Problem C: Trading Strategies

说明

- 此作品为原创作品，一切的更新仅在：https://mianbaoduo.com/o/bread/mbd-Ypebl55s
- 代码语言python，版本python3.8
- 下面的代码是用来说明的，具体代码见代码py文件

  注意：此作品仅在JackPot，其他的均为盗版

数据处理

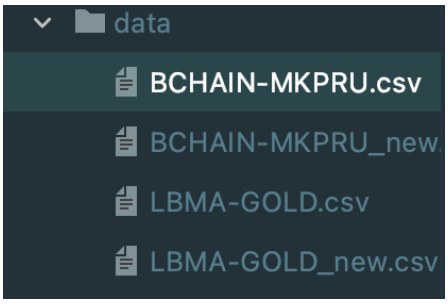
把2个csv原始数据放在data文件夹下：

- 首先我们知道，比特币可以每天交易，但黄金仅在开市日交易，所以黄金的csv文件里面肯定有些日期是空，所以先对空数据进行处理，
- 由于大概有10个空数据，所以不同的处理的方式对结果影响偏差不大，可以选择一下的处理方式之一即可
  - 用平均值填充
  - 删掉此行数据
  - 用此空数据行的上一行或者下一行填充
- 这边以用上一行的数据来填充为例，代码（第6行）如下：

```
1  def handle_gold(data_path):
2      data = pd.read_csv(data_path)
3      # 处理空缺值，设为平均
4      # data['USD (PM)'] = data['USD (PM)'].fillna(data['USD (PM)'].median())
5      # 用缺失值上面的值替换缺失值
6      data = data.fillna(axis=0, method='ffill')
7      data.to_csv('./data/LBMA-GOLD_new.csv')
```

- 如果用均值填充的，就是第4行，

对于比特币的也是如此，不过比特币每天都可以交易，所以没有空值，但是为了统一数据dateframe格式，这里一并处理了，具体见代码，处理后得到新的csv文件



问题一和问题二

建立模型

根据题意，我们可以把总的这个模型定义为：收益W = F(c, g, b)，当然模型的定义不唯一，这里仅提供一种可实现思路

从文件结构分析看，这是比较典型的回归模型组合问题

- 这里以多项式回归模型为例子

- 由题意：初始状态是[1000, 0, 0]，也就是说我们需要预测得到[C, G, B]的下一个状态或者某一个状态
- 我们先分别建立黄金和比特币这两个子模型，并求解第一个问题

