data 中取出这两种疫病的标签数据，其中牛传染性鼻气管炎标签数据保存为 disease_IBR ，牛地方流行性白血病标签数据保存为 disease_EBL 。

4.4 预测模型指标构建

两种疫病的标签数据已经提取完成，现在需要准备建模使用的特征。在第三章建立农场黑名单中计算了各农场的疫病率，农场疫病率对构建模型非常重要，但需要对其进行离散化，同时也对运输死亡数量进行离散化，将离散化后的数据作为新的特征加入模型指标。

因为动物检疫的数据分布在七个环节的数据库中，因此筛选有效特征后，需要将这些特征汇总在一起。由于在数据中，牛只有雌性，因此将性别剔除，同时剔除时间、进口编号等对建模没有意义的特征，最后保存为 cow 。

最终所有特征汇总如表20所示：

表20：原始特征汇总

特征	说明
import_id	进口编号
nation	进口动物来源国
overseas_field_id	境外检疫场
prevention_disease_name	预防疫病名称

特征	说明
prevention_medicine_name	预防用药名称
prevention_method	预防用药方法
desinsectization_medicine_name	驱虫用药名称
desinsectization_dosage	驱虫用药剂量
desinsectization_method	驱虫用药方法
immunization_category	免疫接种类型
immunization_disease_name	免疫接种疫病名称
immunization_method	免疫接种方法
overseas_lab	境外检疫实验室
retention_method	样品保存方法
start_medicine	境外起运用药名称
start_method	境外起运用药方法
fodder	饲料
fodder_country	饲料来源国家
bedding	垫草
bedding_country	垫草来源国家
vehicle	运输工具
start_location	起运地点
arrive_location	抵达地点
death	运输死亡数量
domestic_field_id	境内检疫场
medicine_first	境内检疫场第一次用药
medicine_second	境内检疫场第二次用药
medicine_third	境内检疫场第三次用药
domestic_method	境内检疫用药处理方法
domestic_lab	境内检疫实验室
disease_rate	农场疫病率
year	年份

4.5 特征独热编码

接下来对 cow 进行独热编码得到特征数据 cow_one_hot 。

由于之后将建立逻辑回归模型，与线性回归类似，逻辑回归需要对无序离散型字符特征进行独热编码，其目的是可以考察定性因素对因变量的影响，将不能够定量处理的特征量化。由于数据集cow中的部分特征是无序离散型，因此我们需要对cow中的这部分数据进行独热编码。
使用Pandas的pd.get_dummies()可以实现独热编码，该方法会自动识别字符型数据因此不需要指定特定的列，具体方法为: data_onehot = pd.get_dummies(data, columns=None) data表示数据集 columns表示数据集中需要独热编码的字段

4.6 牛传染性鼻气管炎预测模型构建

完成独热编码后，开始对牛传染性鼻气管炎建立预测模型，将数据按照4:1的比例随机划分为训练集和测试集，然后使用逻辑回归构建模型。

结果发现模型在测试集的召回率较低，使用随机森林构建模型后发现召回率有所提升，但提升并不明显。在尝试不同算法构建模型后，发现模型效果没有明显提升。由于疫病标签数据的样本类别是不均衡的，因此尝试从样本本身出发，使用过采样对数据进行处理，处理后再构建模型，本项目中使用SMOTE算法进行过采样。

在训练集和测试集划分完成之后，对训练集进行过采样处理，使用随机森林对过采样后的数据构建模型，相较于没有进行过采样的模型，召回率有明显提升。

最后在过采样的基础上对随机森林进行参数调优提升模型效果，设置随机森林参数 n_estimators 为 25， max_depth 设置为10，最终模型的召回率再次有了小幅提升。

