概述

题目

题目：

1. 分行业以30家A股上市公司为例，研究2007-2013年期 间25个财务指标对股价收益率的贡献（用季报数据）。
2. 选择前6大贡献的财务指标计算权重，编制财务指标强度指数，并与其2014的股价情况进行对比看是否与股价表现相符，并分析原因。

方案：
我们根据申万的行业分类，选取了生物医药行业的30只股票进行研究。
考察两个因素之间的关系

1. 上市公司的季度公告的财务数据，共25个指标
2. 在公告5个交易日内的股价涨跌情况

试图寻找这两个因素之间的关系。

我们选取的30只股票为：
【【30只股票的数据】】
同时，我们选取了25个财务指标：
【【25个财务指标】】
这些指标的含义是
【【25个财务指标的含义】】

方法

对于第一问，我们尝试通过线性回归进行研究。
对于第二问，我们尝试了多种方法进行预测，主要包括两种方法：

1. 线性回归
2. 决策树

线性回归

线性回归的解释

线性回归方程

线性回归：通过特征的线性组合来进行预测。

$$h(w) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$

$$h(w) = \mathbf{w}^T \mathbf{x}$$

其中w，x为矩阵。

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{bmatrix}$$

同时，我们定义一个损失函数，计作J(w)

$$J(\mathbf{w}) = \sum_{i=1}^{m} (h_w(x_i) - y_i)^2$$

其中

- y_i为第i个样本的真实值
- h_w(x_i)为第i个样本，在模型h_w下的预测值

线性回归系数的求解

我们要求解一组w，使得损失函数J(w)的值最小。
通常有两种方法：

1. 正规方程
2. 梯度下降

正规方程法

正规方程的公式

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

其中

- X是特征值矩阵
- y是目标值矩阵

梯度下降法

我们以只有一个变量的为例。
即，只有w_0和w_1两个参数。
则有

$$w_0 = -w_0 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_0}$$

$$w_1 = -w_1 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_1}$$

其中

- J(w_0,w_1)是损失函数
- \alpha是称为学习率

第一问

线性回归的实现

我们通过Python实现线性回归。
代码分为以下部分。

1. 读取和处理数据

2. 正规方程法求解
3. 梯度下降法求解

读取和处理数据

代码：

```
#%%

# 导入pandas包，以读取数据
import pandas as pd

# 读取 data.xlsx 这个文件，q1这个sheet页的数据
df_q1 = pd.read_excel('data.xlsx',sheet_name='q1')

# 把正跌幅数据作为y
y_train = df_q1['涨跌幅'].values
# 把其他25个财务指标作为x
x_train = df_q1.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅'],axis=1).values

# 打印财务指标
print(df_q1.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅'],axis=1).columns.values.tolist())
```

运行：

```
['销售毛利率', '总资产周转率', '息税前利润/营业总收入', '流动资产周转率', '营业总成本/营业总收入', '经营活动净收益/利润总额', '固定资产周转率', '经营活动产生的现金流净额/营业收入', '扣除非经常损益后的净利润/净利润', '营业外收支净额/利润总额', '总资产报酬率ROA', '扣除非经常性损益的ROE（扣除/摊薄）', '销售商品提供劳务收到的现金/营业收入', '应收账款周转率', '销售期间费用率', 'ROIC', '经营活动产生的现金流量净额/负债合计', '营业总收入(合并报表，亿元)', '净资产收益率（年化）', '存货周转率', '总资产净利润ROA', '净资产收益率（摊薄）', '归属母公司股东的权益/负债合计', '归属母公司股东的净利润-扣除非经常损益（同比增长率）', '总资产负债率']
```

正规方程法求解

代码：

```
#%%

# 导入正规方程的包
from sklearn.linear_model import LinearRegression

print('正规方程')
# 实例化正规方程
lr = LinearRegression()
# 训练模型
lr.fit(x_train,y_train)
# 打印模型的系数
print(lr.coef_)
# 打印模型的截距
print(lr.intercept_)
```

