



華東師範大學  
EAST CHINA NORMAL  
UNIVERSITY

## 软件工程与 GIS 开发作业说明文档

姓名：	钟立
学号：	10213901406
专业：	地理科学
年级：	21 级
日期：	20230702

# 作物适宜区分析

## 1 研究目的

作物的生长需要一定的气温、降水、海拔、坡度、坡向、距离沟谷距离等条件，作物也需通过一定路径销售到餐厅、超市、农贸市场等地。通过 **streamlit** 库进行交互界面分析，用户可上传气候数据文件，可选择作物适宜的年降水范围、年均气温范围、沟谷线 **buffer** 范围、海拔范围、坡向范围，多个范围进行交集，得到矢量适宜作物区域。可视化研究区的气温、降水、海拔随经纬度变化。找到适宜作物区域的中心点，可以假定作物、农产品销售，查询周边需要农作物的地址，与其进行轻量级 **API** 路径规划，得到合适的销售路径。还能通过机器学习，选择相关因子，判断输入的条件是否适宜作物。

## 2 研究区域及基础数据

本次研究区域在陕西省安康市宁陕县境内，经度范围为东经 **108.17-108.44**，纬度范围为北纬 **33.51-33.74**，实际数据区域为一个方形，所有的栅格、矢量数据都在这个方形内。

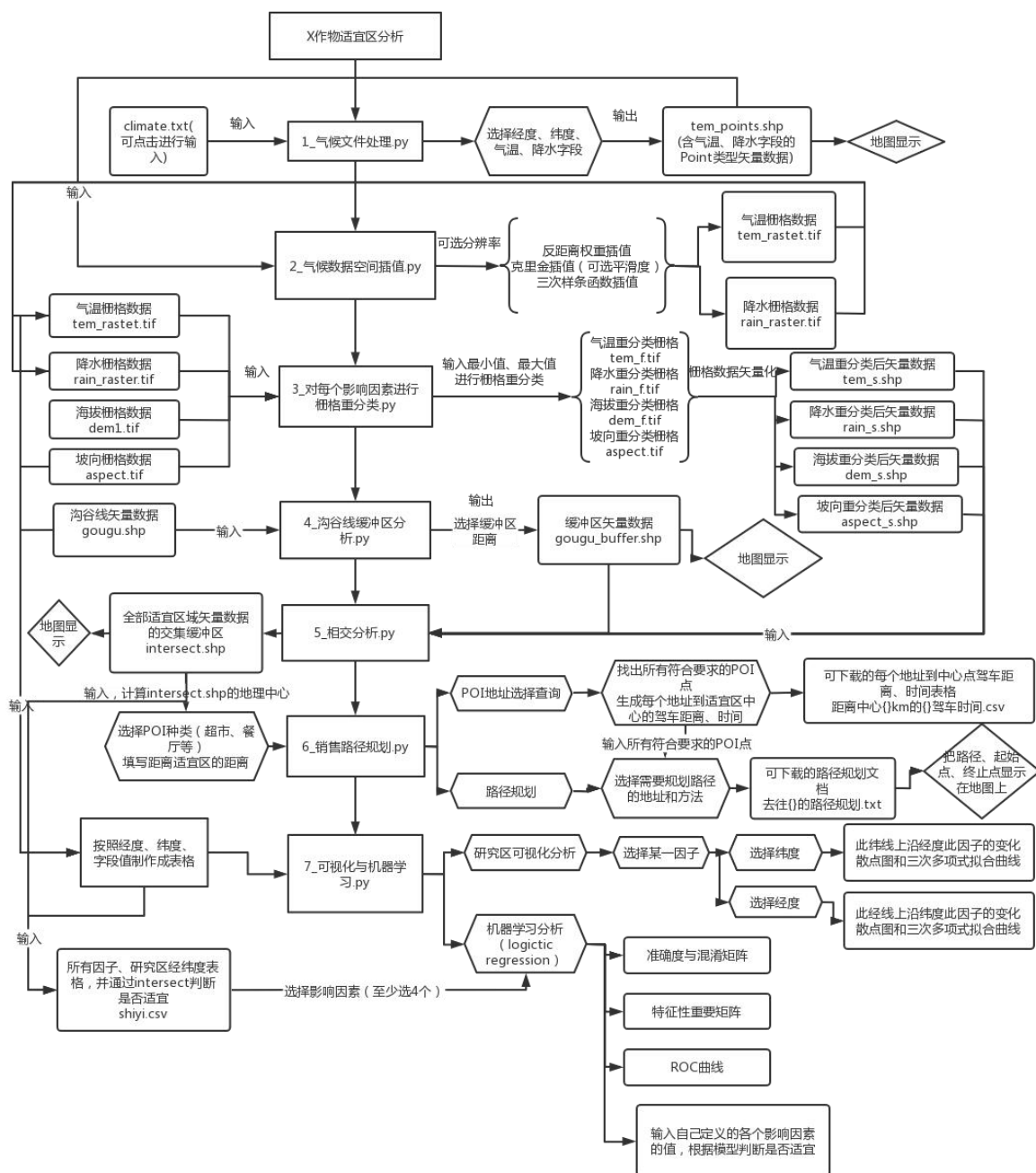
**txt 数据：**气候数据，包括监测点经度、纬度、年降水量、年均气温，可以在交互式页面中进行上传

