

在 Python 中使用 MPT 进行投资组合优化



掌握 Python、机器学习、NLP、深度学习、数据工程等等！

了解更多

家

 Parth Tyagi - 发布于 2021 年 4 月 27 日

[数据探索](#) [数据可视化](#) [中间的](#) [数学](#) [Python](#) [结构化数据](#)

[本文作为数据科学博客马拉松的一部分发表。](#)

介绍

在本文中，我们将学习投资组合管理的概念，并使用 Python 库来实现。

本文分为三个部分，以涵盖投资组合管理的基本方面，如下所示：

1. 资产和投资组合的回报
2. 与资产和投资组合相关的风险
3. 现代投资组合理论：寻找最优投资组合

资产和投资组合的回报

在深入研究资产和投资组合回报率率的计算之前，让我们简要了解一下投资组合和回报率率的定义。

投资组合：投资组合是金融工具的集合，如股票、债券、商品、现金和现金等价物，以及它们的基金对应物。[投资百科]

在本文中，我们的投资组合将包含 4 项资产（“以股票为重点的投资组合”）：苹果公司、耐克 (NKE)、谷歌和亚马逊的股票。

我们的数据预览如下所示：

Symbols	AAPL	NKE	GOOGL	AMZN
Date				
2015-01-02	24.861401	44.347309	529.549988	308.519989
2015-01-05	24.161013	43.633312	519.460022	302.190002
2015-01-06	24.163294	43.376633	506.640015	295.290009
2015-01-07	24.502111	44.272652	505.149994	298.420013
2015-01-08	25.443539	45.294640	506.910004	300.459991

回报：它指的是我们的资产/投资组合在固定时间范围内的收益或损失。在此分析中，我们将回报作为资产收盘价相对于前一天收盘价的百分比变化。我们将使用 `python` 中的 `pct_change()` 函数。

下面显示了执行相同操作的 `python` 代码以及返回的前 5 行（标题）

注意：如果在第一行之前不存在一行，则第一行为 `Null` 以方便计算百分比变化。

我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya，即表示您同意我们的 [隐私政](#)

[策](#)和[使用条款](#)。 [接受](#)

在 Python 中使用 MPT 进行投资组合优化

Date				
2015-01-02	NaN	NaN	NaN	NaN
2015-01-05	-0.028172	-0.016100	-0.019054	-0.020517
2015-01-06	0.000094	-0.005883	-0.024679	-0.022833
2015-01-07	0.014022	0.020657	-0.002941	0.010600
2015-01-08	0.038422	0.023084	0.003484	0.006836

投资组合的回报定义为投资组合中资产回报的加权总和。

为了演示如何在 python 中计算投资组合回报，让我们随机初始化权重（稍后我们将对其进行优化）。投资组合回报的计算如下面的代码所示，并且还显示了投资组合回报的头部：

```
weight_portfolio = [0.2, 0.4, 0.3, 0.1]
portfolio_return = returns_portfolio.dot(weight_portfolio)
portfolio_return.head()
```

```
Date
2015-01-02      NaN
2015-01-05  -0.019842
2015-01-06  -0.012021
2015-01-07   0.011245
2015-01-08   0.018647
dtype: float64
```

我们已经了解了如何计算回报，现在让我们将注意力转移到另一个概念上：[风险](#)

与资产和投资组合相关的风险

计算投资组合风险的方法如下所示，随后我们将解释并给出每个步骤的数学公式：

- 计算收益数据的协方差矩阵
- 通过乘以 252 年化协方差
- 通过将其与权重向量相乘来计算投资组合方差
- 计算上面计算的方差的平方根以获得标准偏差。这个标准偏差被称为投资组合的波动率

计算协方差并将其年化的公式为：

$$\text{协方差} = \text{returns.cov()} * 252$$

由于有 252 个交易日，我们将协方差乘以 252 以对其进行年化。

上述第 3 步计算的投资组合方差取决于投资组合中资产的权重，定义为：

$$\sigma_{portfolio}^2 = (\text{weights})^T \cdot \text{Cov}_{portfolio} \cdot (\text{weights})$$

在哪里，

- $\text{Cov}_{portfolio}$ 是投资组合的协方差矩阵
- 权重是分配给投资组合中每个资产的权重向量

下面展示了用 python 计算资产风险的方法。

我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya，即表示您同意我们的[隐私政策](#)和[使用条款](#)。 [接受](#)

