面 试 测 评

Machine Learning and statistics (可以用中文或英文答题)

1. What is overfitting? How to prevent overfitting (in both traditional method and deep learning)?

Overfitting happens when models are more complicated than rules in data. When the model is overcomplicated (Decision tree with too many leaves, or linear regression with an excessively high level of degree), the model will capture not only knowledge but also noises in the data. The model performs well in the training dataset, but the performance drops significantly in the testing dataset, usually indicates overfitting happens.

Traditional methods to prevent overfitting includes adding regularization parameters, increasing the size and varieties of the training set, and reducing the complexity of models, etc.

For deep learning, one can reduce overfitting by random dropout weights during training, early stopping the training process before it overfits, or by reducing the complexity of the neural network.

2. Here is the confusion matrix:

		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Show how to compute precision and recall from these four values.

Precision = TP / (TP + FP)
Precision is the percentage of being correct for every prediction.

Recall = TP / (TP + FN)
Recall is the percentage of correctly identified cases out of all the positive cases.

设想这样一个量化交易的预测任务，我们希望每个交易信号都尽量盈利。在这个任务里，我们首先需要优化的precision还是recall？为什么，请提供理由。

We want to ensure every positive prediction (which means trade is profitable) is correct, which is to increase TP and reduce FP. Thus we need to focus on precision in this case.

设想这样一个风险预警任务。风险事件并不经常发生。我们十分害怕风险，希望躲过所有风险。我们需要每天提供预警信号。在这个任务里，我们首先需要优化的precision还是recall？为什么，请解释。

We want to identify all the positive cases in this scenario, that is to avoid any neglected positive cases. Thus we want to reduce FN and increase TP, so we need to focus on Recall in this case.

3. 假设你做了一个机器学习模型上线了。每天业务部门产生很多数据，然后你的算法把这些数据进行收集，并制作成特征，然后通过机器学习算法进行预测。但是最近一个月你突然发现你的模型效果变差了很多。请写出你计划从哪些方面，运用何种数据分析或其他办法进行诊断。

1. The problem may come from the process of collecting data. Perhaps the algorithm of collecting data goes wrong (Such as the change of database name, feature name, the website changed for scrapers, etc.), makes the feature for prediction contains more missing data than the usual. So we need to identify the percentage of missing data for each feature.
2. The distribution of the dataset changed. It may because of some changes in reality: Some incident happened changes the behaviours of customers, adjustment of policies changed the actions of markets, etc. Or because of other nuances introduced to the dataset: The scale of some features changed, etc. We need to inspect the distribution of the dataset over features, then conduct statistical tests between the data now and the past to identify the reason.

4. 美国就要大选了。假设公司做了一个民调的网站，来抽样统计一下目前民主党和共和党的支持比例。但是公司有人后来发现访问这个网站的人70%是年轻人，其他年龄段也都有，但是相对比较少。这个比例和真实人口比例不同。另外还发现西海岸的州人数偏多，其他州人相对少，这个比例也和真实人口比例非常不同。假设年龄和州这两个特征是决定民主党和共和党支持比例的最重要因素。请设计一个合理的统计方法，能够更准确的估计美国全国人口真实的支持比例。

按照美国真实的人口年龄分布和人口地域分布对网站统计的民调进行分层比例抽样。
根据实际人口年龄分布，将人口以六年为分段划分为不同18-24， 24-30， 30-36，36-42，42-48，48-54， 54-60，>60。以六年分段较符合人随着年龄变化发生的认知状况的改变，不过也可以适当调整年龄的划分方式。根据美国的实际年龄分布，在收集到的民调中加权抽样（可重复，避免某些年龄段的样本过少）得到和实际人口年龄分布更接近的样本。
同理，对收集到的民调根据实际的人口地域分布在得到的样本之上进行分层比例抽样，得到一个和美国实际年龄地域人口分布更接近的样本。（这种抽样方式假定人口年龄和地域是两个独立的特征，因此在抽样之前需要检验年龄和地域之间的分布关系是否独立，如果

二者之间相关则需要遍历两个特征之间的组合来进行分层比例抽样）。

算法和编程：

能答几题答几题，请不要上网搜答案。

1. 输入一个10进制数字，输出这个数8进制表示

例子：
输入： 10
输出： 12

```
def q1(input_num):
    output = ''
    while input_num > 8:
        output = str(input_num % 8) + output # 除八取余
        input_num = int(input_num / 8) #更新除数
    output = str(input_num) + output     # 除数小于8，余数自身，作为第一位
    return int(output)
```

```
In [8]: q1(1000000)
Out[8]: 3641100
```

2. 给定一个矩阵，找出从左上到右下角的一条路，使得这条路上数字和最大。这条路前进的方向只能向右或向下。输入的第一行是矩阵的行数和列数。输出第一行是一个序列，为该条路上的数字，第二行是这些数字的和。

例子
输入：
3 3
1 4 3
2 3 1
2 3 4
输出
1 4 3 3 4
15

```python
def q2():
    x = input().split(' ')

    rown = int(x[0])
    coln = int(x[1])
    matrix = []

    for i in range(rown):
        input_ = input().split(' ')
        input_ = [int(i) for i in input_]
        matrix.append(input_)
    output = [matrix[0][0]]
    row = 0
    col = 0
    for steps in range(1, rown + coln - 1):
        if row < rown - 1 and col < coln - 1:

            if matrix[row+1][col] > matrix[row][col+1]:
                row += 1
                output.append(matrix[row][col])
            else:
                col += 1
                output.append(matrix[row][col])
        else:
            if row == rown-1:
                col += 1
                output.append(matrix[row][col])
            elif col == coln-1:
                row += 1
                output.append(matrix[row][col])
            else:
                continue
    return sum(output), output
```

```
5 5

1 1 1 1 2

2 3 4 1 4

3 1 4 2 4

2 1 5 7 2

4 3 3 4 5
Out[30]: (35, [1, 2, 3, 4, 4, 5, 7, 4, 5])
```

3. 一个**自重复串**是一个字符串，其前一半和后一半是一样的，例如 abcdbabcdb （长度一定是偶数）。

输入一个字符串，找出其中最长的**自重复子串。这里的子串要求连续。**

例子
输入：
abababcdabcd
输出：

abcdabcd

4. 输入一个无向图邻接矩阵A（Aij=1代表i点和j点相连，0代表不相连）。输出这个图的联通分量的个数（一个联通分量就是一个子图，该子图每两个点间都可以有路径到达）。输入第一行是这个图点的个数。

例子

输入：

5
0 1 1 0 0
1 0 1 0 0
1 1 0 0 0
0 0 0 0 1
0 0 0 1 0

输出

2

（注：该图有两个联通分量，一个是{1,2,3}， 一个是{4,5}）