**栅格数据：**dem 地形数据，研究区内坡向栅格数据

**aspect 坡向数据，**根据 dem 数据在 Arcgis 中获得

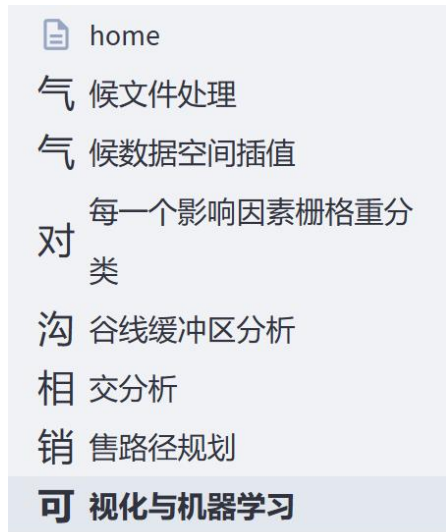
**矢量数据：**沟谷线数据，类型为 **polyline**

3 技术路线图（在 home 界面里也能呈现，并可以放大看），每个 part 的功能、输入输出都在技术路线图中：



#### 4 功能

本次研究的界面如下，操作时需要从上到下的顺序进行操作：



气候文件处理中可以交互式输入气候 `climate.txt` 数据，输出含有气温、降水字段的 `point` 类型 `shp` 数据，可在 `folium` 地图上显示。

气候数据空间插值中，可以对上面获取的 `shp` 数据进行空间插值（可选择分辨率）。有克里金插值（可以选择平滑度）、三次样条函数插值、反距离权重插值种插值方法。输入气温和降水的插值数据（`tif` 格式），插值的区域为研究区的方形区域。

下面对每一个影响因素进行栅格重分类。输入海拔、坡向、降水、气温的矢量数据，可选择合适区域的范围（最大栅格值和最小栅格值），先进行重分类，输出重分类栅格，再把栅格数据矢量化并输出。

接着对沟谷线进行缓冲区分析，输入沟谷线 `shp` 数据，选择 `buffer` 范围，输出沟谷线缓冲区矢量数据。

把上面得到的全部适宜因子矢量数据进行交集分析，输出交集 `shp` 文件，并显示在 `folium` 地图上。

进行销售路径规划。这里可以选择 `POI` 类型、距离缓冲区的距离（以适宜区中心为圆心画圆），得到所有此类型到适宜区的驾车距离、驾车时间的表格，此表格可以下载。可以选择其中一个地址和出行方法（驾车、步行、骑行），进行路径规划，对规划的路径进行整理，按照起点、终点、长度、耗时、转向提示等进行输出，并下载规划的 `txt` 文件，把规划好的路径显示在 `folium` 地图上，可点击上面的点和路线显示每一个路段提示。

进行可视化分析与机器学习。可以选择一个因子（气温、降水、地形、坡向），

选择经线，得到此经线上此因子随纬度变化的散点图和三次多项式拟合曲线。选择一条纬线，得到此纬线上此因子随经度变化的散点图和三次多项式拟合图。通过矢量适宜区数据，判断每一格点是否适宜。选择影响因子（经度、纬度、气温、降水、坡向、海拔中选，不少于四个），对研究区内的格点划分训练集和测试集，进行 **LogisticRegression** 机器分类学习，绘制模型的准确率、混淆矩阵、ROC 曲线、特征重要性柱状图，并可以自定义各个因子的值，判断此因子是否适宜。

5 每一个 py 的界面与具体实现

这里的所有文件打开都是用相对路径。需要先获得保存文件的路径，并每次打开文件时基于此路径打开。

```
import os

# 获取当前脚本的目录
current_dir = os.path.dirname(os.path.abspath(__file__))

ds = gdal.Open( os.path.join(current_dir, "dem1.tif") )
```

