Portfolio Optimization
Credit Risk Managment

Ali Hachem, Erisa Murati, Megi Kasemi
Xinyu Wang, Zinan Zhou

21st January 2020

Group Organisation

Portfolio Optimization

Ali Hachem, Erisa Murati, Megi Kasemi
Studying the portfolio optimization process and proposing methods for having an optimal portfolio.

Credit Risk Managment

Xinyu Wang, Zinan Zhou
Getting the Credit Score Card and optimizing the model by using different methods on new dataset.

Contents

Contents

Introduction

Motivation

Portfolio optimization is the process of selecting the best portfolio (asset distribution), out of the set of all portfolios being considered, subject to maximizing expected return, and minimizing financial risk.

Several mechanisms to manage portfolios:

- Sharpe Ratio
- Omega Ratio

(Maximizing these two functions)

Objectives

Objective

Propose methods to maximize these two functions

- Monte Carlo Simulation
- Analytical method

- Using **3** different datasets *(Istanbul, United States* and *ETF Stocks)*
- Studied for **5** time periods *(daily, weekly, monthly, quarterly, semester)*

Sharpe Ratio

Sharpe Ratio measures the performance of an investment compared to a risk-free asset, after adjusting for its risk.

$$\text{Sharpe Ratio} = f(x_1, x_2, \ldots, x_n)$$

x_i - percentage of money of asset i

- Non-linear
- Convex

It is used to help investors understand the return of an investment compared to its risk.

Monte Carlo Results

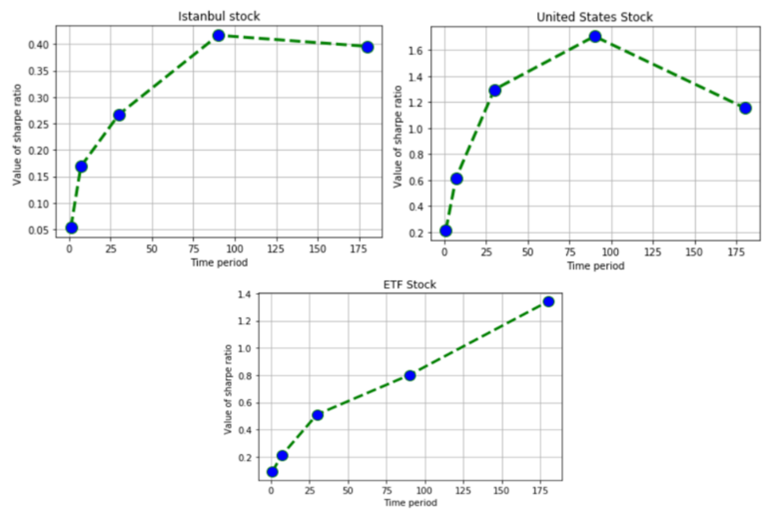
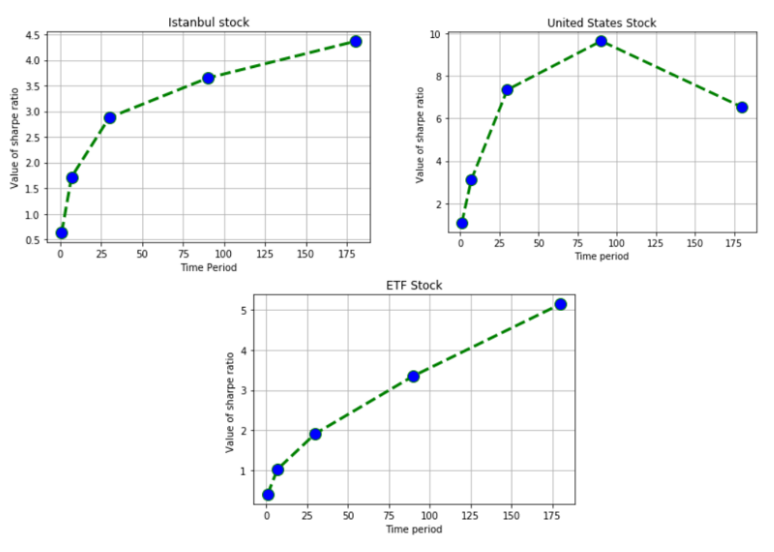


