DJANGOによる家計管理アプリ作成

1

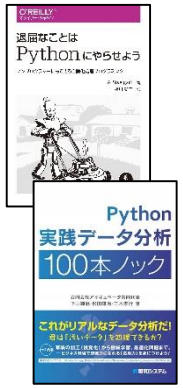
2021/10/01 中向

1. 背景

アプリ作成の勉強を始めました。これまでのAIの知識とアプリの知識を組み合わせて家計管理アプリを作ってみたので共有します。

勉強内容

2019/10 業務効率化のためPython勉強開始



2020/2 機械学習の勉強開始

2020/8 Deep Learningの勉強開始

2020/9 実践のため、Kaggleに挑戦開始（銅メダル取得）

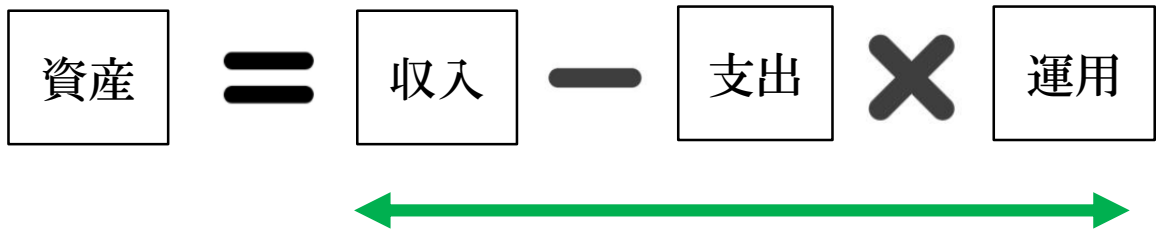
kaggle

2021/8 実践のため、アプリ作成の勉強開始

2. 企画

資産の管理において重要な、収入・支出・運用を管理できるアプリを作ってみました。

コンセプト



収入から運用までをカバーした
家計管理アプリ

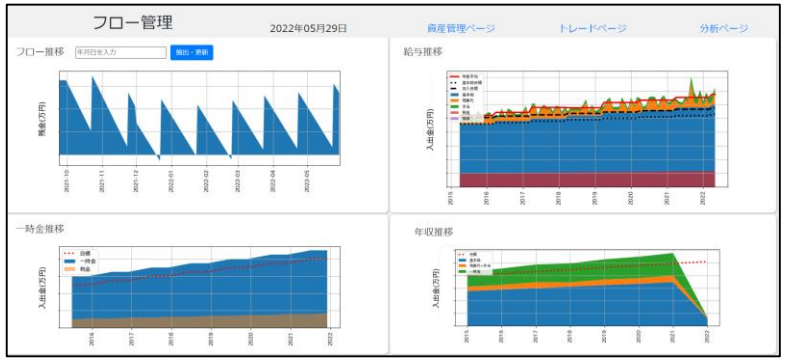
作った機能

- 収入と支出の状況可視化
- 資産状況の可視化
- 将来の市場分析
- 自分の資産状況と市場分析結果を加味した、最適な投資額の計算

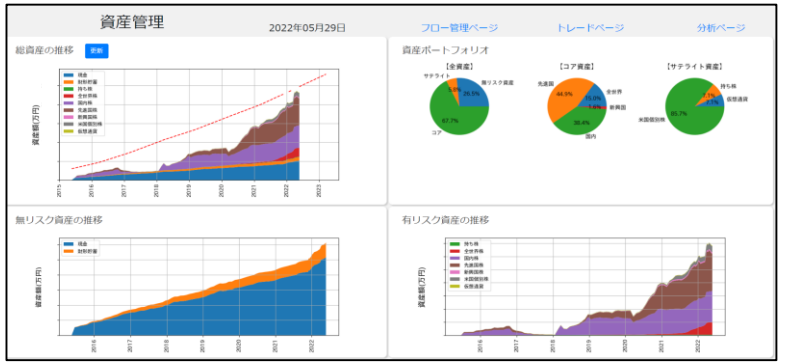
3. 画面遷移

画面は4つあり、お互いに遷移できるようになっています。

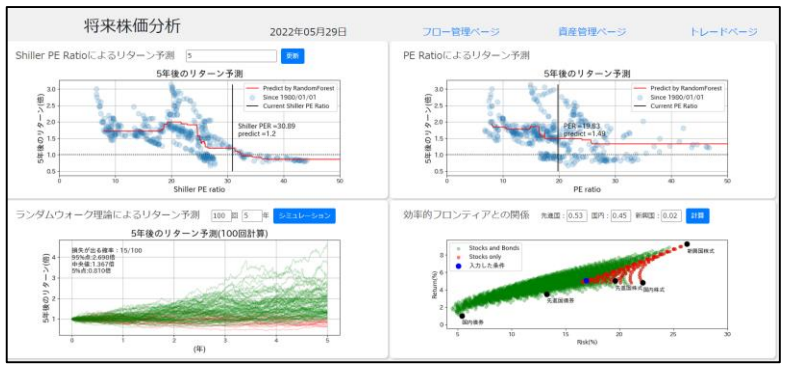
①収入・支出の可視化



②資産状況の可視化



③将来市場の分析