5.1 气候表格数据转化为 shapefile 点数据，并填充字段

输入气候的 txt 数据，输入之后用 dataframe 显示输入数据

### 上传气候数据文件，生成矢量点文件

上传气候文件

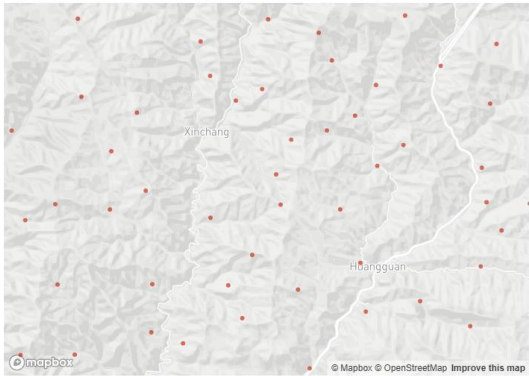
Drag and drop file here

Limit 200MB per file • TXT

Browse files

climate.txt 6.4 KB

X



	ID	X	Y	Temperatur	Rainfall
0	1	108.3587	33.7185	8.5500	662
1	2	108.4250	33.6831	9.9600	800
2	3	108.2093	33.5805	9.9800	614
3	4	108.3097	33.6668	10.6700	801
4	5	108.2898	33.6945	9.7800	596
5	6	108.2865	33.4916	12.0900	772
6	7	108.4565	33.6933	9.2600	575
7	8	108.2305	33.5042	9.8900	636
8	9	108.3685	33.6157	10.3700	719
9	10	108.2689	33.4759	11.8300	790

这里利用一个表单，下拉框内容随输入 **txt** 的表变化。这里可以选择字段为经度、纬度、气温、降水量数据，点击按钮，生成气候的矢量数据。先创立空白的矢量数据，把经度、纬度、气温、降水、坐标（WGS84）数据写入，生成之后，用一个 **folium** 显示生成的矢量数据。

选择经度

X

选择纬度

Y

选择气温

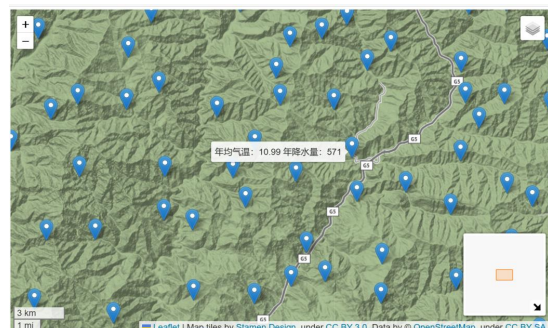
Temperatur

选择降水量

Rainfall

点击按钮，生成气候矢量点数据

由于稳定性原因这里 **streamlit** 可能显示不出来 **folium** 地图，所以截图贴过来。可以看到这里有两个可选的图层，分别为高德地图和地形图。鼠标经过点数据，可以显示这个点的经纬度数据。



## 5.2 矢量数据进行空间插值，转化为气温、降水栅格数据

由于这里的研究区域是一个方形区域，所以插值形成的栅格数据也是为一个与研究区一样的方形栅格数据。通过滑块，可以控制分辨率大小。由于插值涉及距离，所以这里添加 **web** 墨卡托的投影坐标系。

### 空间插值，生成降水栅格数据

请选择分辨率

80 500 100 500

选择插值方式

☒ 反距离权重插值法

☐ 克里金插值

☐ 三次样条函数插值

输出栅格文件

### 空间插值，生成气温栅格数据

请选择分辨率

100 500 100 500

选择气温插值方式

☒ 反距离权重插值法

☐ 克里金插值

☐ 三次样条函数插值

输出气温栅格文件

接下来可以选择插值类型，分别为反距离权重插值、克里金插值、三次样条函数插值。其中，克里金插值可以通过滑块选择插值的光滑度。

```
rbf = Rbf(x, y, values,function="inverse", epsilon=None, smooth=0)
interpolated_values = rbf(grid_x, grid_y)
```

（此函数控制插值方法）

创建输出栅格文件，设置新栅格的地理转换信息，设置新栅格地理坐标系，将插值结果写入栅格数据集。这里有一个进度条，可以显示栅格生成的进度。

```
progress_text = "Operation in progress. Please wait."
my_bar = st.progress(0)
text_element = st.empty()
percent_complete=0
```

```
percent_complete+=20
my_bar.progress(percent_complete)
text_element.text(f"进度: {percent_complete}%")
```