在 Python 中使用 MPT 进行投资组合优化

Symbols	AAPL	NKE	GOOGL	AMZN
AAPL	0.061690	0.020972	0.030726	0.035782
NKE	0.020972	0.055628	0.021395	0.023106
GOOGL	0.030726	0.021395	0.056529	0.044587
AMZN	0.035782	0.023106	0.044587	0.085557

`variance_matrix` 的对角线元素表示每个资产的方差，而非对角线项表示两个资产之间的协方差，例如：(1,2) 元素表示耐克和苹果之间的协方差。

我们在上面看到的方法在python中计算投资组合风险的代码如下所示：

```
# Computing portfolio variance
portfolio_variance = np.transpose(weight_portfolio)@var_matrix@weight_portfolio

# Computing portfolio volatility (Risk)
portfolio_volatility = np.sqrt(portfolio_variance)

print("Portfolio Variance is",portfolio_variance)
print("Portfolio Volatility(Risk) is",portfolio_volatility)
```

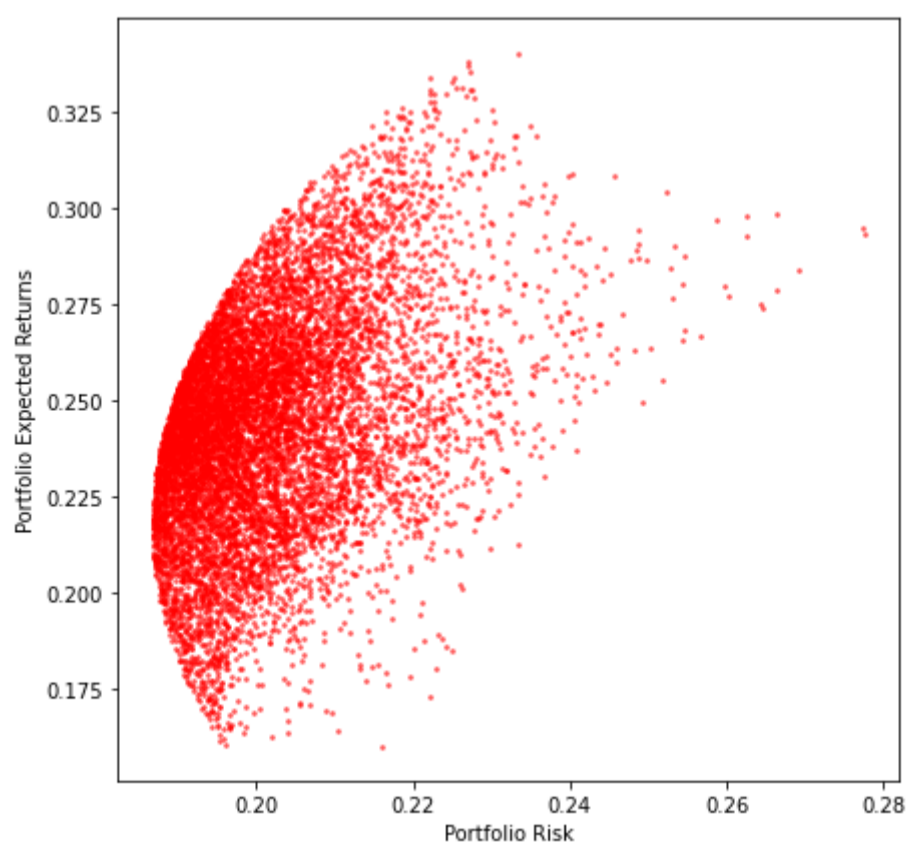
```
Portfolio Variance is 0.03544349973781
Portfolio Volatility(Risk) is 0.18826444098079168
```

现在，我们的任务是优化权重。为什么？这样我们就可以最大化我们的回报或最小化我们的风险——我们通过使用现代投资组合理论来做到这一点！

现代投资组合理论

现代投资组合理论认为，不应单独看待一项投资的风险和收益特征，而应根据投资如何影响整个投资组合的风险和收益来评估。MPT 表明，投资者可以构建多种资产的投资组合，在给定风险水平下实现回报最大化。同样，给定预期回报水平，投资者可以构建风险最低的投资组合。基于方差和相关性等统计指标，个人投资者的表现不如其对整个投资组合的影响重要 [Investopedia]

