基于单日数据预测明日是否下雨

项目的视频及数据保存在百度网盘链接链接：
https://pan.baidu.com/s/1uloxjU-B9wjfgyx_pln6aQ?pwd=8888
提取码：8888

一. 选题原因及研究意义

选题的首要原因是由于在下被几次糟糕的天气预报所害，因此下定决心发挥数据人的精神，造一个自己的天气预报模型来预测第二天是否会下雨。

除了上面这个原因，本次研究还有两个主要的意义，第一点是对于现在我们这些待在城市里的大学生来说，缺乏实用的生活经验。对于以前的农村劳动者来说，仅凭生活经验就能八九不离十地预测第二天的天气情况了，本次实验的数据探索阶段的一部分工作旨在弥补这部分的空缺。第二点就是，本次实验证明了纯数据分析的力量是可观的，仅仅是凭单日数据进行纯数据上的预测，而不进行任何物理上的分析，就可以达到可观的准确率。

二. 数据集介绍及数据预处理

Date	Location	MinTemp	MaxTemp	Rainfall	Evaporatic	Sunshine	WindGustI	WindGustS	WindDir9a	WindDir3p	WindSpee	WindSpee	Humidity9	Humidity3	Pressure9a	Pressure3p	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow
2008/12/1	Albury	13.4	22.9	0.6	NA	NA	W	44	W	WNW	20	24	71	22	1007.7	1007.1	8	NA	16.9	21.8	No	No
2008/12/2	Albury	7.4	25.1	0	NA	NA	WNW	44	NNW	WSW	4	22	44	25	1010.6	1007.8	NA	NA	17.2	24.3	No	No
2008/12/3	Albury	12.9	25.7	0	NA	NA	WSW	46	W	WSW	19	26	38	30	1007.6	1008.7	NA	2	21	23.2	No	No
2008/12/4	Albury	9.2	28	0	NA	NA	NE	24	SE	E	11	9	45	16	1017.6	1012.8	NA	NA	18.1	26.5	No	No
2008/12/5	Albury	17.5	32.3	1	NA	NA	W	41	ENE	NW	7	20	82	33	1010.8	1006	7	8	17.8	29.7	No	No
2008/12/6	Albury	14.6	29.7	0.2	NA	NA	WNW	56	W	W	19	24	55	23	1009.2	1005.4	NA	NA	20.6	28.9	No	No
2008/12/7	Albury	14.3	25	0	NA	NA	W	50	SW	W	20	24	49	19	1009.6	1008.2	1	NA	18.1	24.6	No	No
2008/12/8	Albury	7.7	26.7	0	NA	NA	W	35	SSE	W	6	17	48	19	1013.4	1010.1	NA	NA	16.3	25.5	No	No

本次研究的数据集包含了 140000 条澳洲的天气和降雨数据，涵盖 2008-2016 年若干个地区，包含字段：

Date,Location,MinTemp,MaxTemp,Rainfall,Evaporation,Sunshine,WindGustDir,WindGustSpeed,WindDir9am,WindDir3pm,WindSpeed9am,WindSpeed3pm,Humidity9am,Humidity3pm,Pressure9am,Pressure3pm,Cloud9am,Cloud3pm,Temp9am,Temp3pm,RainToday,RainTomorrow，数据字段数量繁多种类复杂，绝对值跨度大，并且包含 string float int 等类型，特别是数据缺失严重，有两个字段的缺失率在 40%以上，还有很多在 10%左右，所以如果要保留较多数据，还要不缺准确性，数据预处理的难度比较大。