④最適投資額の計算

4. 収入・支出の可視化

収入・支出の可視化画面は、以下のような画面になっています。

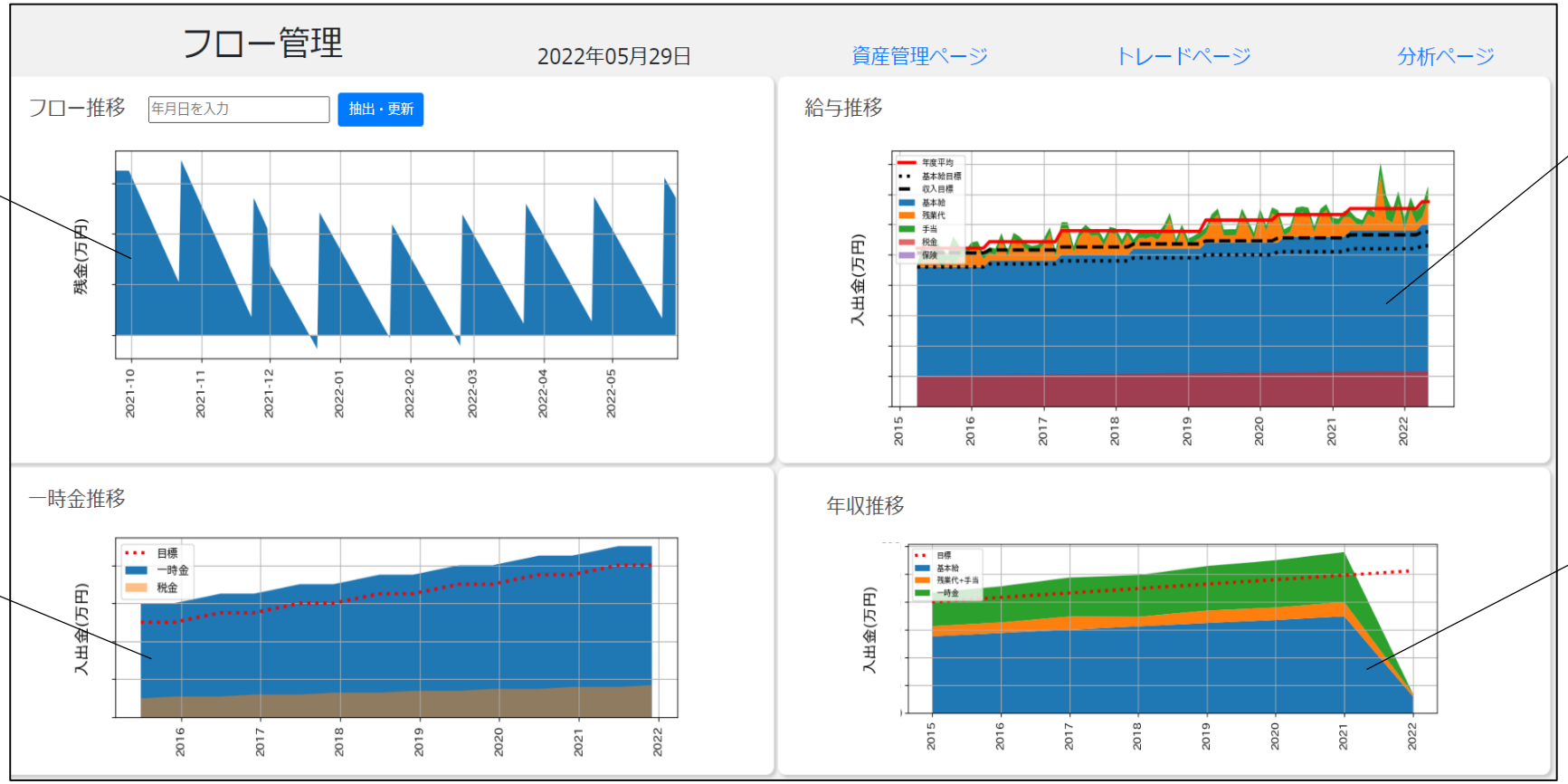
①収入・支出の可視化

毎月の収入・支出の関係がわかる

一時金の推移がわかる

毎月の収入の推移がわかる

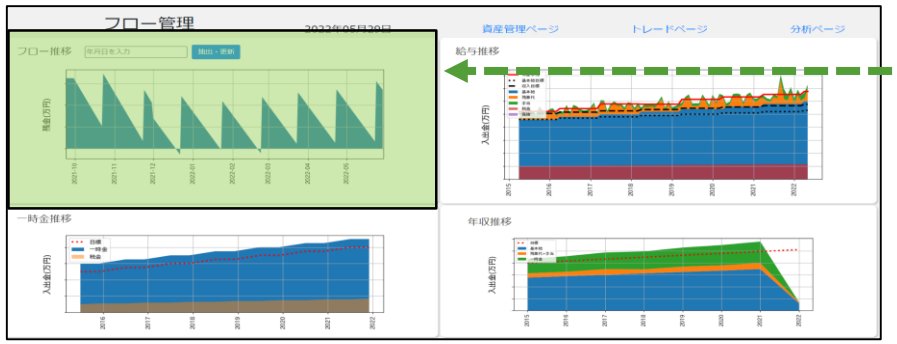
毎年の収入の推移がわかる



4. 収入・支出の可視化

基本的に、Pythonでデータ読み込み・グラフ作成 ⇒ HTMLで画像表示という流れです。

①収入・支出の可視化



flow.html

```
<!-- 左上  -->
<div class="row">
    <div class="card">
        <div class="in-card">
            <form class="form-inline" action="" method="post" autocomplete="off">{% csrf_token %}
                <h2>フロー推移  </h2>
                    <input type="text" id="select_month" name="extract_month" placeholder="年月日を入力" value={{ extract_month }}>
                      
                <input type="submit" class="btn btn-primary" value="抽出・更新" >
            </form>
            <div class="main">
                <img src="{% static 'images/flow_visual.png' %}?20211022" width=100%/>
            </div>
        </div>
    </div>
```

②flow_visual.png画像を埋め込む

flow_visualization.py

```
# 左上グラフ化
plt.rcParams['figure.subplot.bottom'] = 0.3
plt.rcParams['figure.subplot.top'] = 0.95
plt.subplots_adjust(left=0.1, right=0.95, bottom=0.3, top=0.95)
plt.figure(figsize=(10, 4))

plt.fill_between(df_timeseries.index, df_timeseries['残金']/10000*correction, [0])
plt.tick_params(labelsize=11)
plt.xticks(rotation=90)
plt.yticks(fontsize=12)
plt.savefig(str(BASE_DIR)+'/static/images/flow_visual.png')
```

