Universidade De São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação

MAC0499 Trabalho de Formatura Supervisionado

Classificação e Análise de Desempenho de Fundos

José Corsini Filho
prof.corsini@gmail.com

São Paulo
Dezembro de 2012

Universidade De São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação

José Corsini Filho
prof.corsini@gmail.com

Classificação e Análise de Desempenho de Fundos

MAC0499 Trabalho de Formatura Supervisionado do Departamento de Ciência da Computação da Universidade De São Paulo para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Prof. Dr. Renato Vicente*

São Paulo
Dezembro de 2012

À minha lutadora mãe, Ione, e ao meu pai e herói, José Corsini (em memória), pelo amor incondicional e por terem me proporcionado uma vida completa em todos os sentidos. A todos amigos que torceram e acreditaram em mim, até que este sonho pudesse ser realizado. Em especial ao amigo Israel Lacerra, que me incentivou e apoiou na retomada e finalização desta trajetória.

Resumo

Este trabalho tem como objetivo a construção de um sistema para classificação e análise de desempenho de fundos de investimento. Serão abordadas as características técnicas de implementação da busca, armazenamento e processamento de dados, através das tecnologias selecionadas de acordo com as necessidades apresentadas. Serão apresentados os resultados dos estudos realizados quanto à técnica de classificação de fundos utilizada, o *K-means*, bem como os resultados obtidos na técnica de remoção de *outliers* proposta, a qual utiliza a Distância de Mahalanobis.

Sumário

CAPÍTULO 1

Introdução

Um fundo de investimento é uma ferramenta financeira de aplicação que reúne recursos de um conjunto de investidores, visando a obtenção de ganhos financeiros a partir de uma carteira de investimentos. Os ativos que compõem esta carteira são selecionados após a definição de uma estratégia de atuação do fundo, baseada em análises quantitativas e qualitativas. A diversidade de ativos utilizados como aplicação nesta carteira é muito grande: ações, títulos públicos, fundos, dentre outros. A estratégia e o segmento em que um fundo atua podem ser representados pelas classificações (categorias) divulgadas por diversas fontes que atuam no mercado financeiro, como a *CVM* [1] e a *ANBIMA* [2].

Dispomos de alguns indicadores financeiros que nos permitem realizar análises individuais (rentabilidade, volatilidade etc), performance em relação a um *benchmark* [3], comparação entre dois ou mais fundos, e assim por diante. Unindo a criatividade e o conhecimento por parte do investidor, podemos ter uma análise mais concreta sobre o comportamento dos fundos.

Para os investidores ou até mesmo para o gestor de um fundo, é interessante ter a possibilidade de compará-lo aos demais fundos que possuem estratégias semelhantes ou que se enquadrem em uma mesma categoria. Além das fontes citadas anteriormente, pode ser ainda mais interessante que estas categorias sejam definidas com base em análises parametrizadas de acordo com o objetivo de classificação.

O objetivo deste trabalho é realizar a implementação de uma ferramenta que colete dados históricos dos fundos de investimento do mercado e realize o processo de classificação (*clusterização*), disponibilizando algumas estratégias estatísticas já conhecidas, como o método *k-means*, por exemplo, o qual foi implementado. Outro objetivo é disponibilizar recursos para a remoção de *outliers* de um grupo de fundos, através de um algoritmo que utiliza a Distância de Mahalanobis.

Uma das grandes preocupações na realização deste trabalho foi tornar esta ferramenta extremamente flexível e passível a inclusões de algoritmos e novas funcionalidades. Vale ressaltar que um dos grandes desafios encontrados foi a manipulação do grande volume de dados de mercado e como utilizá-los de forma eficiente, evitando o reprocessamento de algumas informações.

[1] http://www.cvm.gov.br/

[2] http://www.anbima.com.br

[3] Instrumento utilizado como meta a ser atingida

Adoratemos a CVM como fonte para os dados históricos dos fundos a serem utilizados nos processos. O processo de busca, alimentação e processamento das informações será explicado posteriormente, bem como as tecnologias utilizadas.

CAPÍTULO 2

Estudos Realizados

2.1 Estudo do Problema

Para entender um pouco mais sobre a classificação de fundos e algumas metodologias relacionadas a este processo, foram sugeridas, inicialmente, duas leituras. A primeira, uma dissertação referente à Estimação Dinâmica do *Beta* do Modelo CAPM em Fundo de Ações com Filtro de Kalman[5]. A segunda, um trabalho de conclusão de curso que envolve a construção de uma API para estimação de *Beta*s variáveis de fundos brasileiros[1].