黄金回归多项式模型

- 看代码

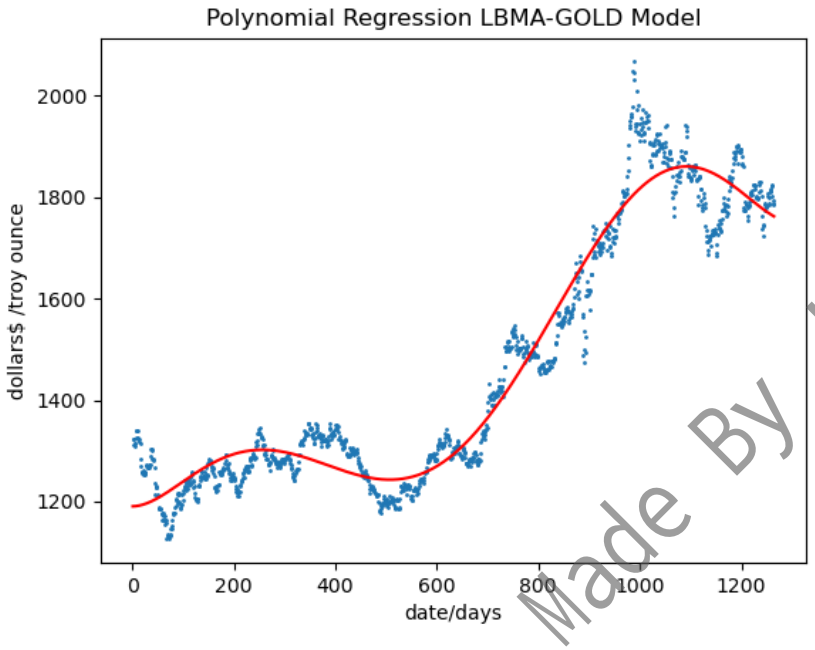
- 71行是处理数据的，如果你第一遍已经处理好，就可以把他注释掉了
- 72行展示数据量的
- 关键是得到38，39行，模型的系数和截距是我们要得到的模型参数，通过这些参数，我们就可以拟合预测出未来第n次交易的情况
- 你可以修改第31行，来调整模型，看哪个效果好几选哪个，注意degree不要选太大，太大的话容易过拟合，太小的话，就不准了


```
1  import pandas as pd
2  import numpy as np
3  from sklearn.linear_model import LinearRegression
4  from sklearn.preprocessing import PolynomialFeatures  # 生成多项式用的
5  import matplotlib.pyplot as plt
6
7
8  def handle_gold(data_path):
9      data = pd.read_csv(data_path)
10     # 处理空缺值，设为平均
11     # data['USD (PM)'] = data['USD (PM)'].fillna(data['USD (PM)'].median())
12     # 用缺失值上面的值替换缺失值
13     data = data.fillna(axis=0, method='ffill')
14     data.to_csv('./data/LBMA-GOLD_new.csv')
15
16
17 def show_gold(data_path):
18     data = pd.read_csv(data_path)
19     x_data = data.iloc[1:, 1]
20     y_data = data.iloc[1:, 2]
21     plt.scatter(x_data, y_data, s=1)
22     plt.show()
23
24 # 建立模型
25 def build_model(data_path):
26     data = pd.read_csv(data_path)
```

```
    x_data = data.iloc[1:, 0]
    y_data = data.iloc[1:, 2]
    # 转换为二维数据
    # degree = n, 相当于n次方拟合
    poly = PolynomialFeatures(degree=6)
    # 特征处理
    x_data = np.array(x_data).reshape((len(x_data), 1))
    x_poly = poly.fit_transform(x_data)
    model = LinearRegression()

    model.fit(x_poly, y_data)
    print('系数: ', model.coef_)
    print('截距: ', model.intercept_)

    # 画图
    plt.scatter(x_data, y_data, s=1)
    plt.plot(x_data, model.predict(x_poly), 'r')  # predict 传的是x_poly,是处理后的数
据
    plt.title('Polynomial Regression LBMA-GOLD Model')
    plt.xlabel('date/days')
    plt.ylabel('dollars$ /troy ounce')
    plt.show()

    return len(y_data), model.coef_[1:], model.intercept_


# 计算最初的 1000 美元投资价值
def prediction(init_money, count, coef: list, intercept, a=0.01):
    """
    :param init_money: 最初的投资
    :param days: 第几次交易
    :param coef: 模型系数
    :param intercept: 模型截距
    :param a: αgold = 1%
    :return: init_money在第days天的投资价值
    """
    res = intercept
    print('第几次交易: ', count)
    for i, c in enumerate(coef):
        res += c * pow(count, i + 1)
    print('每金衡盎司美元(dollars per troy ounce): ', res, '美元$')
    return init_money / res * (1 - a)


if __name__ == '__main__':
    # handle_gold('./data/LBMA-GOLD.csv')
    # show_gold('./data/LBMA-GOLD_new.csv')
    count, coef, intercept = build_model('./data/LBMA-GOLD_new.csv')
```

```
    print('2021 年 9 月 10 日,最初的 1000 美元投资价值:', prediction(1000, count, coef, intercept), '盎司')

```

- 结果




```
系数:  [ 0.00000000e+00  5.30147327e-05  8.40271443e-03 -4.23485209e-05
  7.68679724e-08 -5.74630762e-11  1.51678800e-14]
截距:  1190.3827674894403
第几次交易:  1264
每金衡盎司美元(dollars per troy ounce):  1762.235394677591 美元$
2021 年 9 月 10 日,最初的 1000 美元投资价值: 0.5617864690438391 盎司

Process finished with exit code 0
```

- 结果第5行，也就是2021 年 9 月 10 日是第1264次交易，我们由模型可以得到当天1盎司黄金 = 1762.235394677591 美元，由此可以衡量出1000美金的投资价值
- 模型公式从系数和截距得到，比如上面结果得到的如下

```
系数：a1,a2,a3
截距：b
```