但是数据预处理对于本次研究非常重要，经过测试，直接使用去除缺失值的数据构造模型，和数据预处理后构造的模型准确率相差了 8%~9%。

本次研究的数据预处理大致分为三个阶段，下面简要介绍各个阶段的方法和目的：

2.1 数据预处理第一阶段

第一阶段做了以下工作：

1. 解析‘Date’字段 并把 YES NO 转化为 1/0
2. 把表示方向的字段转化为向量

```
# 首先把方向数据转化为单位向量，把YES转化为1，NO转化为0，方便计算使用
wind_dir_to_vector = {
    'N': (0, 1),
    'NNE': (math.cos(math.radians(22.5)), math.sin(math.radians(22.5))),
    'NE': (math.cos(math.radians(45)), math.sin(math.radians(45))),
    'ENE': (math.cos(math.radians(67.5)), math.sin(math.radians(67.5))),
    'E': (1, 0),
    'ESE': (math.cos(math.radians(112.5)), math.sin(math.radians(112.5))),
    'SE': (math.cos(math.radians(135)), math.sin(math.radians(135))),
    'SSE': (math.cos(math.radians(157.5)), math.sin(math.radians(157.5))),
    'S': (0, -1),
    'SSW': (math.cos(math.radians(202.5)), math.sin(math.radians(202.5))),
    'SW': (math.cos(math.radians(225)), math.sin(math.radians(225))),
    'WSW': (math.cos(math.radians(247.5)), math.sin(math.radians(247.5))),
    'W': (-1, 0),
    'WNW': (math.cos(math.radians(292.5)), math.sin(math.radians(292.5))),
    'NW': (math.cos(math.radians(315)), math.sin(math.radians(315))),
    'NNW': (math.cos(math.radians(337.5)), math.sin(math.radians(337.5))),
    'NA': (None, None)
}
```