Figure: Monte Carlo results: max Sharpe Ratio value

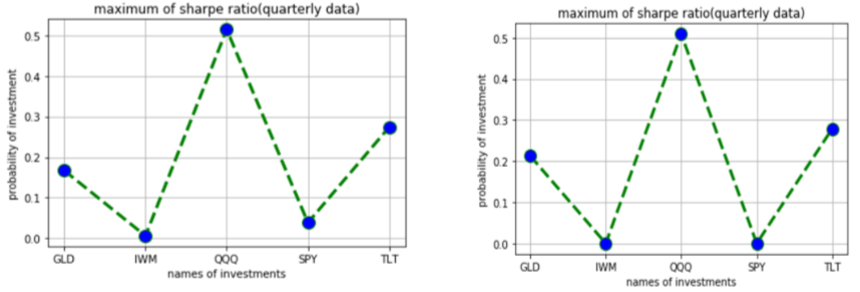
Analytic method

- Going from non-linear to quadratic programming problem
- Using libraries supporting quadratic programming in Python

Analytic method results



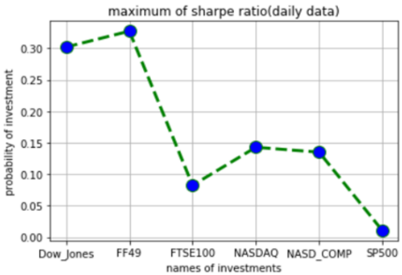
Results comparison

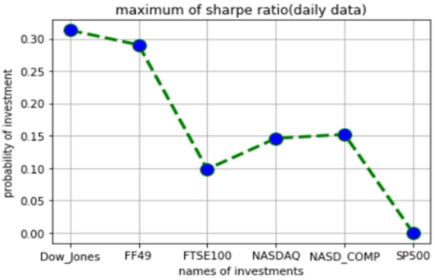


	GLD	IWM	QQQ	SPY	TLT
MC	0.16	0.006	0.51	0.03	0.27
AM	0.212	0.0	0.509	0.0	0.27

Figure: Values of the weights for the assets of ETF stock: MC (left) AM (right)

Results comparison





	DOW_JONES	FF49	FTSE100	NASDAQ	NASDQ_COMP	SP500
MC	0.301	0.327	0.082	0.142	0.135	0.01
AM	0.312	0.290	0.0098	0.146	0.152	0.0

Figure: Values of the weights for the assets of United States stock

Results comparison

$$\text{Omega Ratio} = f(x_1, x_2, \ldots, x_n, \mathbf{L})$$

x_i - percentage of money of asset i

- Non-linear
- Non-convex

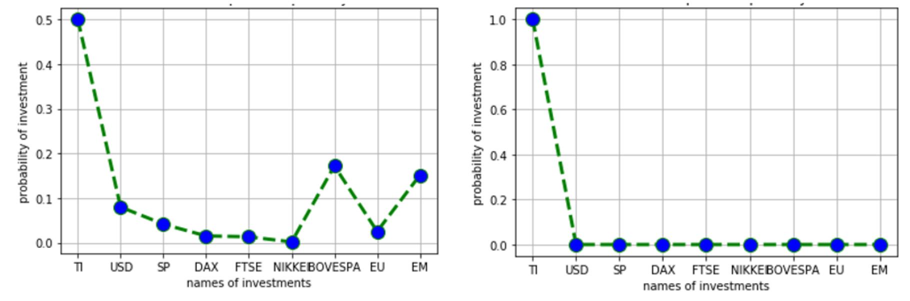
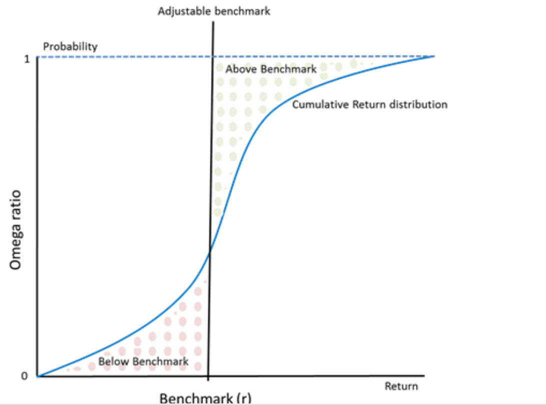