完成之后，点击 expand，可以显示完成的降水、气温栅格数据。

选择插值方式

- ☐ 反距离权重插值法
- ☒ 克里金插值
- ☐ 三次样条函数插值

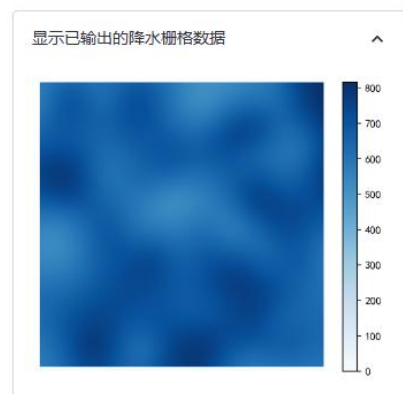
请选择平滑度，较小的值会导致插值函数更接近于输入数据，较大的值会使插值函数更平滑



输出栅格文件

进度：100%

栅格数据输出完毕

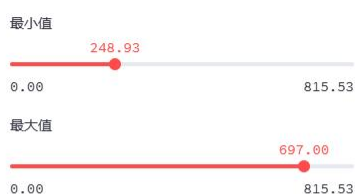


### 5.3 选择栅格数据合适的范围

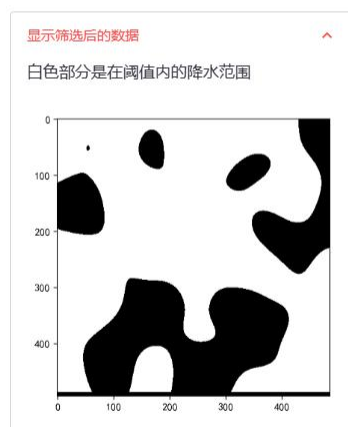


## 年降水量分析

请选择合适的年降水量阈值，系统将对坡向栅格数据进行重分类，输出选择范围的矢量数据



点击按钮，生成筛选范围的降水矢量数据



进度：100%

已输出筛选阈值完成的降水矢量数据！

## 海拔高度分析

请选择合适的海拔高度阈值，系统将对dem栅格数据进行重分类，输出选择范围的矢量数据



## 年降水量分析

请选择合适的年降水量阈值，系统将对坡向栅格数据进行重分类，输出选择范围的矢量数据



## 坡向分析

请选择合适的坡向阈值，系统将对坡向栅格数据进行重分类，输出选择范围的矢量数据



## 年均气温分析

请选择合适的气温阈值，系统将对气温栅格数据进行重分类，输出选择范围的矢量数据



这里可以通过滑块，选择新生成的气温、降水栅格数据和海拔、坡向栅格数据的最大值和最小值。由于数据量和内存的原因，这里的坡向数据范围不能超过140度。读取栅格数据的值，判断值是否落在选中的范围内，新建栅格数据把二值写入并设置坐标转换（WGS84）。把这几个栅格图像进行重分类，输出二值图，其中选中的为1，未选中的为0，并显示。

把这些栅格数据值为1的部分输出为矢量数据。创建内存中的矢量数据源，读取多边形要素，进行投影转化，并筛选栅格中值为1的部分，创建矢量数据的dataFrame对象，设置矢量数据的坐标参考信息和投影信息，将生成的矢量数据保存成文件。通过进度条呈现矢量数据输出的状态。实际操作中，由于数据量大，坡度矢量数据会输出得慢一些。

## 5.4 沟谷线缓冲区

这里通过滑块，可以选择缓冲区范围（单位：米），生成沟谷线缓冲区。这

里采用 `shapely` 和 `geopandas` 里自带的缓冲区生成函数。用一个 `folium`，把原始的沟谷线数据和生成的缓冲区显示在地图上。



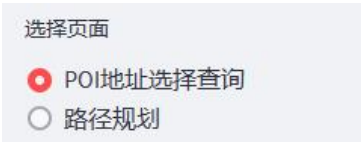
### 5.5 相交分析

输入所有筛选过的影响因素矢量数据（气温、降水、海拔、坡向、沟谷缓冲区），读取为 `geopandas` 格式，并统一坐标系（这里全部统一成 WGS84 坐标）。取其交集（`intersection`），并显示在 `folium` 中。

粉色区域即为得到的交集，即作物合适的适宜区。



### 5.6 销售路径规划



#### 5.6.1 POI 地址选择查询

选择需要农作物的场所和距离适宜区的范围

生产的农作物可以通过运输送到附近的餐厅、超市等地点。下面请选择合适的地址，规划轻量级路线。

选择需要农作物的POI种类

餐厅

填写距离x作物适宜区的距离，单位：km

0.50

已选择

选择需要农作物的POI种类

餐厅

超市

农贸市场

农田

农场

农业合作社

农业大学

可以选择需要农场品场所，即 POI 类型。这里可供选择的 POI 种类有餐厅、超市、农贸市场、农田、农场、农业合作社、农业大学等。这里还需填写距离 x 作物适宜区中心的距离（以作物适宜区中心为圆心画圆），输出所有此范围此 POI 类型的地址和经纬度，并计算到作物适宜区中心的驾车距离（单位 km）和驾车时间（单位：h），按照时间大小进行排列，用表格进行显示。此表格可以下载 csv 格式文件。