运行：

```
正规方程
[ 3.21523548e-04  1.24377559e-02  8.12393296e-05 -1.34994789e-02
  8.89782463e-04 -7.05145937e-06 -4.48405192e-04  1.00783342e-04
  9.56534823e-07 -7.34734904e-05 -2.71302628e-03  1.07909736e-04
  2.08276654e-05  6.54633857e-05 -4.41750652e-04 -2.43751108e-03
  1.52893275e-02  3.05942090e-05  4.97117594e-04  9.89231145e-04
  7.11994193e-03 -1.12882209e-03 -1.79840744e-03 -5.15626140e-06
  4.88520571e-04]
-0.09438018040824082
```

为了便于查看，我们整理如下：

1. 销售毛利率：3.21523548e-04
2. 总资产周转率：1.24377559e-02
3. 息税前利润/营业总收入：8.12393296e-05
4. 流动资产周转率：-1.34994789e-02
5. 营业总成本/营业总收入：8.89782463e-04
6. 经营活动净收益/利润总额：-7.05145937e-06
7. 固定资产周转率：-4.48405192e-04
8. 经营活动产生的现金流净额/营业收入：1.00783342e-04
9. 扣除非经常损益后的净利润/净利润：9.56534823e-07
10. 营业外收支净额/利润总额：-7.34734904e-05
11. 总资产报酬率ROA：-2.71302628e-03
12. 扣除非经常性损益的ROE（扣除/摊薄）：1.07909736e-04
13. 销售商品提供劳务收到的现金/营业收入：2.08276654e-05
14. 应收账款周转率：6.54633857e-05
15. 销售期间费用率：-4.41750652e-04
16. ROIC：-2.43751108e-03
17. 经营活动产生的现金流量净额/负债合计：1.52893275e-02
18. 营业总收入(合并报表，亿元)：3.05942090e-05
19. 净资产收益率（年化）：4.97117594e-04
20. 存货周转率：9.89231145e-04
21. 总资产净利润ROA：7.11994193e-03
22. 净资产收益率（摊薄）：-1.12882209e-03
23. 归属母公司股东的权益/负债合计：-1.79840744e-03
24. 归属母公司股东的净利润-扣除非经常损益（同比增长率）：-5.15626140e-06
25. 总资产负债率：4.88520571e-04
26. 截距：-0.09438018040824082

梯度下降法求解

代码：

```python
#%%

# 导入梯度下降的包
from sklearn.linear_model import SGDRegressor

print('梯度下降')
# 实例化梯度下降
sr = SGDRegressor()
# 训练模型
```

```
sr.fit(x_train,y_train)
# 打印模型的系数
print(sr.coef_)
# 打印模型的截距
print(sr.intercept_)
```

特别注意：由于梯度下降法自身的特点，每次求解均可能得到不同的解，不能说明每个财务指标的权重，运行结果略。

第二问

我们根据上一问线性回归方程，选择线性回归系数最大的六个财务指标重新进行分析。

用2014年的数据进行评估

在上一问中，得到的系数最大的六个财务指标为

1. 经营活动产生的现金流量净额/负债合计：1.52893275e-02
2. 流动资产周转率：-1.34994789e-02
3. 总资产周转率：1.24377559e-02
4. 总资产净利润ROA：7.11994193e-03
5. 总资产报酬率ROA：-2.71302628e-03
6. ROIC：-2.43751108e-03