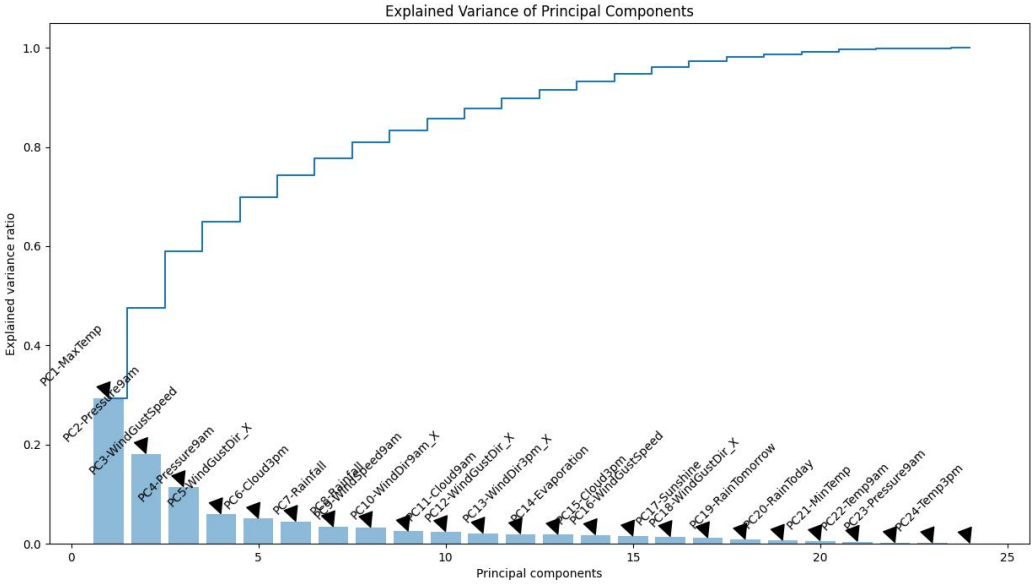
3. 数据标准化，避免大绝对值数据对后续分析造成主导性影响

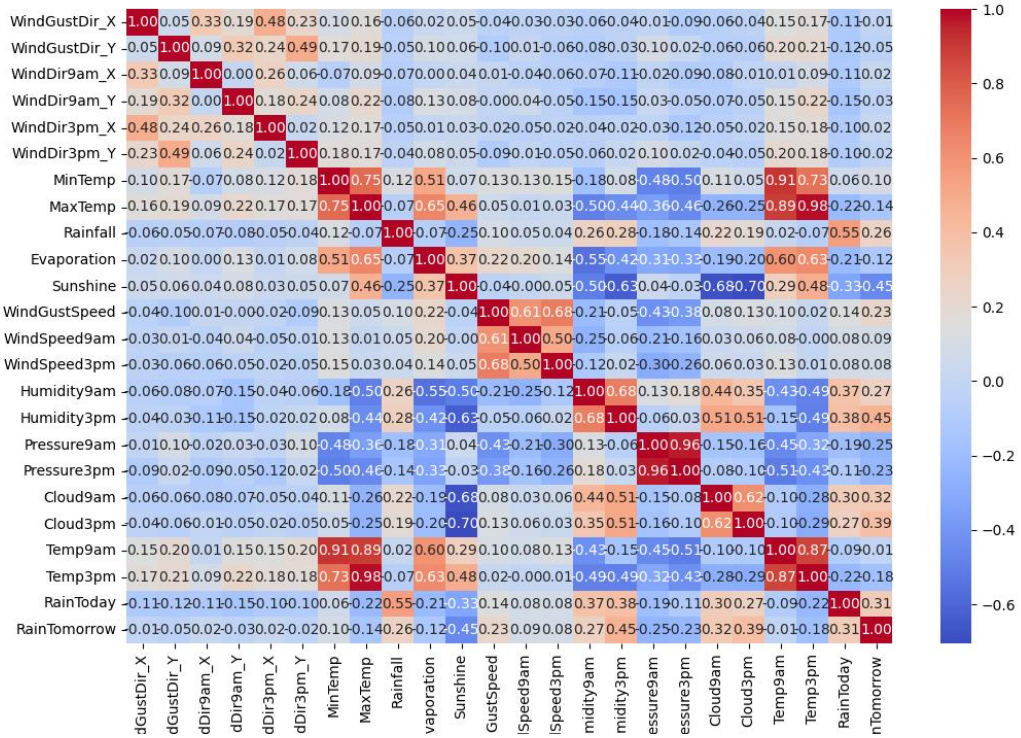
4. 对于缺失不严重的字段（15%以内），使用热卡填充（利用最接近的一行数据来填充缺失值），这一步定义了相对距离函数，从而在同地区的不同年份的同一个月份的几百条数据中找到最相似的数据

```
def find_similar_record(base_record, comparison_df, features):
    """
    在comparison_df中找到与base_record在特定特征上最相似的记录
    """
    valid_comparison_full = comparison_df.dropna()
    valid_comparison = valid_comparison_full[features]
    if valid_comparison.empty:
        return None
    # 处理base_record中的NaN值
    base_record_filled = base_record[features].fillna(valid_comparison[features].mean())
    try:
        distances = euclidean_distances( X: [base_record_filled.values], valid_comparison)
    except Exception as e:
        print("计算距离出错：", e)
        print(base_record_filled)
        return None
    min_index = np.argmin(distances)
    return valid_comparison_full.iloc[min_index]
```

2.2 数据预处理第二阶段

由于还是剩下很多缺失值，所以本阶段的主要目的是使用 PCA 技术和热力图，评估每个原始特征对每个主成分的贡献以及每个特征之间的相似度，评估完成后对于无关紧要的特征，直接删去这些特征，这样一来不用删掉太多数据行，这一阶段中，得到两张图：





通过分析上图，发现有关风向的三个特征在主成分载荷量中排名都很低，并且与第二天是否会下雨关系不大，因此大可以直接删除这三列数据。缺失最多的两个字段‘sunshine’和‘evaporation’中，‘sunshine’与要预测的结果有很强关联性，如果在 sunshine 字段用填充技术恐怕会造成很大影响，影响最终模型的性能，因此不得不把缺失 sunshine 行直接删除。而对于‘evaporation’来说，他对于结果有一定的影响，但不是决定性的影响，因此不必直接删除缺失的行，这样会损失太多的数据，直接去除这个特征也不妥，所以应该进行谨慎的填充。