O modelo CAPM (*Capital Asset Pricing Model*, ou Modelo de Precificação de Ativos Financeiros) é um modelo de precificação de ativos que busca estabelecer a relação entre retorno e risco de acordo, entre um ativo e uma carteira teórica. Muito utilizado no mercado financeiro, possui como um dos indicadores o *Beta*, que mede o risco do ativo em relação à carteira teórica não-diversificada, isto é, determina a sensibilidade do ativo (covariância) em relação ao mercado estudado.

Na prática, o *Beta* é o coeficiente angular da reta (regressão linear) gerada entre o ativo e a carteira que representa o mercado. Se o *Beta* é igual a 1, o ativo tende a acompanhar o mercado. Se o *Beta* é menor que 1, o ativo tende a oscilar menos que o mercado, no mesmo sentido. Se o *Beta* é maior que 1, o ativo tende a oscilar mais que o mercado, no mesmo sentido, proporcionalmente ao valor obtido.

2.2 Estimação Dinâmica do *Beta* do Modelo CAPM em Fundos de Ações: Uma Aplicação do Filtro de Kalman

A dissertação de Roberta Anchieta da Silva[5], traz como tema um estudo sobre o indicador *Beta* do modelo CAPM e a análise de sua invariância no tempo.

Inicialmente, foram definidos os fundos e o período a serem trabalhados. Foi realizada a análise de invariância do *Beta* (CAPM) no tempo. Utilizando o teste ADF sobre as séries temporais dos retornos calculados dos fundos, não foi rejeitada a premissa de estacionaridade das séries dos mesmos, continuando com a aplicação do modelo CAPM, a fim de avaliar o dinamismo do *Beta*.

Foi realizada a regressão linear para cada subperíodo definido, utilizando-se do indicador R^2 para concluir que havia um bom ajustamento dos retornos à própria regressão. Aplicando o modelo CAPM às séries de retornos, com diversas janelas de cálculo diferentes, chegou-se à conclusão de que o *Beta* variava ao longo do tempo. Além disso, foi realizado um teste de Chow para a quebra estrutural da série, o qual rejeitou também a hipótese de estabilidade de *Beta*.

Vale lembrar que o modelo CAPM tem como uma das premissas a invariância do *Beta* ao longo do tempo, já que um único *Beta* é calculado para o período completo. A partir deste momento, foi realizada uma modificação do modelo CAPM: ao invés de realizar apenas uma regressão, a qual geraria um único *Beta*, foram realizadas diversas regressões, onde o número de regressões fosse igual ao número de subperíodos definidos por uma janela, com 1 passo de unidade. Escolheram-se alguns subperíodos e, após a realização deste processo, juntamente com a aplicação dos testes ADF, foi difícil estabelecer uma conclusão, dado que, quanto menor a janela, maior a influência de um ponto da série, além do fato que foram utilizadas janelas pequenas.

Assim, recorreu-se ao Filtro de Kalman, um algoritmo recursivo de estimação sobre equações linearmente relacionadas, a fim de estimar o estado de um processo, geralmente não observável. Esta aplicação foi dividida em 3 partes: filtragem (visando obter a série de *Beta*s), aprendizagem (equação de filtragem) e solução (adotou-se o modelo *off-line*). No caso, supôs-se que os valores sucessivos de *Beta* são gerados a partir de um passeio aleatório (*Beta* anterior mais um choque aleatório), tendo em vista que as atitudes do gestor baseiam-se na carteira do dia anterior.

Utilizando o *Beta* do modelo CAPM como parâmetro de entrada (*Beta* médio), realizou-se o Filtro de Kalman juntamente com o teste ADF, concluindo que as séries podem ser consideradas estacionárias. Isto é um grande indício de que os gestores, mesmo baseando-se nas carteiras do dia anterior, possuem grande tendência de retornar à média.

2.3 API para Estimação de *Beta*s Variáveis

O Trabalho de Conclusão de Curso de Caio Ramos Casimiro[1], baseia-se na implementação de uma API para a análise de dados e estimação dos *Beta*s dinâmicos de fundos brasileiros.

Desenvolvido como um módulo de um sistema dentro de uma empresa, o trabalho conta com a utilização de diversas tecnologias semelhantes às que utilizarei neste projeto. Do ponto de vista técnico, este trabalho cita a metodologia de desenvolvimento utilizada (XP) e todas as características que envolvem um sistema de gestão de projetos.

Do ponto de vista prático, foram realizadas algumas análises com base nos dados dos fun-

dos. Além do algoritmo de agrupamento *k-means*, foram realizadas a Análise de Componentes Principais (PCA), que decompõe fundos em fatores de risco, e Regressões Lineares Múltiplas, sendo uma delas a Regressão Linear Múltipla Descontada.