同时，我们用2014年的数据对模型进行评估。

代码：

```
#%%

# 导入pandas包，以读取数据
import pandas as pd

# 读取 data.xlsx 这个文件，q1这个sheet页的数据
df_q1 = pd.read_excel('data.xlsx',sheet_name='q1')
df_q2 = pd.read_excel('data.xlsx',sheet_name='q2')

# 把正跌幅数据作为y
y_train = df_q1['涨跌幅'].values
# 把6个财务指标作为x
x_train = df_q1[['经营活动产生的现金流量净额/负债合计','流动资产周转率','总资产周转
率','总资产净利润ROA','总资产报酬率ROA','ROIC']].values

# 把正跌幅数据作为y
y_test = df_q2['涨跌幅'].values
# 把6个财务指标作为x
x_test = df_q2[['经营活动产生的现金流量净额/负债合计','流动资产周转率','总资产周转率','总
资产净利润ROA','总资产报酬率ROA','ROIC']].values

# 导入正规方程的包
from sklearn.linear_model import LinearRegression
print('正规方程')
# 实例化正规方程
lr = LinearRegression()
# 训练模型
lr.fit(x_train,y_train)
# 打印模型R2
```

```
print(lr.score(x_test,y_test))
# 打印模型的系数
print(lr.coef_)
# 打印模型的截距
print(lr.intercept_)
```

运行：

```
正规方程
-0.057865816933251546
[ 0.0045437  -0.00307429  0.00501173  0.00338773 -0.00237922 -0.00106071]
0.008468582578686226
```

解释：
上述R2的值为负数，是sklearn的部分特性导致的，我们可以理解为R2的值是0，即模型的效果非常不理想。

线性回归模型的调整

在上述的数据中，不同的财务指标的度量衡不同，比如`销售毛利率`和`息税前利润/营业总收入`的值都是-100到100的值（单位是%）。但是，对于`营业总收入(合并报表，亿元)`的值会超过100。这样会导致可能实际上`营业总收入(合并报表，亿元)`是权重很高，但是因为`营业总收入(合并报表，亿元)`度量衡的原因，导致其权重很低，影响我们的特征选取。
解决这个问题的方法是归一化

代码：

```
#%%

# 导入pandas包，以读取数据
import pandas as pd

# 读取 data.xlsx 这个文件，q1这个sheet页的数据
df_q1 = pd.read_excel('data.xlsx',sheet_name='q1')
df_q2 = pd.read_excel('data.xlsx',sheet_name='q2')

# 主要添加和修改了这部分的代码
# 主要添加和修改了这部分的代码
# 主要添加和修改了这部分的代码
df = pd.concat([df_q1,df_q2],axis=0)
from sklearn.preprocessing import MinMaxScaler
mm = MinMaxScaler()
mm.fit(df.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅'],axis=1).values)
# 把正跌幅数据作为y
y_train = df_q1['涨跌幅'].values
# 把6个财务指标作为x
x_train = df_q1.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅'],axis=1).values
x_train = mm.transform(x_train)

# 导入正规方程的包
from sklearn.linear_model import LinearRegression
print('正规方程')
```

```
# 实例化正规方程
lr = LinearRegression()
# 训练模型
lr.fit(x_train,y_train)
# 打印模型R2
print(lr.coef_)
```

运行：

```
正规方程
[ 0.02577715  0.03291155  0.02533681 -0.06389033  0.0626228  -0.07485842
 -0.03077499  0.01570608  0.01556237 -0.13778144 -0.16308978  0.01070896
  0.00241851  0.01997719 -0.03004152 -0.18054352  0.04669666  0.00673217
  0.06063806  0.01925796  0.42843823 -0.11353693 -0.02181666 -0.10771239
  0.04004559]
```

根据上述运行结果
我们新选取的特征为

1. 总资产净利润ROA
2. ROIC
3. 总资产报酬率ROA
4. 营业外收支净额/利润总额
5. 净资产收益率（摊薄）
6. 归属母公司股东的净利润-扣除非经常损益（同比增长率）