①pythonのmatplotlibを使ってflow_visual.pngというグラフ画像を作成

5. 資産状況の可視化

資産状況の可視化画面は、以下のような画面になっています。

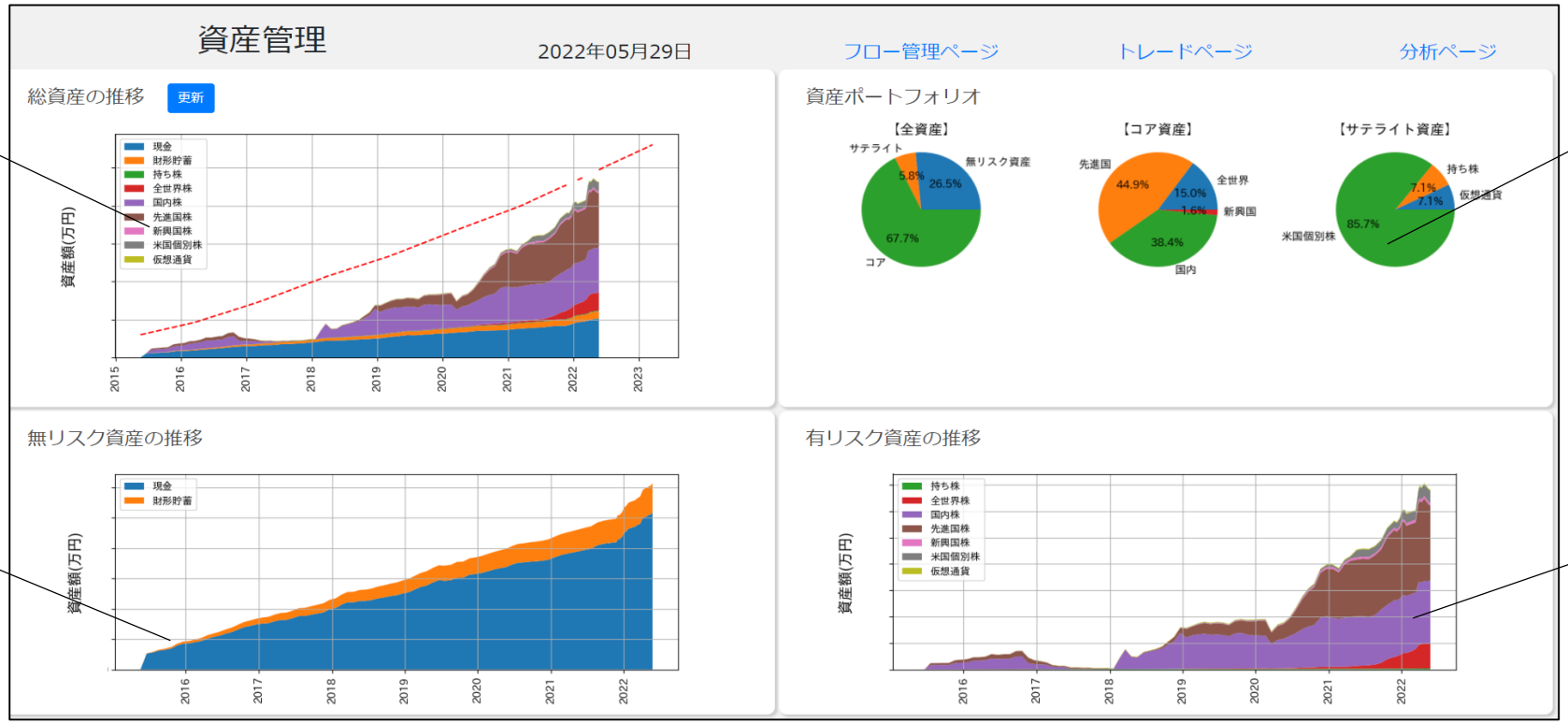
②資産状況の可視化

資産の推移がわかる

無リスク資産の
推移がわかる

今の資産の
配分がわかる

有リスク資産の
推移がわかる



6. 将来市場の分析

将来市場の分析画面は、以下のような画面になっています。