CAPÍTULO 3

Classificação de Fundos

3.1 *k-means*

O método *k-means* é uma técnica de agrupamento de dados[2] apresentada por James MacQueen em 1967. Dada uma amostra de elementos a serem analisados, são definidos heuristicamente k centróides como base para o processo, isto é, o número de centros a serem definidos, que naturalmente é igual ao número de grupos formados. De acordo com uma medida de similaridade definida, os elementos da amostra são alocados em grupos associados aos centróides. Como próximo passo, os centróides de cada grupo são recalculados e é realizada uma nova iteração do processo de enquadramento. O processo é encerrado em 2 circunstâncias: uma nova iteração não surte realocações de elementos ou ao se atingir o número limite de iterações pré-definido.

Utilizamos no projeto a implementação deste método através da linguagem *R*, cuja integração ao sistema é explicada na seção 5.6.

3.2 Estudos e resultados obtidos

A fim de testar o comportamento do método, selecionamos uma amostra de fundos disponíveis no mercado, de acordo com a classificação fornecida pela CVM. Para tanto, foram escolhidos aleatoriamente 10 (dez) fundos das categorias Ações, Multimercado, Cambial e Renda Fixa, totalizando 40 (quarenta) fundos no estudo. Foram passados 4 (quatro) cenários de estudo, cada um deles possuindo 2 (dois) indicadores, dos quais foram definidos:

- Volatilidade e Retorno, na Figura 3.1 e Tabela 3.1
- *Beta*(CDI) e *Beta*(IBOV), na Figura 3.2 e Tabela 3.2
- *Beta*(CDI) e *Beta*(PTAX), na Figura 3.3 e Tabela 3.3
- *Beta*(IBOV) e *Beta*(PTAX), na Figura 3.4 e Tabela 3.4

Os dados foram calculados com valores reais e com data de referência para 30 de Setembro de 2010, sendo que o período escolhido para a análise é de 6 meses. Os nomes dos fundos foram alterados a fim de evitar quaisquer conclusões equivocadas sobre os mesmos.

Vale lembrar que o intuito desta seção não é avaliar o comportamento dos fundos e a classificação definida pela CVM. Conforme citado anteriormente, a obtenção de classificações depende de fatores como parametrizações, cenários e critérios de agrupamento estabelecidos pelo usuário final. Nosso objetivo principal é ilustrar as diferentes classificações obtidas de acordo com cada um dos parâmetros utilizados nos respectivos cenários.

Definindo como 4 (quatro) o número de centróides em todos os casos, obtemos os resultados a seguir:

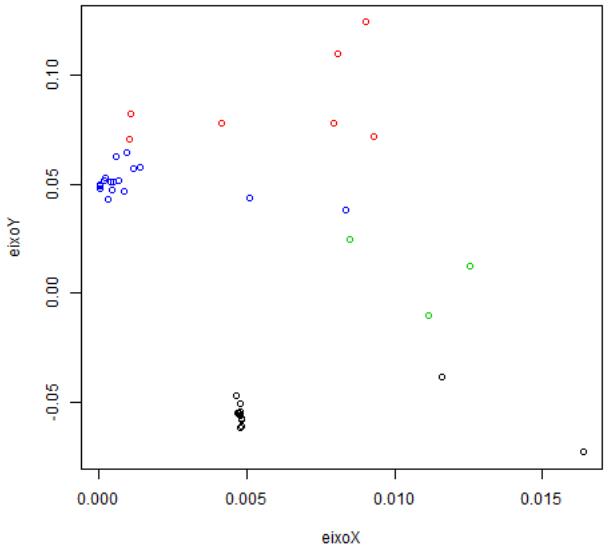

Figura 3.1 Volatilidade x Retorno

Realizando uma avaliação superficial dos resultados obtidos, notamos claramente (Tabela 3.4) que a utilização dos *Betas* referentes ao IBOVESPA e ao PTAX ofereceram um melhor ajuste às categorias definidas pela CVM, para esta amostra de fundos. Ao realizar a mesma experiência com um volume maior de fundos (eventualmente com todos os fundos da indústria), é esperado que os fundos alternem entre os grupos, como ocorrido nos três primeiros casos de estudo.

Vale a pena ressaltar a variabilidade no agrupamento dos fundos, de acordo com o caso escolhido. Eventualmente, realizar este mesmo estudo com diversos períodos, para uma mesma data de referência, poderá auxiliar na percepção de comportamentos similares para determinados grupos de fundos.