下图总结了这一理论。我们找到如下所示的边界，并最大化风险水平的预期回报或最小化给定预期回报水平的风险。



我们的目标是为我们的投资组合中的每项资产选择权重，以便我们在给定风险水平的情况下最大化预期回报。

我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya，即表示您同意我们的 [隐私政策](#) 和 [使用条款](#)。 [接受](#)

在 Python 中使用 MPT 进行投资组合优化

$$\begin{aligned} &\text{maximize } E[(\text{weights})^T \cdot \text{returns}] \\ &\text{subject to } (\text{weights})^T \cdot \text{cov}_{\text{portfolio}} \cdot \text{weights} = \sigma^2 \end{aligned}$$

查看有效边界和目标函数的另一种方法是，我们可以最小化风险，因为预期回报至少大于给定值。在数学上，这个目标函数可以写成：

$$\begin{aligned} &\text{minimize } (\text{weights})^T \cdot \text{cov}_{\text{portfolio}} \cdot \text{weights} \\ &\text{subject to } (\text{weights})^T \cdot \mu \geq \mu_{\text{target}} \\ &(\text{weights})^T \cdot \mathbf{1} = 1 \\ &\text{weights}_i \geq 0 \end{aligned}$$

第一行表示目标是最小化投资组合方差，即投资组合的波动性，从而最小化风险。

受制于约束意味着回报必须大于特定的目标回报，所有权重之和应为 1，并且权重不应为负。

现在，我们知道了这个概念，让我们现在开始动手吧！

首先，我们通过运行一个循环来创建有效边界。在每个循环中，我们为投资组合中的资产考虑一组随机分配的不同权重，并计算该权重组合的回报和波动率。

为此，我们创建了 3 个空列表，一个用于存储收益，另一个用于存储波动率，最后一个用于存储投资组合权重。

```
# Initialize an empty list for storing the portfolio returns
port_returns = []
# Initialize an empty list for storing the portfolio volatility
port_volatility = []
# Initialize an empty list for storing the portfolio weights
port_weights = []

num_assets = len(portfolio.columns)
num_portfolios = 10000

# Computing the individual asset return
individual_rets = portfolio.resample('Y').last().pct_change().mean()
```

一旦我们创建了列表，我们会重复随机生成资产的权重，然后将权重归一化为 1。然后我们以与之前计算相同的方式计算 L

我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya，即表示您同意我们的 [隐私政策](#) 和 [使用条款](#)。 [接受](#)

在 Python 中使用 MPT 进行投资组合优化

```
# Randomly generate weight combination
weights = np.random.random(num_assets)
# Normalize weight so that they sum to 1
weights = weights/np.sum(weights)
port_weights.append(weights)
# Returns are the dot product of individual expected returns of asset and its weights
returns = np.dot(weights, individual_rets)
port_returns.append(returns)

# Computing Portfolio Variance
var = var_matrix.mul(weights, axis=0).mul(weights, axis=1).sum().sum()
# Daily standard deviation : volatility is square root of variance
sd = np.sqrt(var)
# Annualizing the standard deviation will give us the volatility
ann_sd = sd*np.sqrt(250)
port_volatility.append(ann_sd)
```

我们现在将数据聚合到一个字典中，然后创建一个数据框来查看资产的权重组合以及它们产生的相应回报和波动率。

```
# Creating a dictionary of returns and volatility
data = {'Returns':port_returns, 'Volatility':port_volatility}

for counter, symbol in enumerate(portfolio.columns.tolist()):
    data[symbol+' weight'] = [w[counter] for w in port_weights]

# Converting the dictionary to dataframe
portfolios_V1 = pd.DataFrame(data)
portfolios_V1.head()
```

这里是投资组合收益、波动率和权重数据的预览。

	Returns	Volatility	AAPL weight	NKE weight	GOOGL weight	AMZN weight
0	0.208832	0.197568	0.100482	0.257367	0.421928	0.220222
1	0.202642	0.195550	0.050817	0.387201	0.319514	0.242468
2	0.241573	0.202058	0.202403	0.179829	0.310002	0.307765
3	0.261990	0.203404	0.229123	0.358215	0.007626	0.405036
4	0.229728	0.196022	0.144460	0.341310	0.213912	0.300318