Figure: Values of the weights for the assets of Istanbul stock

Omega Ratio

$$\text{Omega Ratio} = f(x_1, x_2, \ldots, x_n, \mathbf{L})$$

- Non-linear
- Non-convex



Monte Carlo Results

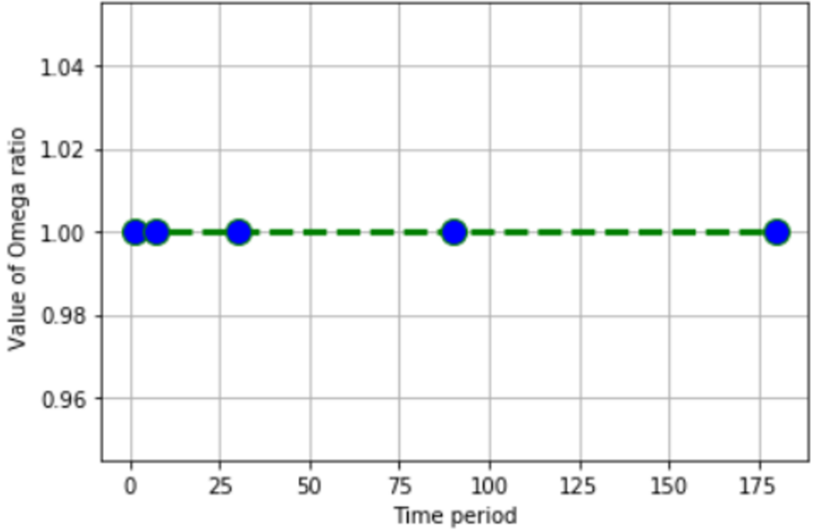


Figure: Monte Carlo results: max Omega Ratio for all datasets

Analytic method

L is usually chosen by the investor by considering how much risk he want to take.

Approximate the value of L by solving a linear programmingApproximation from non-linear to linear programming of Omega Ratio problem

Analytic method results

Figure: Analytic method: max Omega Ratio value

Conclusions

Contents

Objective

Finding the module which can help bank do the right choice for issuing credit card and reduce the risk.

Data explication and feature selection

	Label	Explanation
X0	SeriousDlqin2yrs	Good clients and defaulting clients
X1	RevolvingUtilisationOfUnsecuredLines	Recycling of unsecured loans(Credit card and personal credit limit, except for real estate and non-amortization debts, which are divided by the sum of credit lines, such as car loans)
X2	Age	Age of client
X3	NumberOfTime30-59DaysPastDueNotWorse	Times of 35-59 days overdue but not bad
X4	DebtRatio	Debt ratio
X5	MonthlyIncome	Monthly Income real
X6	NumberOfOpenCreditLinesAndLoans	Number of open credits and loans, open loans (instalments such as car loans or mortgages) and credit (such as credit cards)
X7	NumberOfTimes90DaysLate	The number of times the client has overdue for 90 days or more
X8	NumberRealEstateLoansOrLines	Real estate loans or quotas: Mortgage and real estate lending including home equity credit lines
X9	NumberOfTime60-89DaysPastDueNotWorse	Times of 60-89 days overdue but not bad
X10	NumberOfDependents	Number of dependents not including myself

Figure: Data explication

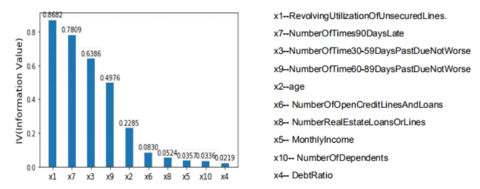


Figure: Feature selection

logistic regression

Logit Regression Results

Dep. Variable:	SeriousDlqin2yrs	**No. Observations:**	99785
Model:	Logit	**Df Residuals:**	99779
Method:	MLE	**Df Model:**	5
Date:	Sun, 15 Dec 2019	**Pseudo R-squ.:**	0.2214
Time:	19:50:11	**Log-Likelihood:**	-17593.
converged:	True	**LL-Null:**	-22595.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	2.7302	0.016	171.528	0.000	2.699	2.761
RevolvingUtilizationOfUnsecuredLines_woe	0.6379	0.017	37.261	0.000	0.604	0.671
age_woe	0.5660	0.035	15.974	0.000	0.497	0.635
NumberOfTime30-59DaysPastDueNotWorse_woe	0.5504	0.017	31.964	0.000	0.517	0.584
NumberOfTimes90DaysLate_woe	0.5878	0.014	40.689	0.000	0.559	0.616
NumberOfTime60-89DaysPastDueNotWorse_woe	0.4437	0.019	23.599	0.000	0.407	0.481

Figure: Results of Regression Logistic

Result

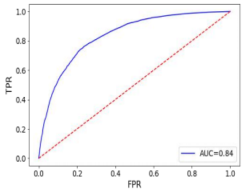
Several scorecard parameters: the base score, PDO (the ratio doubles the score), and the quality ratio.