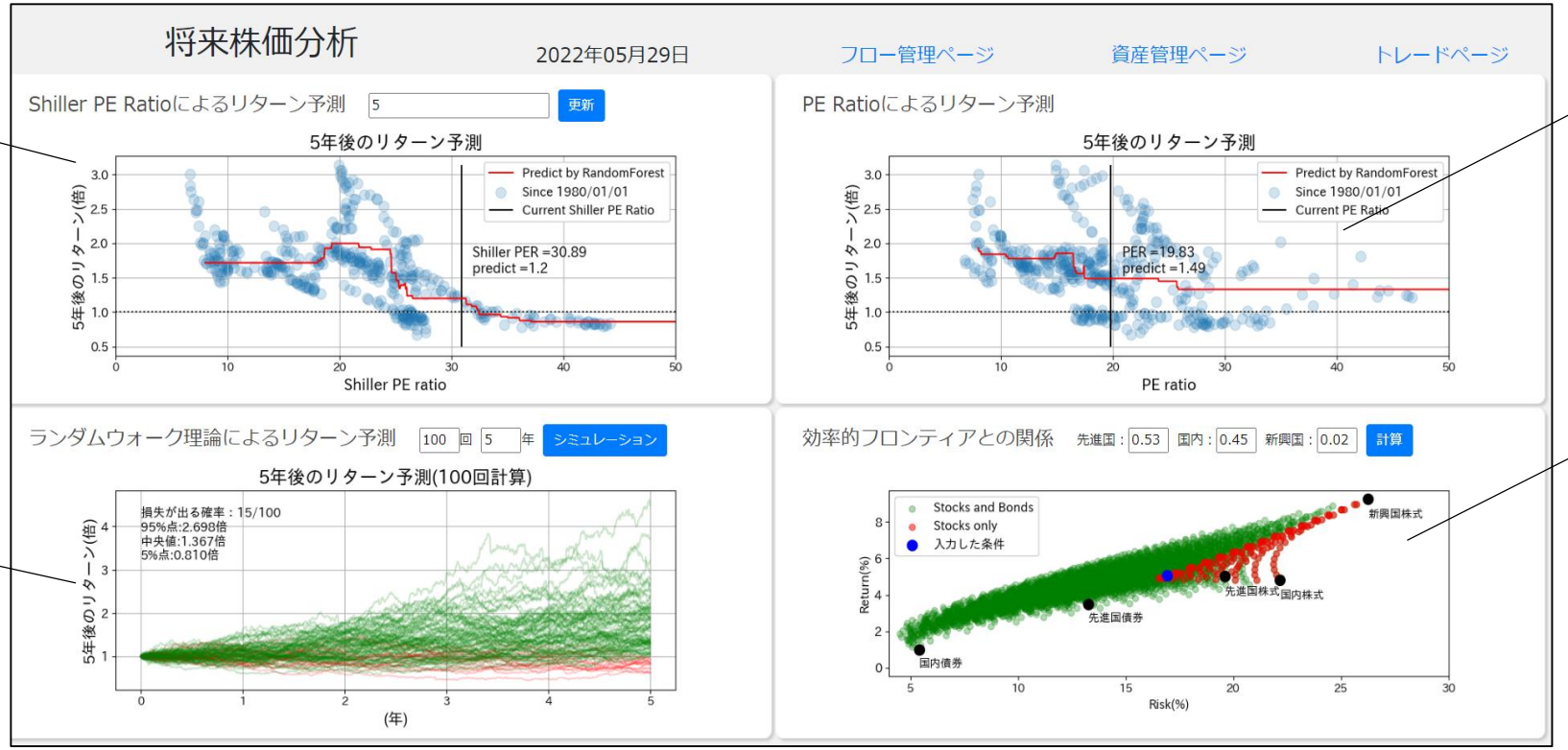
③将来市場の分析

数年後のリターンを予測

- Webスクレイピングで Shiller PERを取得
- AI(RandomForest)で 将来株価を予測
- デフォルトは5年後の予測 だが変更可能

数年後のリターンを予測

- 過去の株価データと AI(OneClassSVM + 最尤推定) を使って分布を計算
- シミュレーションを複数回 行って、リターンを予測
- デフォルトは5年後の予測を 100回シミュレーションだが 変更可能