现在我们拥有一切，可以进入这场比赛的最后一圈并找到最佳配重组！

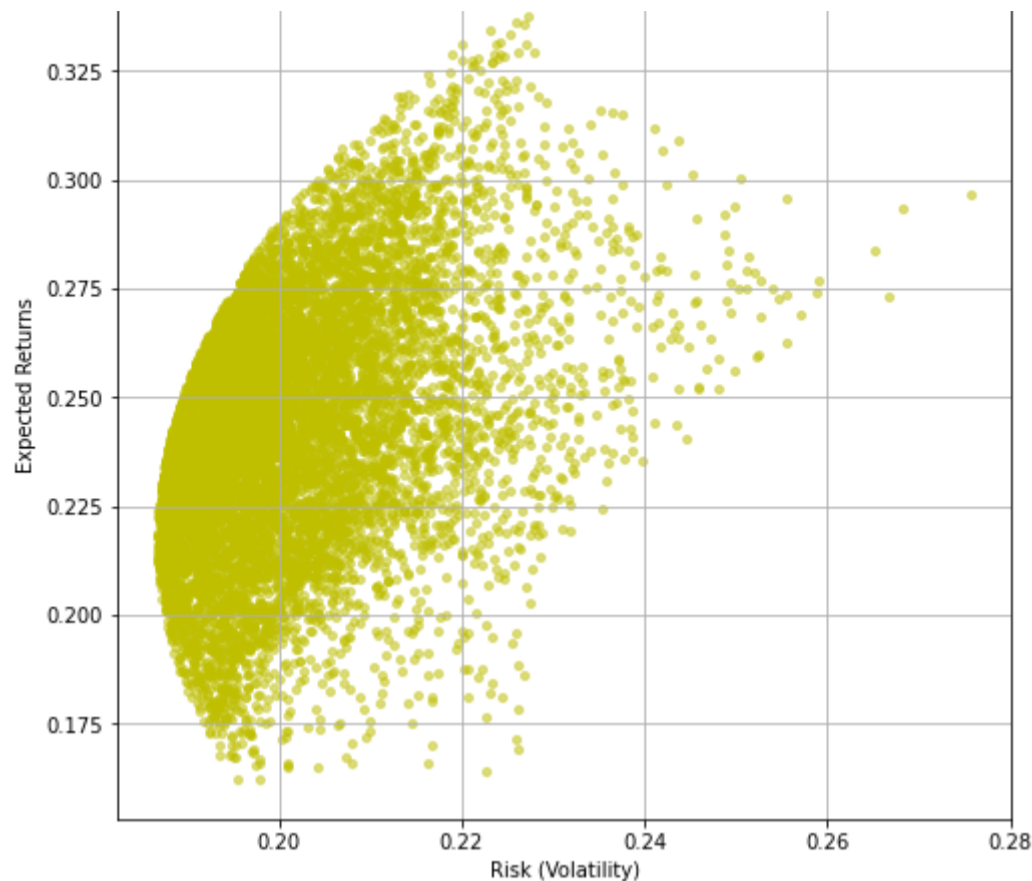
我们根据上面计算的回报绘制波动率，这将为我们提供本文开头想要创建的有效边界。

```
# Plot efficient frontier
portfolios_V1.plot.scatter(x='Volatility', y='Returns', marker='o', color='y',
                           s=15, alpha=0.5, grid=True, figsize=[8,8])
plt.xlabel("Risk (Volatility)")
plt.ylabel("Expected Returns")
```

现在我们开始！！



在 Python 中使用 MPT 进行投资组合优化



现在我们有有效边界，让我们找到最佳权重。

我们可以优化使用多种方法，如下所示：

- 具有最小波动性（风险）的投资组合
- 最佳投资组合（最大夏普比率）
- 风险水平的最大回报；预期回报水平下的最低风险；索提诺比率最高的投资组合

在本文中，我将通过前两种方法进行优化。在第三部分中，“风险水平的最大回报”和“预期回报水平的最小风险”非常简单，而基于索提诺比率的部分类似于夏普比率

1. 最低波动率

为了找到最小波动率组合，我们在数据框中选择对应于最小方差的行，然后使用 `.idxmin()` 函数找到该行。执行相同操作的代码如下所示：

```
# MIN VOLATILITY
# idxmin() gives us the minimum value in the column that has been specified.
min_vol_port = portfolios_V1.iloc[portfolios_V1['Volatility'].idxmin()]
min_vol_port
```