生产的农作物可以通过运输送到附近的餐厅、超市等地点。下面请选择合适的地址，规划轻量级路线。

选择需要农作物的POI种类

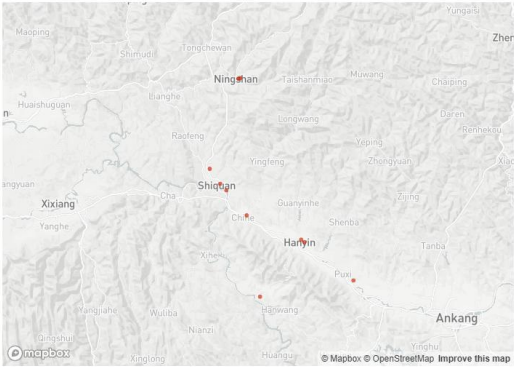
农贸市场

填写距离x作物适宜区的距离，单位：km

100.00

已选择

共检索到10个POI：



生成每个地址到中心点的距离和时间，根据时间大小排列

	名字	纬度	经度	和适宜区中心驾车距离	到适宜区驾车时间，单位：h
4	三昌农贸市场(迎宾大道店)	33.3160	108.3163	53.7550	0.8600
0	兴宁农贸市场	33.3178	108.3216	54.1680	0.8803
1	古堰农贸市场	33.0831	108.2273	77.0390	1.1314
5	江南农贸市场	33.0276	108.2788	91.5150	1.3161
2	桃园农贸市场	33.0444	108.2597	82.9480	1.3844
6	池河镇农贸市场	32.9627	108.3414	102.1570	1.4089
7	东关街菜市场	32.8934	108.5199	122.0370	1.6661
3	汉阴县城南农贸市场(汉太路)	32.8999	108.5100	123.2040	1.7414
8	双乳农贸市场(双乳菜市场)	32.7940	108.6716	135.3150	1.7811
9	渡湾镇农贸市场	32.7518	108.3829	139.5000	2.4625