2.3 数据预处理第三阶段

这一阶段简单地使用随机森林模型预测缺失的 evaporation 字段从而进行填充，根据测试，模型的预测性能良好。

```
均方误差 (MSE): 0.35766673136829963
```

三．构造衍生特征

数据预处理后，简单地构造几个衍生特征：当日温差（TempDiff）当日气压差（PressureDiff）当日湿度差（HumidityDiff）希望这些衍生特征在之后构建模型时可以产生作用

```
df['TempDiff'] = df['MaxTemp'] - df['MinTemp']
df['PressureDiff'] = df['Pressure3pm'] - df['Pressure9am']
df['HumidityDiff'] = df['Humidity3pm'] - df['Humidity9am']
```

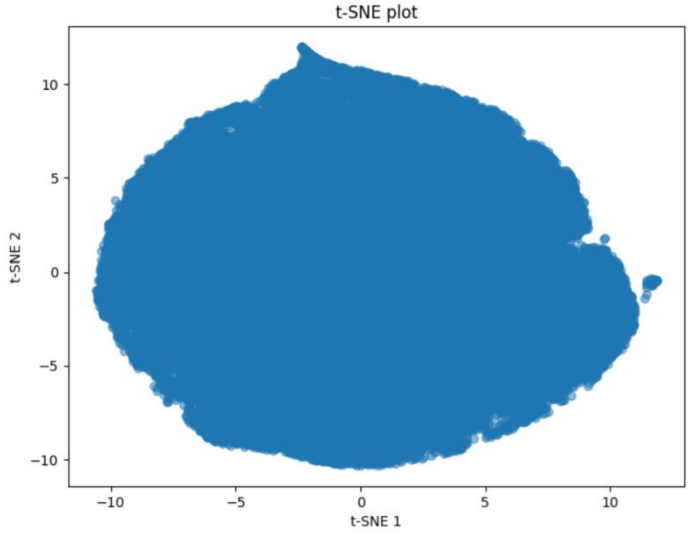
经过实验，这些衍生特征有三个功能，一是更好地揭示了特征之间的联系和特殊性，在数据探索中发挥了作用，二是包含了其他特征，因此可以在训练时删去被包含的特征，从而降低模型的复杂度和训练时间，三是对模型的性能有一定的帮助。

四．数据探索

数据探索主要分为两个阶段，第一个阶段是对于数据整体的分析，使用 T-sne 技术和热力图，分析数据整体的关联，第二个阶段是对于数据进行分组分析，发现了一些有趣的现象，在下面简要介绍：