数年後のリターンを予測

- Webスクレイピングで PERを取得
- AI(RandomForest)で 将来株価を予測
- デフォルトは5年後の予測 だが変更可能

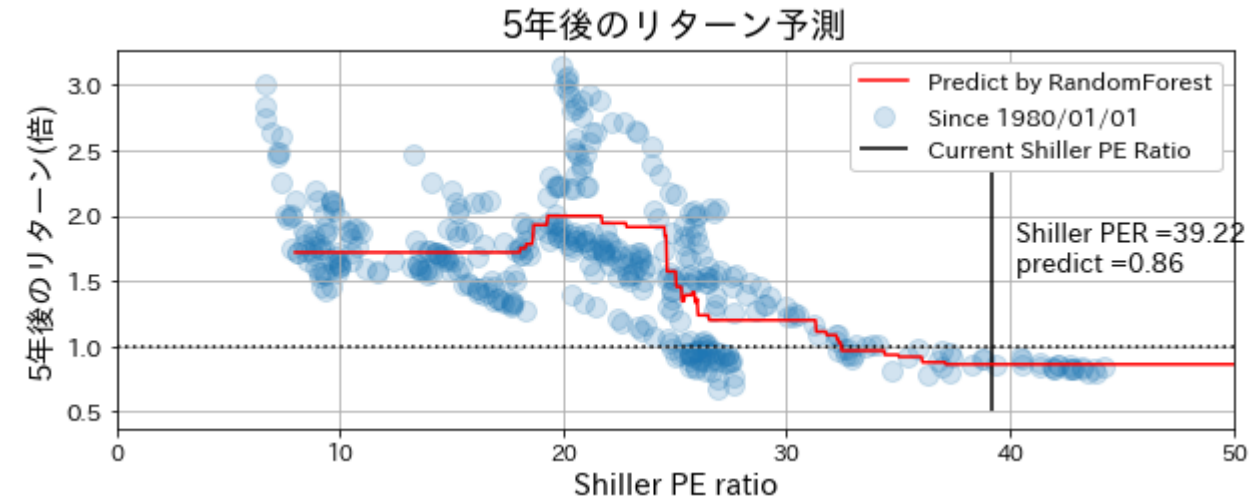
現在の資産のリスク・ リターンを計算

- 各資産のリスク、相関係数から、現在持っている資産のリスク・リターンを計算
- 青の点が現状の位置、左上に行くほど良い

6. 将来市場の分析

Shiller PER と 将来リターンは図のように相関があると言われています。
そこに、ランダムフォレストを使って将来リターンを予測しています。

Shiller PERによる将来リターン予測

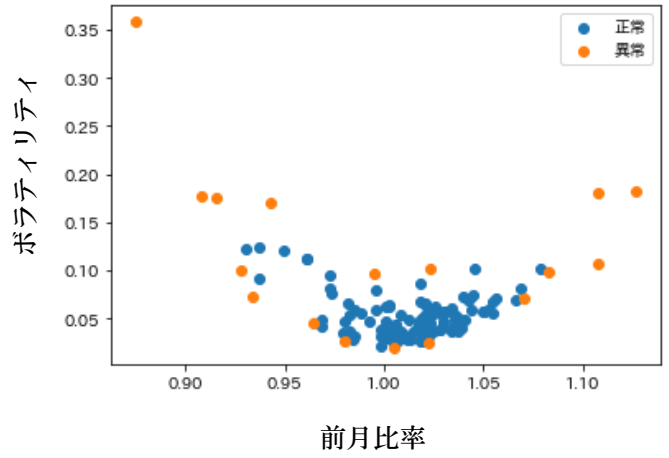


①Webスクレイピングで、
　現在のShiller PERの値を取得(黒線)
②過去のデータ(青点)を
　RandomForest(赤線)で学習
③5年後のリターンが0.86倍になると予測

6. 将来市場の分析

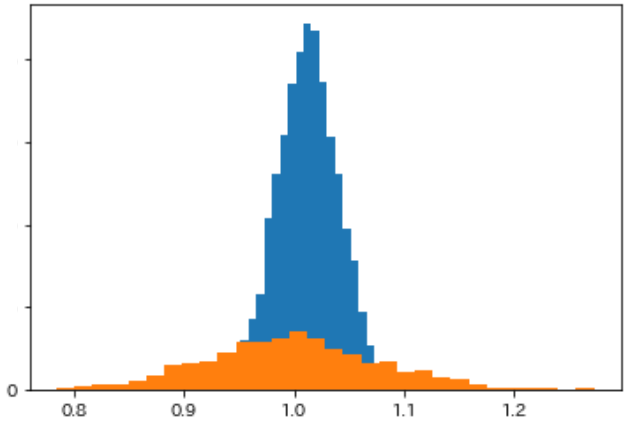
株価の推移は、正常時と異常時でそれぞれ正規分布していると言われています。正常と異常をOneClassSVMで分けた後、最尤推定をすることで正規分布を推定でき、シミュレーションが可能になります。