模型效果一直不理想的一个重要原因是训练集存在类别不平衡问题，患病样本的数量要远远少于不患病样本的数量，使得模型不能充分学习到患病样本的信息，从而直接影响判断能力。过采样是处理类别不平衡问题的一种方法，基本思想是通过一定的方法合成少数类样本，增加少数类样本的数量，从而缩小少数类样本与多数类样本的数量差距。Synthetic minoritye over-sampling technique(SMOTE,合成少数类过采样技术)算法是常见的过采样技术之一，我们将通过SMOTE算法对训练集进行过采样处理，平衡训练集中两种类别样本的数量。 由结果可以看到，召回率较之前确实有了较为明显的提升，过采样的效果明显。

4.7 牛地方流行性白血病预测模型构建

接着对牛地方流行性白血病构建模型，首先按照4:1的比例随机划分数据为训练集和测试集，并对训练集进行过采样处理；然后使用逻辑回归对过采样后的数据建立模型，并对测试集进行预测；最终牛地方流行性白血病预测模型在测试集的召回率非常高，对患病样本的判断能力强。

5 动物疫病预测模型效果评估

前面已经完成两种疫病预测模型的构建，本环节评估模型的效果如何。

5.1 动物疫病预测模型评价指标介绍

动物疫病预测是一个二分类问题，即动物患有某种病或不患有某种病，因此本项目中将使用混淆矩阵和AUC值评估模型效果，并绘制每个模型的ROC曲线。

		预测值	
		1	0
真实值	1	TP	FN
	0	FP	TN

图9：混淆矩阵

在疫病预测中，通常关注患病的个体，用1表示患病即正例，0则表示不患病即负例。

- TP ：表示真实值为1且预测值为1的数据，即真正例。
- FN ：表示真实值为1但预测值为0的数据，即假负例。
- FP ：表示真实值为0且预测值为1的数据，即假正例。
- TN ：表示真实值为0且预测值为0的数据，即真负例。

下面介绍ROC曲线和AUC值：

ROC曲线全称为受试者工作特征曲线(Receiver Operating Characteristic Curve，简称ROC曲线)，其横坐标为假正率(False Positive Rate)，纵坐标为真正率(True Positive Rate)，TPR和FPR的计算方法为：

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$
$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

ROC曲线下的面积即为AUC值，它能够从整体上衡量一个模型的好坏。

> 需要使用roc_curve()函数计算出ROC曲线需要的数据，sklearn.metrics.roc_curve(y_true,y_score)，y_true和y_score分别为真实标签和预测为正类（1）的概率。该函数将返回一个三元组（fpr(假正率)，tpr(真正率)，threshold(用于计算fpr和tpr的阈值)）。

5.2 牛传染性鼻气管炎预测模型评估

牛传染性鼻气管炎预测模型在测试集的混淆矩阵热力图如图10所示。

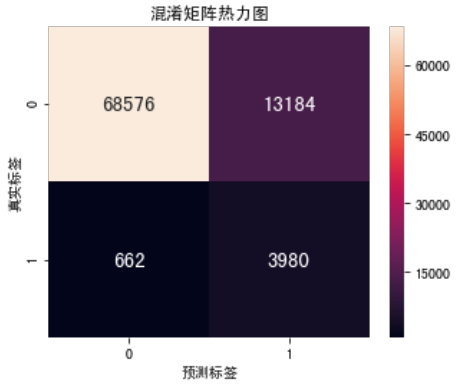

图10：牛传染性鼻气管炎预测模型混淆矩阵

通过图10可以看出，大部分患有牛传染性鼻气管炎的动物被模型准确识别，但模型判断有牛传染性鼻气管炎的动物中，有近两万只并没有患病，因此误判比较高。下面观察牛传染性鼻气管炎预测模型的ROC曲线，其AUC值为0.907，ROC曲线如图11所示：