	A	B	C	D	E	F	G	H	I
1		SeriousDlqin	BaseScor	zationOf	age	-59DaysP	fTimes90D	-89DaysP	Score
2	112283	1	435	−19	−9	−27	−34	−23	323
3	62127	1	435	−19	15	−14	−34	−23	360
4	43871	1	435	−19	−3	−14	−34	−23	342
5	103321	1	435	−19	−9	−14	−34	−23	336
6	65734	1	435	22	11	−14	−34	−23	397
7	26964	1	435	−19	−2	−14	−34	−23	343
8	24938	1	435	21	−9	−14	−34	−23	376

Figure: Credit Score Card

Module evaluation

Verify how predictive the model is. Automatic calculation of ROC and AUC



	Relevant	NonRelevant
Retrieved	true positives (tp) 219	false positives (fp) 1278
Not Retrieved	false negatives (fn) 163	true negatives (tn) 23287

Figure: ROC curve which demonstrate the effect of binary classification and Confusion Matrix

Credit dataset

- German credit dataset - 1000 sample
- Credit data: imbalanced data

Confusion Matrice

```
[[   188   1309]
 [   145  23305]]
```

		precision	recall	f1-score	support
bad clients	0	0.56	0.13	0.21	1497
good clients	1	0.95	0.99	0.97	23450
	accuracy			0.94	24947
	macro avg	0.76	0.56	0.59	24947
	weighted avg	0.92	0.94	0.92	24947

Figure: Results of LR for GiveMeSomeCredit

- Balance data
  - LR : using SMOTE before LR
  - SVM : using parameter ”classweight=’balanced’” in the fuction SVC

Balancing data

- SMOTE : a simple upsampling method to directly copy a few samples of the minority class and add them to the sample set.

		precision	recall	f1-score	support
Good clients	1	0.82	0.86	0.84	148
Bad clients	2	0.55	0.46	0.50	52
accuracy				0.76	200

Figure: Results of LR in German data

	precision	recall	f1-score	support
1	0.87	0.80	0.84	148
2	0.54	0.65	0.59	52
accuracy			0.77	200

Figure: Results of LR after SMOTE in German data

- SVC(classweight='balanced') : Increasing C value of the weights based on the class.

$$C_j = C * w_j$$
$$w_j = \frac{n}{k * n_j}$$

C is the hyperparameter responsible for penalizing misclassified data.

Support vector machines(SVM)

- In python, two methods SVC and LinearSVC.
- **SVC** : five different kernel functions of SVC method.
  - SVM rbf
  - SVM linear
  - SVM poly
  - SVM sigmoid
  - SVM precomputed
- Cross Validation: find the optimal penalty coefficient C.
- **LinearSVC**: more flexibility in the choice of penalties and loss functions and scale better to large numbers of samples.

	precision	recall	f1-score
0	0.17	0.71	0.28
1	0.98	0.79	0.87
accuracy			0.78

Figure: Results of SVC(kernel=linear) in GiveMeSomeCredit

	precision	recall	f1-score
0	0.56	0.17	0.26
1	0.95	0.99	0.97
accuracy			0.94

Figure: Results of LinearSVC in GiveMeSomeCredit

Comparision of LR and LinearSVC

- Compare LR and LinearSVC after data balance

```
accuracy of LR: 0.942237543592416
[[  188  1309]
 [  145 23305]]
              precision    recall  f1-score

           0       0.56      0.13      0.21
           1       0.95      0.99      0.97

    accuracy                           0.94
```

Figure: Results of LR in GiveMeSomeCredit

```
accuracy of LinearSVC:  0.942237543592416
[[  255  1242]
 [  199 23251]]
              precision    recall  f1-score

           0       0.56      0.17      0.26
           1       0.95      0.99      0.97

    accuracy                           0.94
```

Figure: Results of LinearSVC in GiveMeSomeCredit

Results and Conclusions

- Risk of credit card is considered as binary classification problems.
- Before training the model, feature selection is a important part.
- We compare LR with SVM. SVM represent a better result for the imbalanced data.
- For further research, fuzzy logic is also a good model for credit data.

Contents

THANK YOU FOR YOUR ATTENTION

Questions?