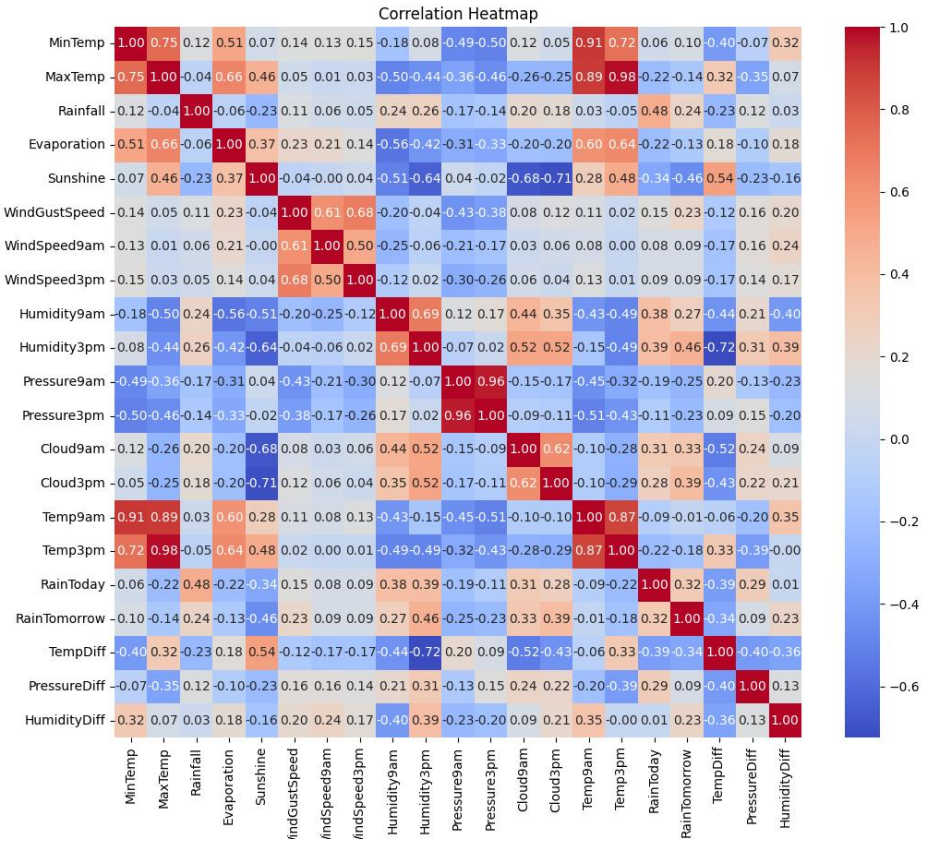
4.1 数据探索第一阶段

这一阶段首先使用 T-sne 技术进行高维数据的可视化，观察图像，发现 T-sne 分析的散点图均匀地形成一个近似圆形，但周围有一小部分离群的点集，这暗示数据没有明显的分组，但是依然存在一些小的特殊组，一定程度上可能说明天气的数据大体上是可预测的，但还是存在一些随机性。



其次使用热力图观察删除不重要特征并添加衍生特征后的特征之间的相互关系，此处特别关注一个衍生特征：温度差，对降水有很强的负面作用，差不多可以抵消云层量对于降水的正面作用，如果不构造衍生特征，这是不容易察觉到的，并且在生活中也不容易察觉。

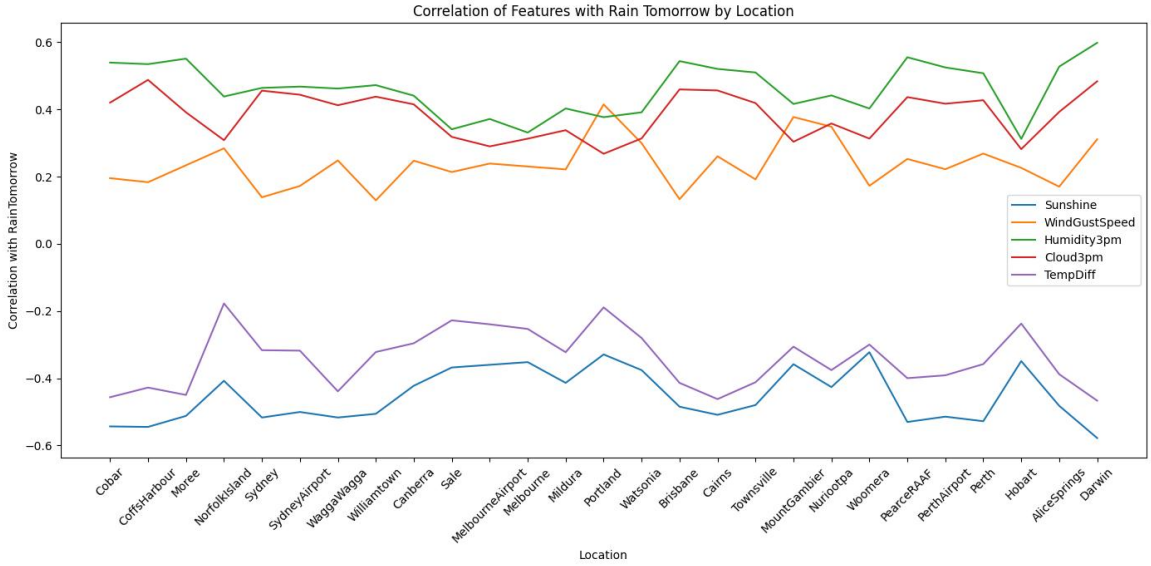
经过查阅一些资料，对此有如下解释：由于水的比热容较高，所以昼夜温差大可以说明空气中含水量少，因此第二天降水概率低。