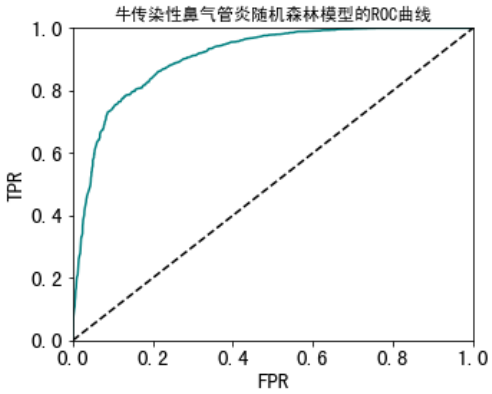

图11：牛传染性鼻气管炎预测模型ROC曲线

然后使用随机森林的 feature_importances_ 属性输出牛传染性鼻气管炎预测模型排在前20位的特征名称，由于特征是经过独热编码之后的，因此需要提取出原始特征名，最终结果如表21所示：

表21：牛传染性鼻气管炎测模型重要特征

特征	含义
disease_rate	农场疫病率
start_location	起运地点
nation	进口动物来源国家
fodder_country	饲料来源国家
prevention_medicine_name	预防用药名称
bedding_country	垫草来源国家
desinsectization_medicine_name	驱虫用药名称
overseas_field_id	境外检疫场
start_medicine	起运用药名称
arrive_location	抵达地点

我们使用训练好的随机森林模型输出特征重要性，具体的方法为rf.feature_importances_，筛选出重要性排在前10位的特征。

5.3 牛地方流行性白血病预测模型评估

牛地方流行性白血病预测模型在测试集的混淆矩阵如图12所示。

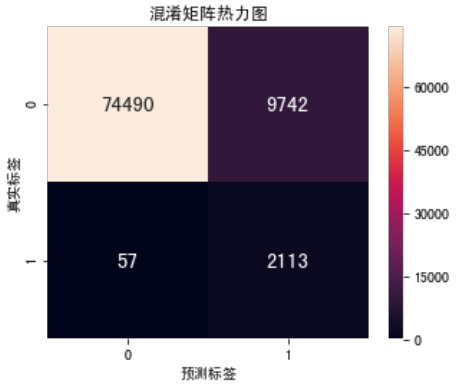

图12：牛地方流行性白血病预测模型混淆矩阵

可以发现在牛地方流行性白血病预测模型中，超过90%患病样本被模型识别，并且误判并没有非常高，说明模型预测效果较好。下面观察牛地方流行性白血病预测模型的ROC曲线，其AUC值为0.976，ROC曲线如图13所示：