我们从中得到的权重是：

```
Returns      0.218711
Volatility    0.186736
AAPL weight   0.265604
NKE weight    0.389033
GOOGL weight  0.303345
AMZN weight   0.042018
```

现在，找出上面曲线上的最小波动点在哪里？答案见文末！不过不要作弊！

2. 最高夏普比率

夏普比率是超过每单位波动率或总风险的无风险利率的平均回报。用于计算夏普比率的公式如下：

$$\text{夏普比率} = (R_p - R_f) / SD_p$$

在哪里，

- R_p 是投资组合的回报

我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya，即表示您同意我们的 [隐私政策](#) 和 [使用条款](#)。 [接受](#)

在 Python 中使用 MPT 进行投资组合优化

```
# Finding the optimal portfolio: Highest SHARPE RATIO
# risk factor
rf = 0.01
optimal_risky_port = portfolios_V1.iloc[((portfolios_V1['Returns']-rf)/portfolios_V1['Volatility']).
                                         idxmax()]
optimal_risky_port
```

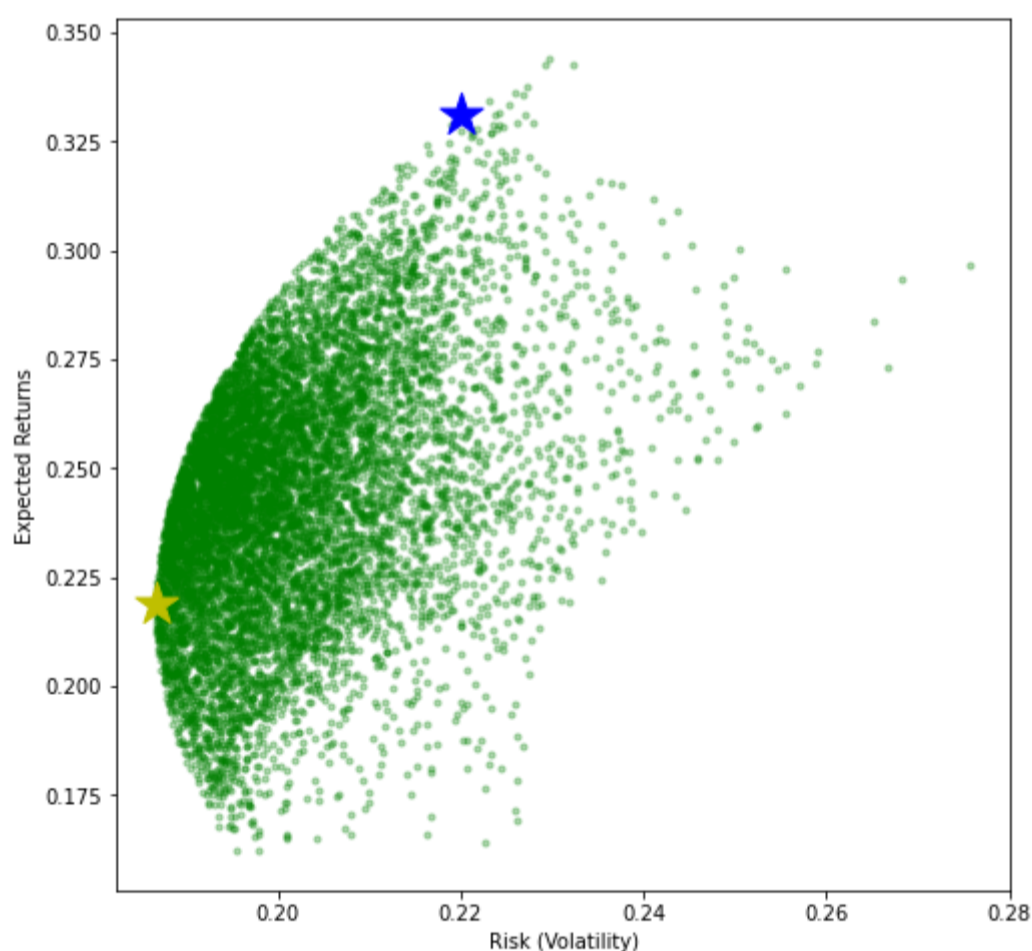
给出最高夏普比率的投资组合的权重如下所示：

```
Returns      0.330981
Volatility    0.220135
AAPL weight   0.715670
NKE weight    0.086810
GOOGL weight  0.000870
AMZN weight   0.196650
```