OneClassSVMによる株の正常・異常分類



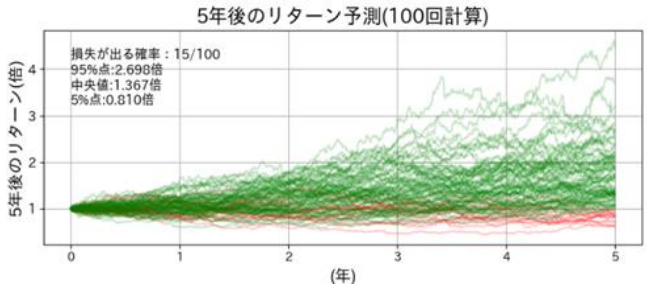
ボラティリティで正常・異常を分けるというのをよくやります。

正常時、異常時の正規分布推定



正常時は平均がプラスなのに対して、
異常時は平均がマイナスで標準偏差が大きい
（悲観的になると下がりやすい傾向）

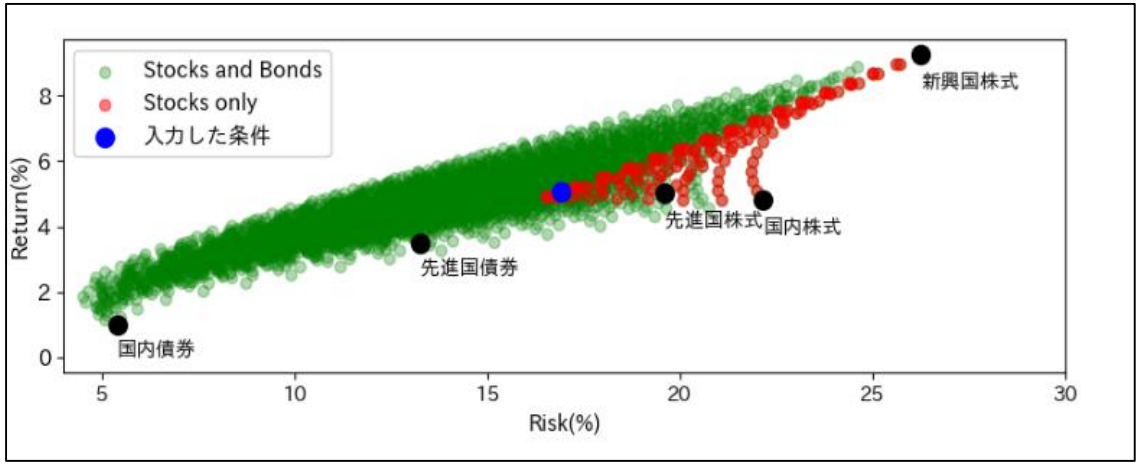
正規分布を使ったシミュレーション



6. 将来市場の分析

保有している資産同士の相関係数と標準偏差から、リスク・リターンを計算することができます。この時、効率の良い条件である左上の領域を効率的フロンティアと言います。

効率的フロンティアと現在の資産の関係

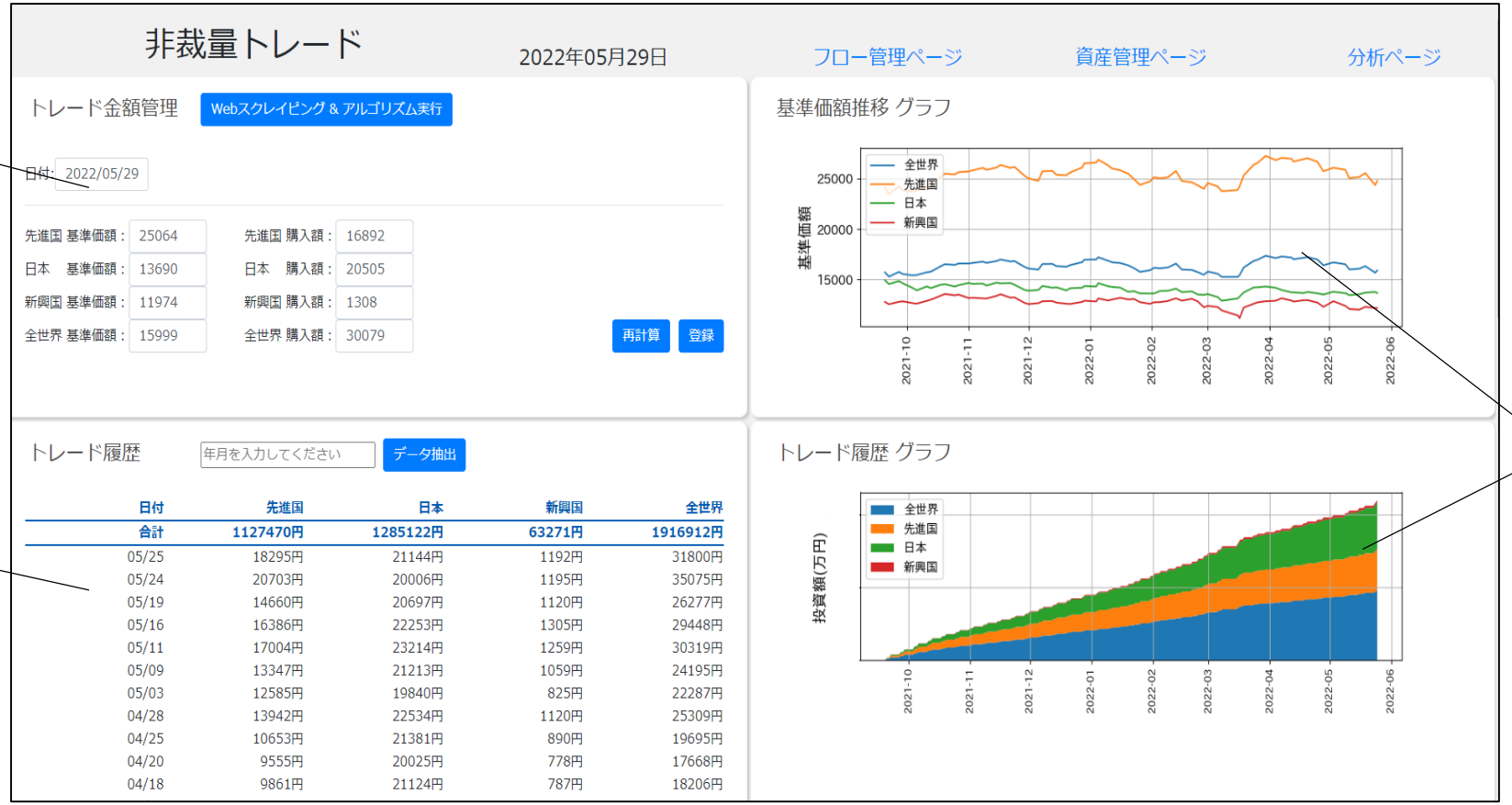


現在の資産から計算されるリスク・リターンが青点。
左上に行くほど良い。

7. 最適投資額の計算

最適投資額の計算画面は、以下のような画面になっています。

④最適投資額の計算



日付	先進国	日本	新興国	全世界
合計	1127470円	1285122円	63271円	1916912円
05/25	18295円	21144円	1192円	31800円
05/24	20703円	20006円	1195円	35075円
05/19	14660円	20697円	1120円	26277円