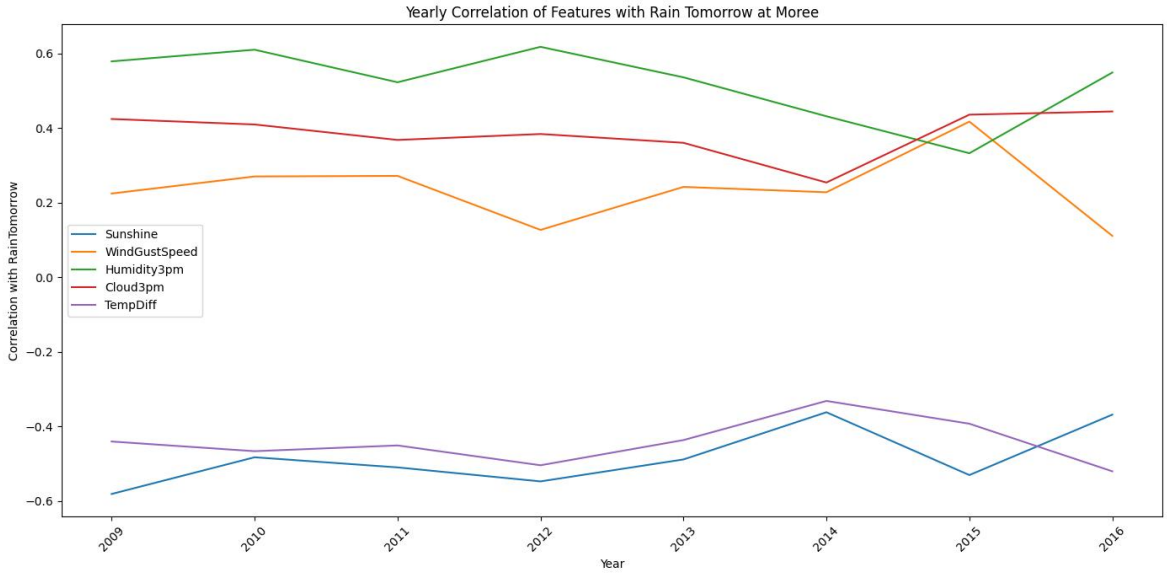
4.2 数据探索第二阶段

第二阶段根据上一阶段热力图提供的数据，选取 5 个感兴趣且和第二日降水密切相关的特征进行分地区分年份探索：Sunshine WindGustSpeed Humidity3pm Cloud3pm TempDiff

首先分地区计算每个地区这些特征与第二天是否下雨之间的相关性，此处一个有趣的结果是 阳光和温差同时与第二天是否降水负相关，并且对于一个地区来讲，这两个因素关于降水的相关性强弱同时增大，同时减小。同样的，湿度和云量同时正相关，并且同增同减。