那么模型就是`y = a1*x^0 + a1*x^1 + a2*x^2 + b`

上面黄金模型得到的是6次多项式模型，所以最高项也就是6次

比特币的模型也是类似

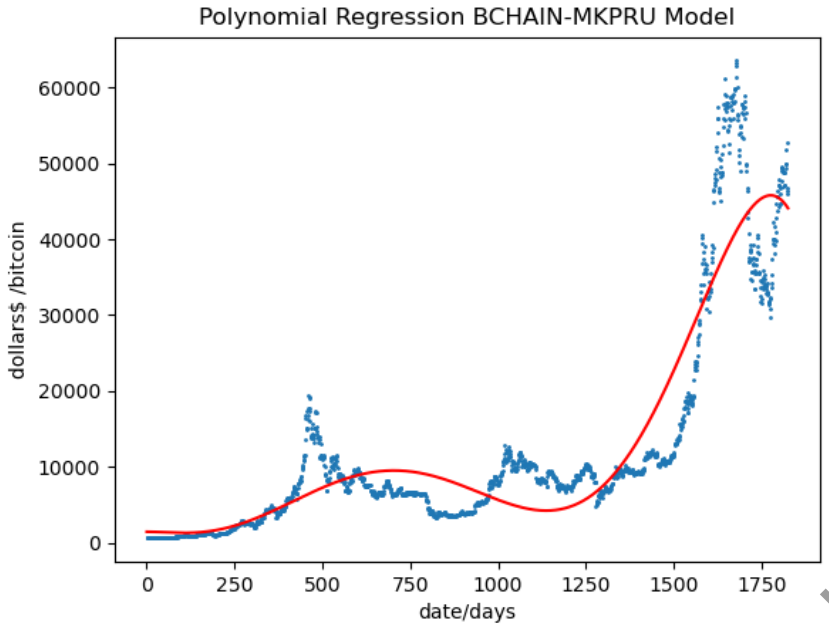
比特币多项式回归模型

- 需要改动的地方和黄金模型类似，不过个别地方需要注意，abitcoin = 2%这些

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures  # 生成多项式用的
import matplotlib.pyplot as plt


def handle_bitcoin(data_path):
    data = pd.read_csv(data_path)
    # 用缺失值上面的值替换缺失值
    data = data.fillna(axis=0, method='ffill')
    data.to_csv('./data/BCHAIN-MKPRU_new.csv')


def show_bitcoin(data_path):
    data = pd.read_csv(data_path)
    x_data = data.iloc[1:, 1]
    y_data = data.iloc[1:, 2]
    plt.scatter(x_data, y_data, s=1)
    plt.show()


def build_model(data_path):
    data = pd.read_csv(data_path)
    x_data = data.iloc[1:, 0]
    y_data = data.iloc[1:, 2]
    # 转换为二维数据
    # degree = n，相当于n次方拟合
    poly = PolynomialFeatures(degree=6)
    # 特征处理
    x_data = np.array(x_data).reshape((len(x_data), 1))
    x_poly = poly.fit_transform(x_data)
    model = LinearRegression()

    model.fit(x_poly, y_data)
    print('系数：', model.coef_)
    print('截距：', model.intercept_)

    # 画图
    plt.scatter(x_data, y_data, s=1)
    plt.plot(x_data, model.predict(x_poly), 'r')  # predict 传的是x_poly,是处理后的数据
    plt.title('Polynomial Regression BCHAIN-MKPRU Model')
    plt.xlabel('date/days')
    plt.ylabel('dollars$ /bitcoin')
```

```
    plt.show()

    return len(y_data), model.coef_[1:], model.intercept_


# 计算最初的 1000 美元投资价值
def prediction(init_money, count, coef: list, intercept, a=0.02):
    """
    :param init_money: 最初的投资
    :param count: 第几次交易
    :param coef: 模型系数
    :param intercept: 模型截距
    :param a: αbitcoin = 2%
    :return: init_money在第days天的投资价值
    """
    res = intercept
    print('第几次交易: ', count)
    for i, c in enumerate(coef):
        res += c * pow(count, 1 + i)
    print('比特币每日价格(dollars per bitcoin): ', res, '美元$')
    return init_money / res * (1 - a)


if __name__ == '__main__':
    # handle_bitcoin('./data/BCHAIN-MKPRU.csv')
    # show_bitcoin('./data/BCHAIN-MKPRU_new.csv')
    count, coef, intercept = build_model('./data/BCHAIN-MKPRU_new.csv')
    print('2021 年 9 月 10 日,最初的 1000 美元投资价值:', prediction(1000, count, coef,
intercept), '比特币')

```

- 结果




```
系数:  [ 0.00000000e+00 -5.52957703e-04 -4.65922448e-02  3.80701831e-04
 -6.85859284e-07  4.58595999e-10 -1.02636974e-13]
截距:  1433.8246533975998
第几次交易:  1825
比特币每日价格(dollars per bitcoin):  44117.196522225626 美元$
2021 年 9 月 10 日,最初的 1000 美元投资价值: 0.022213560181828185 比特币

Process finished with exit code 0
```


- 这里的分析类似黄金模型
- 最后得到，2021 年 9 月 10 日比特币每日价格(dollars per bitcoin)：44117.196522225626 美元$

求解

- 求解函数主要在prediction
- 所以我们得到：初始的[C, G, B]从[1000， 0， 0] 到 [1000, 1762.24, 44117.20]的结果
- 模型我们可以设为: 收益W = F(c, g, b)，而g和b就是我们上面得到的2个子模型，c就是当前的投资，限制c，g，b之间关系的就是策略条件，也就是哪些时候交易，交易价的最低点和最高点的时间点，以及汇率都有影响

策略

- 很明显，无论是比特币还是黄金，我们要交易的思路，肯定是在汇率最低的时候买入，在汇率最高的时候卖出，这样我们赚的才是最多的，思想有点类似股票模型，大家也可以去参考参考一下股票预测模型
- 接下来讨论W = F(c, g, b)
- 下面继续更新中，一切的更新仅在：https://mianbaoduo.com/o/bread/mbd-Ypebl55s

问题三

问题四