代码：

```
#%%

# 导入pandas包，以读取数据
import pandas as pd

# 读取 data.xlsx 这个文件，q1这个sheet页的数据
df_q1 = pd.read_excel('data.xlsx',sheet_name='q1')
df_q2 = pd.read_excel('data.xlsx',sheet_name='q2')

# 主要添加和修改了这部分的代码
# 主要添加和修改了这部分的代码
# 主要添加和修改了这部分的代码
df = pd.concat([df_q1,df_q2],axis=0)
from sklearn.preprocessing import MinMaxScaler
mm = MinMaxScaler()
mm.fit(df[['总资产净利润ROA','ROIC','总资产报酬率ROA','营业外收支净额/利润总额','净资产
收益率（摊薄）','归属母公司股东的净利润-扣除非经常损益（同比增长率）']].values)

# 把正跌幅数据作为y
y_train = df_q1['涨跌幅'].values
# 把6个财务指标作为x
x_train = df_q1[['总资产净利润ROA','ROIC','总资产报酬率ROA','营业外收支净额/利润总
额','净资产收益率（摊薄）','归属母公司股东的净利润-扣除非经常损益（同比增长率）']].values
x_train = mm.transform(x_train)


# 把正跌幅数据作为y
y_test = df_q2['涨跌幅'].values
```

```
# 把6个财务指标作为x
x_test = df_q2[['总资产净利润ROA','ROIC','总资产报酬率ROA','营业外收支净额/利润总
额','净资产收益率（摊薄）','归属母公司股东的净利润-扣除非经常损益（同比增长率）']].values
x_test = mm.transform(x_test)

# 导入正规方程的包
from sklearn.linear_model import LinearRegression
print('正规方程')
# 实例化正规方程
lr = LinearRegression()
# 训练模型
lr.fit(x_train,y_train)
# 打印模型R2
print(lr.coef_)
print(lr.score(x_test,y_test))
```

运行：

```
正规方程
[ 0.16963449 -0.06924679 -0.09888731 -0.10070491 -0.02462757 -0.07394035]
-0.06106685746345675
```

解释：
上述的模型实际上效果依然很不理想，甚至不如调整之前的效果

决策树和随机森林

除了用线性回归处理上次问题外，我们还尝试了非线性回归以及机器学习中的集成学习方法，结果均不够理想。
所以，我们考虑不预测股价的涨跌幅，转为预测股价的涨或跌。

数据处理

1. 对于涨跌幅大于0的股票，我们认为其属于上涨，标记涨跌字段为1。
2. 对于涨跌幅小于0的股票，我们认为其属于下跌，标记涨跌字段为-1。
3. 对于涨跌幅等于0的股票，我们认为其不涨不跌，标记涨跌字段为0

决策树

决策树的解释

决策树的思路起源于信息论。

信息熵

信息熵计作H，单位是比特

$$H(X)=\sum_{i\in X}P(i)*\log_2 P(i)$$

信息熵越大，不确定性也越大。

信息增益

特征A对训练数据集D的信息增益，计作g(D,A)。
其含义是训练数据集D的信息熵H(D)与给定A条件下信息熵H(D|A)的差。
公式为

$$g(D,A) = H(D) - H(D|A)$$

其中，H(D|A)的公式为

$$H(D|A) = \sum_{i-1}^{n} \frac{D_i}{D} * H(D_i)$$

$$H(D|A) = \sum_{i-1}^{n} \frac{D_i}{D} \sum_{i=1}^{K} \frac{D_{ik}}{D_i} * \log_2 \frac{D_{ik}}{D_i}$$

即，信息增益是指在已知特征X_1的信息，而使目标Y的信息不确定性的减少程度。

每次将信息增益最大的特征作为决策树的一个节点，这便是决策树的生成规则。

决策树的实现

代码：

```
#%%

# 导入pandas包，以读取数据
import pandas as pd
from sklearn.tree import DecisionTreeClassifier

# 读取 data.xlsx 这个文件，q1这个sheet页的数据
df_q1 = pd.read_excel('data2nd.xls',sheet_name='q1')
df_q2 = pd.read_excel('data2nd.xls',sheet_name='q2')

# 把正跌幅数据作为y
y_train = df_q1['涨跌'].values
# 把6个财务指标作为x
x_train = df_q1.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅','涨跌'],axis=1).values

# 把正跌幅数据作为y
y_test = df_q2['涨跌'].values
# 把6个财务指标作为x
x_test = df_q2.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅','涨跌'],axis=1).values

dec = DecisionTreeClassifier(random_state=1)

dec.fit(x_train,y_train)
print(dec.predict(x_test))
print(dec.score(x_test,y_test))
```