找出最高夏普比率点位于上述曲线的哪个位置？答案如下图！再说一次，不要作弊！

```
# Plotting optimal portfolio and minimum volatility
plt.subplots(figsize=(8, 8))
plt.scatter(portfolios_V1['Volatility'], portfolios_V1['Returns'], marker='o', s=10,
            alpha=0.3, color='green')
plt.scatter(min_vol_port[1], min_vol_port[0], color='y', marker='*', s=500)
plt.scatter(optimal_risky_port[1], optimal_risky_port[0], color='b', marker='*', s=500)
plt.xlabel("Risk (Volatility)")
plt.ylabel("Expected Returns")
```

正如这里所承诺的，就是以上两个问题的答案！



蓝星对应最高夏普比率点，黄色星对应最小波动率点。

现在，在结束本文之前，我鼓励您找到“风险水平的最大回报”和“预期回报水平的最小风险”投资组合，并将它们标记在上图中。

除了使用夏普比率外，我们还可以使用索提诺比率来优化投资组合。

简要介绍一下：

Sortino比率是夏普比率的一种变体，它通过使用资产的负投资组合回报标准差（下行偏差）而不是投资组合回报的总标准差

在 Python 中使用 MPT 进行投资组合优化

在哪里,

- R_p 是投资组合的回报
- r_f 是无风险利率
- SD_d 是下行的标准差

如果您在实施或理解我上面描述的 3 种优化方法时遇到任何问题, 请写信给我: parhtyagi.iitd@gmail.com | parhtba2022@email.iimcal.ac.in | 或在 LinkedIn 上联系 <https://www.linkedin.com/in/parth-tyagi-4b867452/>

关于作者的注释: Parth Tyagi 目前正在加尔各答 IIM、IIT Kharagpur 和印度统计研究所加尔各答攻读 PGDBA, 并已完成 IIT Delhi 的 B.Tech。他在高级分析领域拥有约 4 年的工作经验。

本文中关于 *Python* 投资组合优化的媒体不归 *Analytics Vidhya* 所有, 由作者自行决定使用。

[博客马拉松](#) [投资组合优化](#)

关于作者



[帕斯泰亚吉](#)

我们的顶级作者

[查看更多](#)

下载

最新博客/文章的 Analytics Vidhya App



上一篇

[如何使用 CatBoost 进行心理疲劳评分预测](#)

下一篇

[Shapash- 使机器学习可解释的 Python 库](#)

发表评论

您的电子邮件地址不会被公开。必需的地方已做标记 *



我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya, 即表示您同意我们的 [隐私政策](#) 和 [使用条款](#)。 [接受](#)

在 Python 中使用 MPT 进行投资组合优化

通过电子邮件通知我后续评论。 通过电子邮件通知我新帖子。

提交

热门资源



[From Zero to Millionaire: Generate Passive Income using ChatGPT](#)

[Aravindpai Pai](#) - APR 17, 2023



[Microsoft Releases VisualGPT: Combines Language and Visuals](#)

[K.sabreena](#) - APR 15, 2023



[FreedomGPT: Personal, Bold and Uncensored Chatbot Running Locally on Your..](#)

[K.sabreena](#) - APR 08, 2023



[Understand Random Forest Algorithms With Examples \(Updated 2023\)](#)

[Sruthi E R](#) - JUN 17, 2021

[Analytics Vidhya](#)

[About Us](#)

[Our Team](#)

[Careers](#)

[Data Scientists](#)

[Blog](#)

[Hackathon](#)

[Discussions](#)

我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya，即表示您同意我们的[隐私政策](#)和[使用条款](#)。 [接受](#)

在 Python 中使用 MPT 进行投资组合优化

Trainings

Hiring Hackathons

Advertising

© Copyright 2013-2023 Analytics Vidhya.

[Privacy Policy](#) [Terms of Use](#) [Refund Policy](#)



我们在 Analytics Vidhya 网站上使用 cookie 来提供我们的服务、分析网络流量并改善您在网站上的体验。使用 Analytics Vidhya，即表示您同意我们的[隐私政策](#)和[使用条款](#)。 [接受](#)