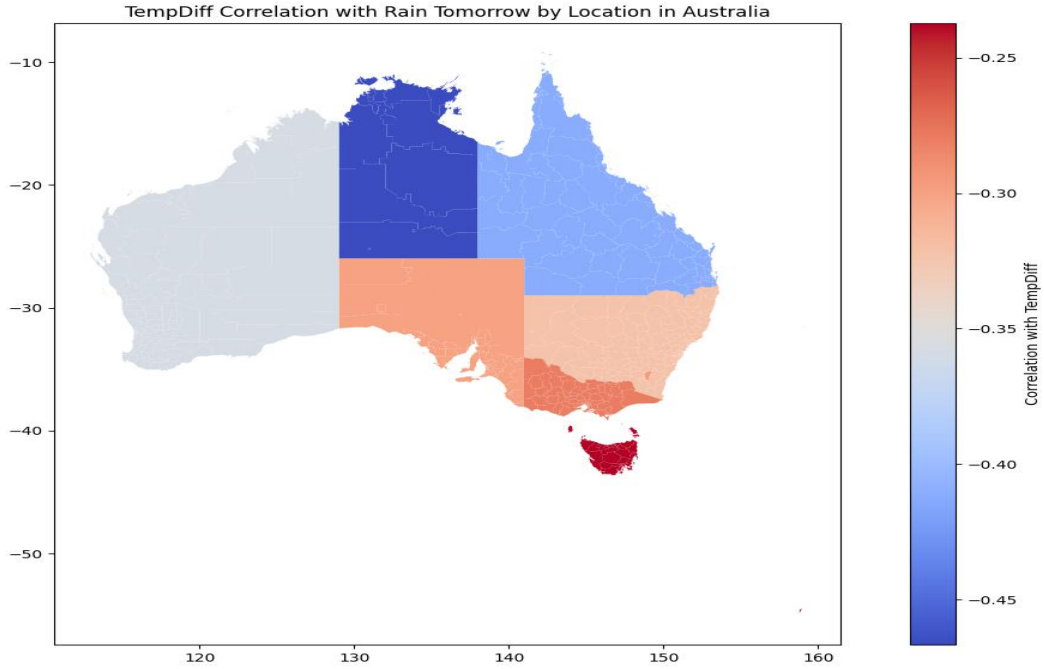
为了研究这两组同增同减关系是由于属性本身的物理关系决定的，还是由其他因素导致的，所以接下来考察一个地区不同年份这些特征与第二日是否下雨的相关系数：



根据图像，可以看见，cor（温差，第二日降水）与 cor（阳光，第二日降水）以及 cor（湿度，第二日降水）与 cor（云量，第二日降水）之间，本身并不天然存在联系，更大原因是由于地区的特性导致的。

为了了解地理位置是如何影响这些特征对降水的影响，选取 Tempdiff 特征对其可视化，可以看到，低纬度地区相关系数绝对值大，低纬度地区相关系数绝对值小，而处在中间的地区相关系数也处在中间（灰色，-0.35 左右），这可以充分说明问题：纬度越低，降水就越容易受到温差的影响。本人简单地翻了一下搜索引擎，这一点似乎很少有人提出。按理来说，

纬度低的地区由于白天受日照强烈，应该自然拥有更大的温差，从而相同的温差纬度低的地区应该对降雨的影响更小一些，而此处数据反映了相反的情况。这从某种程度上又一次说明了数据对反映客观规律的重要作用。



五．建模

建模阶段，选取了四个经典的模型：**逻辑回归 随机森林 支持向量机 神经网络**，以及一个新兴的模型：**XGBoost** 。经过反复调参和修改选取的特征，所有模型的准确率都可以达到 85%，但是都达不到 87%以上。尽管神经网络模型可以通过添加隐藏层层数和节点数，增加 batch 的大小和 epoch 的值，对于训练数据达到 97%以上的预测准确率，但这明显是过拟合的，这样严重过拟合的模型对于测试集只有 83%左右的预测准确率。

在建模阶段，我还尝试了一种**组合模型**（让逻辑回归 随机森林 支持向量机投票，少数服从多数）但是并没有准确率的提升，这表明多个模型可能预测错的条目都是差不多一样的。