下载csv文件

Download CSV

5.6.2 路径规划

上面已经选择了某种类型某范围的所有地址，这里输入这些地址的名称，可选择某一地址，进行路径规划。这里有三种路径规划方法，分别为驾车、骑行、步行。

## 路径规划

选择想要路径规划的地址

徐教授烤鱼馆

选择想要路径规划的方法

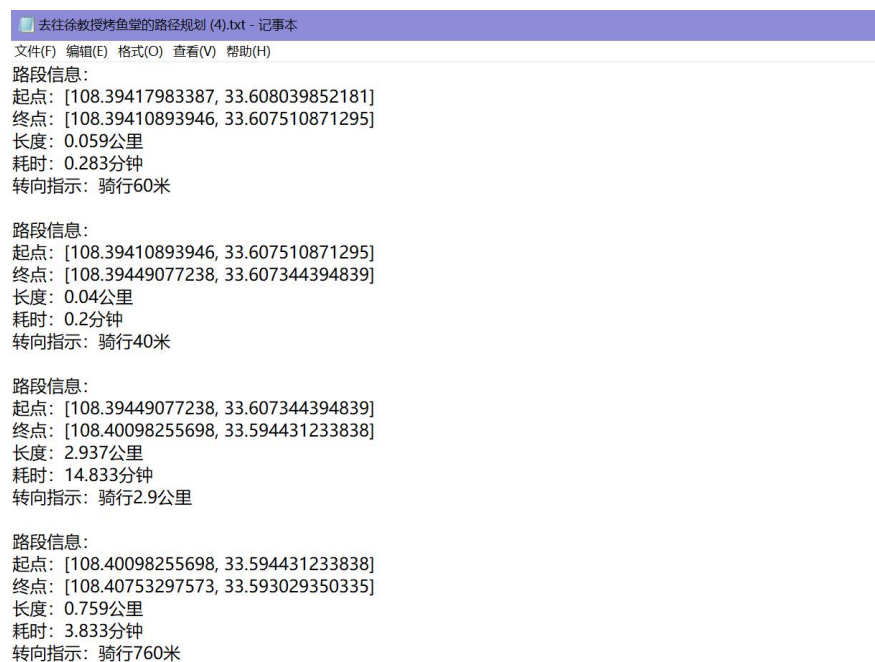
驾车

已选择,在地图中显示路径

下载路径规划的文档

Download txt

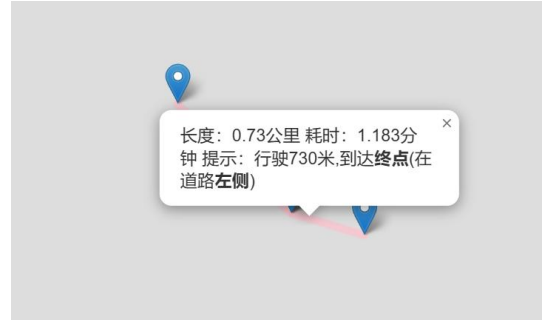
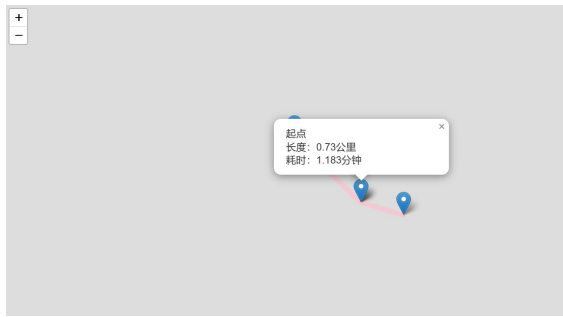
对规划的路径进行整理，按照起点、终点、长度、耗时、转向提示等进行输出。对路径规划的结果进行下载，为 txt 格式的文件。



对完成的路径在 folium 地图中进行显示。这里可以显示每一个路段的起始点、终止点，点击路段，可以显示此路段的信息。

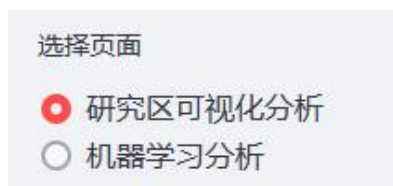
为了使提示框宽度更大，在 HTML 头部添加了一段 CSS 样式，将 leaflet-popup-content-wrapper 类的宽度设置为 300px。

```
#.get_root().header.add_child(folium.Element('<style>.leaflet-popup-content-wrapper { width: 300px !important; }</style>'))
for i in range(len(route_info)):
    start_location = route_info[i]['起点']
    end_location = route_info[i]['终点']
    length_km = route_info[i]['长度']
    duration_min = route_info[i]['耗时']
    instruction = route_info[i]['转向指示']
    # 添加起始点和终点标记，并设置 popup 属性
    folium.Marker(start_location[::1], popup=f'<div style="width: 300px;">起点<br>长度: {length_km:.2f} 公里<br>耗时: {duration_min} 分钟</div>').add_to(m)
    folium.Marker(end_location[::1], popup=f'<div style="width: 300px;">终点<br>长度: {length_km:.2f} 公里<br>耗时: {duration_min} 分钟</div>').add_to(m)
    # 添加路段线
    road_line = Polyline(
        locations=[start_location[::1], end_location[::1]],
        color='pink',
        weight=8,
        opacity=0.7,
        popup=f'<div style="width: 300px;">长度: {length_km:.2f} 公里<br>耗时: {duration_min} 分钟</div>').add_to(m)
    road_line.add_to(m)
output2 = st_folium(m)
st.write(output2) |
```



(这里地图实在显示不出来了)

## 5.7 可视化与机器学习



### 5.7.1 研究区可视化分析

#### 研究区可视化分析

请选择一个因子

- ☒ 气温
- ☐ 降水
- ☐ 海拔
- ☐ 坡向

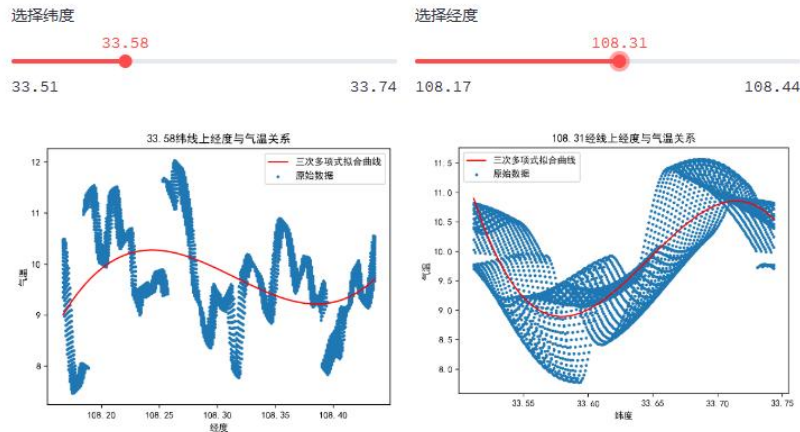
您选择的选项是: 气温

	Longitude	Latitude	value
0	108.1668	33.7432	9.72819995880127
1	108.1668	33.7422	9.767495155334473
2	108.1668	33.7412	9.803605079650879
3	108.1668	33.7403	9.836504936218262
4	108.1668	33.7393	9.86620044708252
5	108.1668	33.7384	9.892725944519043
6	108.1668	33.7374	9.916138648986816
7	108.1668	33.7364	9.936521530151367
8	108.1668	33.7355	9.953970909118652
9	108.1668	33.7345	9.968602180480957