运行：

```
[ 0  1  1  1  0  1 -1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1  0  1 -1 -1
  1 -1 -1  1  1 -1 -1  1  0 -1 -1  1  1  1 -1 -1 -1  1  1 -1 -1  1  1  1
  1  1  1  1  1  1  0  1  1 -1 -1 -1 -1 -1  1 -1  1  1  1  1  0 -1 -1  1
 -1  1  1 -1 -1 -1 -1  1  1  1 -1 -1  1  1  1  1 -1  1  1 -1 -1  1 -1  1
  1 -1  1 -1  1  1 -1 -1  1 -1  1 -1  1 -1 -1 -1  1 -1 -1  1 -1 -1 -1 -1]
0.5
```

解释：
我们设置了三种结果：-1，0和1。最后用决策树预测涨跌的准确率是0.5。大于三分之一。
我们认为，该模型具有一定的准确率。

随机森林

随机森林的解释

多颗决策树的集合就是随机森林。随机森林的建立过程如下:

假设现在有N个样本，M个特征。

我们以一棵树的建立过程为例

1. 随机在N个样本中选择一个样本，重复N次。
    1. 随机有放回抽样，所以可能有重复样本。
2. 随机在M个特征中选出m个特征
    1. m < M

通过这种方法，循环往复，就可以得到森林了。

所以，多棵决策树，样本，特征都大多不一样，所以每棵决策树的分类结果都不太一样。

如果不进行随机抽样，那么每棵树的训练集都是一样的，最后分类结果当然也是一样的。那么投票会没有意义，一棵树和随机森林的结果会是一样的。

如果不是有放回的抽样，那么每棵树的训练集都是不一样的，没有交集。这样每棵树都是'有偏的'。每棵树的训练结果也都有很大的差异。那么会导致没有多数票，整个随机森林不能得到有效的分类结果。

随机森林的实现

代码：

```
# 导入pandas包，以读取数据
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV


# 读取 data.xlsx 这个文件，q1这个sheet页的数据
df_q1 = pd.read_excel('data2nd.xls',sheet_name='q1')
df_q2 = pd.read_excel('data2nd.xls',sheet_name='q2')

# 把正跌幅数据作为y
y_train = df_q1['涨跌'].values
# 把6个财务指标作为x
```

```
x_train = df_q1.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅','涨跌'],axis=1).values

# 把正跌幅数据作为y
y_test = df_q2['涨跌'].values
# 把6个财务指标作为x
x_test = df_q2.drop(['证券代码','证券名称','年度','季度','报告期','日期1（报告发布日）','股价1（报告发布日股价）','日期2（发布日后5日）','股价2（发布日后5日股价）','涨跌幅','涨跌'],axis=1).values

rf = RandomForestClassifier(random_state=1)
# 网格搜索与交叉验证

print('GridSearchCV   BEGIN')
gc = GridSearchCV(rf,param_grid={'n_estimators':
[120,200,300,500,800,1200],'max_depth':[2,4,8,16,32]},cv=10)
gc.fit(x_train,y_train)
print('GridSearchCV   END')

print('准确率有')
print(gc.score(x_test,y_test))

print('最佳模型')
print(gc.best_estimator_)
print(gc.best_score_)
```

运行：

```
GridSearchCV   BEGIN
GridSearchCV   END
准确率有
0.4166666666666667
最佳模型
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=2, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=500,
                       n_jobs=None, oob_score=False, random_state=1, verbose=0,
                       warm_start=False)
0.5142857142857142
```

结论

基于线性回归的财务指标强度指数

该指数用以比较涨跌幅，该指数的效果不太好。

1. 经营活动产生的现金流量净额/负债合计：0.0045437
2. 流动资产周转率：-0.00307429
3. 总资产周转率：0.00501173
4. 总资产净利润ROA：0.00338773
5. 总资产报酬率ROA：-0.00237922

6. ROIC：-0.00106071
7. 截距：0.008468582578686226

基于决策树的财务指标强度指数

该指数用以比较涨跌，该指数具有一定的准确性。