下面简单地评价一下这些模型：

5.1 数据建模第一阶段（逻辑回归 随机森林 支持向量机 组合模型）

```
Logistic Regression Accuracy: 0.8528781793842035
              precision    recall  f1-score   support

         0.0       0.88      0.94      0.91      5783
         1.0       0.72      0.57      0.64      1687

    accuracy                           0.85      7470
   macro avg       0.80      0.75      0.77      7470
weighted avg       0.85      0.85      0.85      7470

Logistic Regression Coefficients: [[-0.12645412  0.02169965  0.0424357  -0.01179639 -0.57389511  0.76386453
  -0.075519   -0.19453838  0.41845153  0.7907173   0.1606403  -0.57372134
  -0.07392828  0.33558574  0.18935625 -0.00173183  0.53595015  0.14815377
  -0.73436164  0.37226577]]
Logistic Regression Intercept: [-1.96003441]
Random Forest Accuracy: 0.8649263721552878
              precision    recall  f1-score   support

         0.0       0.88      0.95      0.92      5783
         1.0       0.77      0.57      0.66      1687

    accuracy                           0.86      7470
   macro avg       0.83      0.76      0.79      7470
weighted avg       0.86      0.86      0.86      7470

Random Forest Number of Trees: 100
Random Forest Parameters: {'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt'
```

```
SVM Accuracy: 0.8617135207496653
              precision    recall  f1-score   support

         0.0       0.87      0.96      0.91      5783
         1.0       0.79      0.53      0.63      1687

    accuracy                           0.86      7470
   macro avg       0.83      0.74      0.77      7470
weighted avg       0.86      0.86      0.85      7470

Random Forest Number of Trees: 100
Random Forest Parameters: {'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt'
```

可以发现，这三个模型对于降雨的预测准确率都比较低，而且召回率尤其低，因此考虑模型可能倾向于预测不下雨，因此尝试调整阈值：

```
new_threshold = 0.47
y_pred_lr_new = (y_pred_proba_lr >= new_threshold).astype(int)
print("Logistic Regression Accuracy with New Threshold:", accuracy_score(y_test, y_pred_lr_new))
```

遗憾的是，随着阈值的降低，下雨的召回率虽然有一定提高，但是 precision 也在下降，因此准确率没有明显的提升。

既然单个模型效果并非完美，我尝试使用一种让三个模型进行投票，少数服从多数的组合模型来进行预测，评分如下图，效果没有提升，这意味着多个模型在同样的数据上产生了预测错误。

```
Combined Model Accuracy: 0.8605087014725569
              precision    recall  f1-score   support

         0.0       0.88      0.95      0.91      5783
         1.0       0.76      0.56      0.64      1687

    accuracy                           0.86      7470
   macro avg       0.82      0.75      0.78      7470
weighted avg       0.85      0.86      0.85      7470
```

5.2 数据建模第二阶段（神经网络与 XGBoost）

神经网络的评分：

```
              precision    recall  f1-score   support

     Class 0       0.88      0.95      0.91      4875
     Class 1       0.74      0.53      0.62      1350

    accuracy                           0.86      6225
   macro avg       0.81      0.74      0.77      6225
weighted avg       0.85      0.86      0.85      6225
```

可见，神经网络的各项性能和简单模型差不多。

XGBoost 的准确率：

```
Accuracy: 85.75%
```

最后尝试了 XGboost 模型，也没有明显的准确率提升。既然无论是传统模型还是神经网络，无论如何调参，都不能突破 87%的预测准确率，所以，可以猜测已经达到目前给出数据可以预测的瓶颈，剩下的不能预测的部分可能很大一部分来自于数据探索第一阶段 T-sne 技术所展示出的那些离散点。

六．总结与进一步研究的方向

本次实验在进行充分的数据预处理之后，在尽可能保证数据准确性的情况下避免 drop 大量的数据的前提下，创造了一些衍生特征，之后先进行了数据探索，后进行了数据建模。在数据探索中通过可视化和分组分析的方式，结合地理条件，得到了一些有趣的结论，在数据建模通过五种模型得到了性能相似的预测模型，我尝试多种方式进行优化，其中包括修改阈值，调整模型迭代方式，甚至还包括组合三种模型，这些方法都没有性能的明显改善。这些现象表明使用目前这些数据可能无法进一步提升性能。

我猜测，通过以下方式，可能可以进一步提升性能，但是由于时间上不再允许，并且与每次预测只使用单日数据的本意不符，所以目前没有进行，之后可能会进行：

1. 使用跨日期的衍生数据：比如本周前几天降雨的概率 本周前几天的平均温差
2. 把每个地区的经纬度等物理情况考虑在内，添加在数据里