05/16	16386円	22253円	1305円	29448円
05/11	17004円	23214円	1259円	30319円
05/09	13347円	21213円	1059円	24195円
05/03	12585円	19840円	825円	22287円
04/28	13942円	22534円	1120円	25309円
04/25	10653円	21381円	890円	19695円
04/20	9555円	20025円	778円	17668円
04/18	9861円	21124円	787円	18206円

最適投資額を計算

- Webスクレイピングで現在の株価を取得
- 先ほどの将来市場の分析結果と現在の株価から購入額を計算

投資額を記録

- 左上画面で登録ボタンを押すと投資額が記録されていく

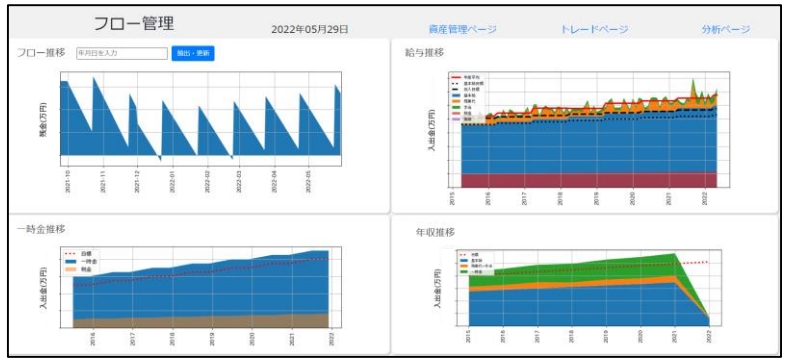
投資額を可視化

- 左下画面のデータをグラフ化

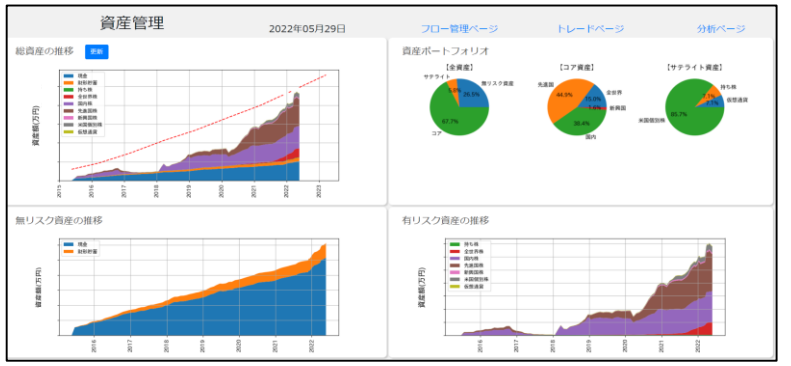
8. 纏め

ということで、Pythonでひたすらグラフを作って、HTMLに埋め込むことでアプリっぽいものができました。

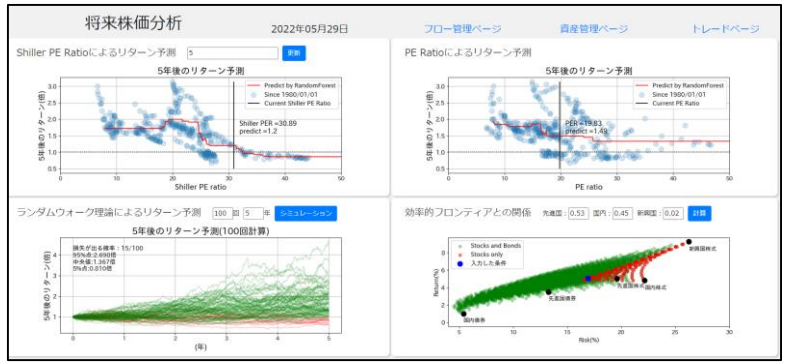
①収入・支出の可視化



②資産状況の可視化



③将来市場の分析



④最適投資額の計算