程序中输入四个因子的栅格数据，这里可以选择一个影响因子，显示研究区每一个格点的值。

下面可以选择某一纬度，显示在这条纬线上经度随这一因子变化的散点图，并用三次多项式曲线进行拟合；选择某一经度，显示这条经线上研究区随此因子变化的散点图，同样用三次多项式曲线进行拟合。笔者也进行了实验，发现大于5次多项式会出现过拟合现象。





## 5.7.2 机器学习分析

由于这里机器学习前的处理数据太慢了，所以我先处理了气温在 8.5 到 10.5 摄氏度，降水在 600-700mm，海拔在 1000-2500m，坡向在 30-150 度，缓冲区在 1500m 范围内的数据，并以此为例进行机器学习分类分析，数据处理的代码已经保存成 jupyter 格式放在文件夹里。在此数据处理过程中，需要读取 intersect.shp 数据成 geopandas 格式，判断某一格点是否落在此 shp 数据中，以此判断此格点是否适合作物（主要是这一过程迭代次数过多，所以数据处理过程慢）。

```
import geopandas as gpd
from shapely.geometry import Point

# 假设 shapefile 文件名为 shapefile.shp
# 请将路径替换为您实际的文件路径
shapefile_path = 'intersect.shp'

# 读取面数据的 shapefile 文件
gdf = gpd.read_file(shapefile_path)

# 假设 dataframe 是包含经度 ('Longitude') 和纬度 ('Latitude') 字段的 DataFrame
df = ...

# 创建一个空的新字段 "适宜" 并初始化为 "不适宜"
df['适宜'] = '不适宜'

# 遍历 DataFrame 的经纬度
for index, row in df.iterrows():
    point = Point(row['Longitude'], row['Latitude'])
    for geometry in gdf['geometry']:
        if geometry.geom_type == 'Polygon':
            if point.within(geometry):
                dfk.at[index, '适宜'] = '适宜'
                break
        elif geometry.geom_type == 'MultiPolygon':
            for polygon in geometry.geoms:
                if point.within(polygon):
                    dfk.at[index, '适宜'] = '适宜'
                    break

# 打印修改后的 DataFrame
print(dfk)
```

	Longitude	Latitude	tem	rain	dem	aspect	适宜
0	108.17	33.46	10.403170	713.741940	1950.103035	176.995117	不适宜
1	108.17	33.47	10.333575	767.954903	1947.888633	168.402512	不适宜
2	108.17	33.48	8.691312	833.169150	1967.152542	170.620619	不适宜
3	108.17	33.49	7.801497	795.112190	1988.559431	175.550107	不适宜
4	108.17	33.50	7.491377	657.651438	2006.862517	175.836746	适宜
...	...	...	...	...	...	...	...
952	108.49	33.70	8.894662	737.263357	1740.224378	185.920868	不适宜
953	108.49	33.71	8.185500	745.317328	1733.225532	180.281189	不适宜
954	108.49	33.72	8.597057	722.921574	1709.274106	175.190643	不适宜
955	108.49	33.73	10.207662	706.886827	1687.487257	174.740479	适宜
956	108.49	33.74	10.411073	701.620560	1656.807903	175.643906	不适宜

	Unns	Longitude	Latitude	tem	rain	dem	aspect	适宜
0	0	108.1700	33.4600	10.4032	713.7419	1,950.1030	176.9951	不适宜
1	1	108.1700	33.4700	10.3336	767.9549	1,947.8886	168.4025	不适宜
2	2	108.1700	33.4800	8.6913	833.1692	1,967.1525	170.6208	不适宜
3	3	108.1700	33.4900	7.8015	795.1122	1,988.5594	175.5561	不适宜
4	4	108.1700	33.5000	7.4914	657.6514	2,006.8625	175.8368	适宜
5	5	108.1700	33.5100	7.3876	524.6804	2,026.6362	172.1650	不适宜
6	6	108.1700	33.5200	7.5883	470.5265	2,046.8593	169.2102	不适宜
7	7	108.1700	33.5300	8.0890	471.8552	2,067.4692	170.9148	不适宜
8	8	108.1700	33.5400	8.9077	490.2445	2,083.3469	170.0225	不适宜
9	9	108.1700	33.5500	9.8140	502.8299	2,085.6419	175.5492	不适宜