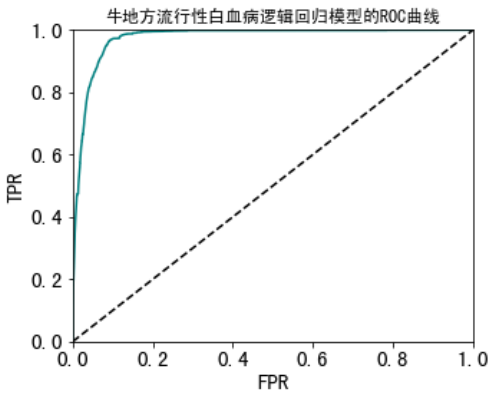

图13：牛地方流行性白血病预测模型ROC曲线

6 项目总结与心得体会

6.1 项目总结

项目目的：　本项目通过对脱敏的动物检疫数据进行处理、分析和可视化，并针对特定疫病建立疫病检测模型,从而实现对进口动物疫病的智能检测。
项目流程：　首先，动物检疫流程包含七个步骤，针对每个检疫步骤的数据，进行数据预处理和分析;其次，通过数据库的表连接操作，结合不同检疫步骤的数据，更深入地对数据进行分析;然后，从数据中筛选出疫病率最高的动物品种，并从该动物品种中选出患病数量最多的两种疫病作为建模目标，分别对这两种疫病建立检测模型;最后，对两种疫病的检测模型进行评估。 流程图： 数据源：数据读取---->检疫流程数据处理与分析------>检疫整体分析------->简历疫病预测模型---------->疫病模型评估分析
检疫流程数据处理与分析：　动物检疫流程共包含七个步骤，需要对每个步骤的数据进行数据读取和探索性分析。首先，从数据库中读取出数据并对部分数据进行异常值处理;然后进行初步分析和可视化分析包括动物批次的数量统计、农场的数量统计、进口动物的来源国家地图分布、运输过程中使用的运输工具等等。
建立农场黑名单：　如果某些农场出口的动物有50%以上带有疫病，则说明该农场产出的动物患病的可能性很高，从这些农场进口动物的风险很大，因此需要建立一份农场黑名单来获取这些农场的信息。　将从计算的农场疫病率farm_disease_rate 中筛选出疫病率高于0.5的农场。 farm_disease_rate是一个Series对象，它的索引是农场编号，因此我们需要将疫病率高于0.5的索引筛选出来，然后再使用tolist()方法将索引转为列表，即可得到农场黑名单。 对Series对象按照某个条件筛选索引并转为列表的方法如下: list_1 = data_series[data_series>0.5].index.tolist()，其中data_series是一个Series对象。
检疫整体分析：　首先进行初步分析和可视化分析包括动物批次的数量统计、农场的数量统计、进口动物的来源国家地图分布、运输过程中使用的运输工具等等。统计对象患病数量与总数量，两者相除获得患病率。最后整合多个步骤的数据，对多个数据表进行连接操作，以更深入地理解动物检疫数据，挖掘更多有价值的信息。主要从动物来源国家、动物运输地点、和境外检疫三个方面进行综合分析。我们在进行分析的时候需要利用多表之间的共同字段采用SQL语句进行连接。
构建动物疫病检测模型：　数据中包含的动物品种共有四种，分别是猪、马、牛和羊。首先根据每个动物品种的疫病率，选择疫病率最高的动物品种作为目标品种;然后在目标品种的所有疫病中选择患病数量最多的两种疫病作为目标疫病;最后分别对每种目标疫病建立疫病检测模型，并使用过采样、参数调优等方法进一步提升模型效果。
疫病模型评估分析：　在上一环节中，分别对两种疫病建立了检测模型，那么每个模型的效果究竟如何?这一环节中，将使用召回率和AUC值对模型进行评估。召回率是指在患病动物中能够被模型识别的比例;AUC值代表ROC曲线下的面积，其取值范围是[0.5,1]，它能衡量模型整体的好坏，它的取值越大代表模型效果越好。
项目总结：　本项目基于真实动物检疫数据，首先从数据库中分别读取每个检疫步骤的数据，对读取后的

数据进行处理和分析，并使用Seaborn和Matplotlib呈现数据分析结果;其次通表连接，进一步了解和探索数据，从数据中得到更多的信息;然后在疫病率最高的动物品种中选择患病数量最多的两种疫病构建疫病检测模型，使用过采样、参数调优逐步提升模型效果;最后通过召回率和AUC值对两种疫病的检测模型进行了评估。　本项目开发的疫病检测模型可以帮助检疫单位有效的检测动物疫病，减轻工作量并进一步提高工作效率。

6.2 心得体会

此次实训将稀有的完整的实际工程项目拆解成模块进行教学和训练，让学生进入真实项目中去实践和锻炼；　我学习了大数据分析的一般流程：农业大数据综合实训平台中，通过对脱敏的动物检疫数据进行处理、分析和可视化，并针对特定疫病建立疫病检测模型，从而实现对进口动物疫病的智能检测。　从第一周的tensorflow的基础学习、爬虫小项目再到第二周的完整项目，无论是理论部分还是实践部分，都教我学会了如何细致分析一个问题并进行建模求解。特别是过采样方法去优化模型。　这次实践从小步骤：将数据转换格式，学会统计次数，学习各个包去绘制图，学习多表连接；到大的模型架构搭建，优化模型评估效果。从每一个函数的功能学习到实践，感谢老师的悉心教导。通过此次实践，总体掌握了大数据项目规范的开发过程，提高了专业技能的综合应用能力。