这里可以用复选框选择影响因素，至少要选择四个影响因素，构建模型 Logistic Regression，用合适的方法划分训练集和测试集（测试集 20%），设置随机森林的深度，训练模型并计算模型准确率。

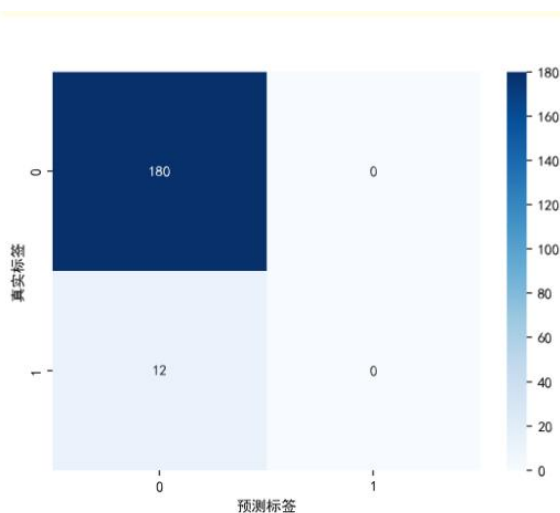
```
X_scaled = scaler.fit_transform(X)
X_train, X_val, y_train, y_val = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
accuracy = model.score(X_val, y_val)
```

选择影响因素

Longitude ×
Latitude ×
tem ×
dem ×
aspect ×
rain ×

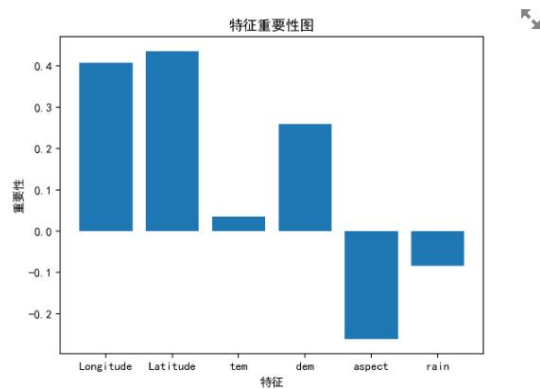
模型准确率: 0.9375

显示混淆矩阵，0 代表不适宜，1 代表适宜。



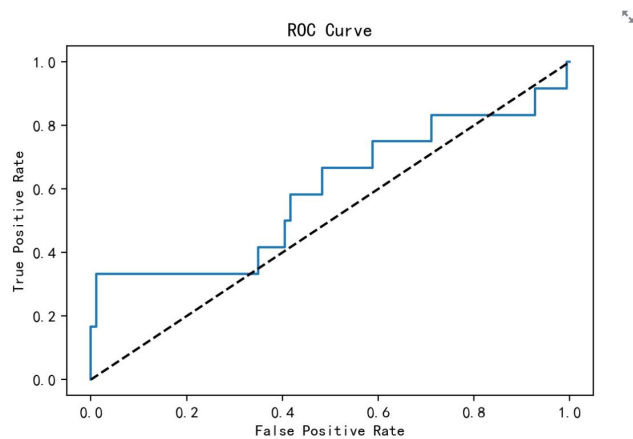
从混淆矩阵中可以看出，预测为不适合但是真实为适合的测试样本数为 12，预测为不适合并且预测正确的样本数为 180。

绘制特征重要性柱状图。



从这张图中可以看出，经度、纬度、地形对是否是适宜区的影响大，气温相对较小，降水与坡向与结果呈负相关。

由于此模型为二分模型，所以可以生成 ROC 曲线进行模型评估。0 为不适合种植，1 为适合种植。



`y_pred_proba` 返回了预测样本属于适合的概率。横轴为 FPR（假阳性率，实际不适合但是筛选为适合），纵轴为 TPR（召回率）。对角线表示随机猜测的情况，ROC 曲线越靠近左上角，模型的性能越好。

根据选中的影响因素，生成表单，可以自定义每个影响因子的值（值不可以超过此研究区内的最大值、最小值）。模型可以判断这些输入是否处于适宜区。

请填写以下影响因素，系统将判断是否处于适宜区：

Longitude	108.25	-	+
Latitude	33.51	-	+
tem	7.46	-	+
rain	470.66	-	+
dem	1246.29	-	+
aspect	150.85	-	+

提交

该点是否适宜:

0
0 不适宜