Tabela 3.1 Volatilidade x Retorno

Grupo1	Grupo2	Grupo3	Grupo4
ACOES1	ACOES2	ACOES4	ACOES5
ACOES9	ACOES3	ACOES6	MULTIM1
CAMBIAL1	ACOES7	ACOES8	MULTIM2
CAMBIAL2	ACOES10		MULTIM3
CAMBIAL3	MULTIM4		MULTIM7
CAMBIAL4	MULTIM5		MULTIM8
CAMBIAL5	MULTIM6		MULTIM9
CAMBIAL6			MULTIM10
CAMBIAL7			RF1
CAMBIAL8			RF2
CAMBIAL9			RF3
CAMBIAL10			RF4
			RF5
			RF6
			RF7
			RF8
			RF9
			RF10

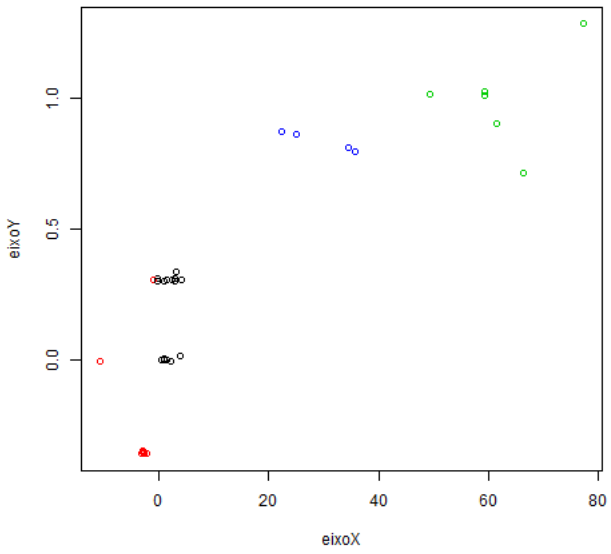

Figura 3.2 *Beta*(CDI) x *Beta*(IBOV)

Tabela 3.2 *Beta*(CDI) x *Beta*(IBOV)

Grupo1	Grupo2	Grupo3	Grupo4
MULTIM1	MULTIM5	ACOES1	ACOES2
MULTIM2	CAMBIAL1	ACOES6	ACOES3
MULTIM3	CAMBIAL2	ACOES7	ACOES4
MULTIM4	CAMBIAL3	ACOES8	ACOES5
MULTIM6	CAMBIAL4	ACOES9	
MULTIM7	CAMBIAL5	ACOES10	
MULTIM8	CAMBIAL6		
MULTIM9	CAMBIAL7		
MULTIM10	CAMBIAL8		
RF1	CAMBIAL9		
RF2	CAMBIAL10		
RF3	RF5		
RF4			
RF6			
RF7			
RF8			
RF9			
RF10			

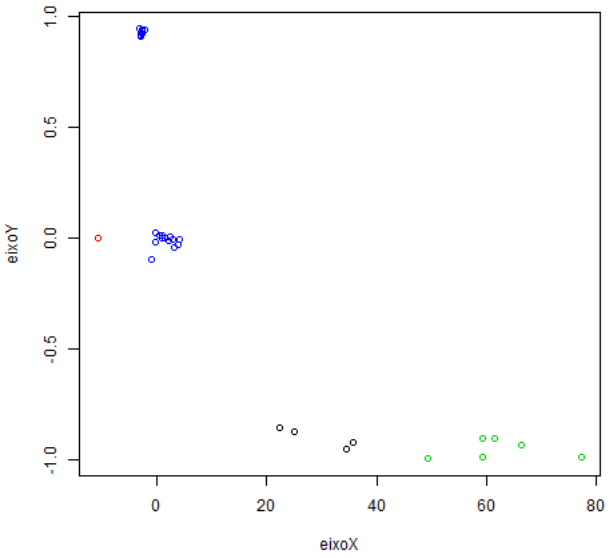

Figura 3.3 *Beta*(CDI) x *Beta*(PTAX)

Tabela 3.3 *Beta*(CDI) x *Beta*(PTAX)

Grupo1	Grupo2	Grupo3	Grupo4
ACOES2	RF5	ACOES1	MULTIM1
ACOES3		ACOES6	MULTIM2
ACOES4		ACOES7	MULTIM3
ACOES5		ACOES8	MULTIM4
		ACOES9	MULTIM5
		ACOES10	MULTIM6
			MULTIM7
			MULTIM8
			MULTIM9
			MULTIM10
			CAMBIAL1
			CAMBIAL2
			CAMBIAL3
			CAMBIAL4
			CAMBIAL5
			CAMBIAL6
			CAMBIAL7
			CAMBIAL8
			CAMBIAL9
			CAMBIAL10
			RF1
			RF2
			RF3
			RF4
			RF6
			RF7
			RF8
			RF9
			RF10

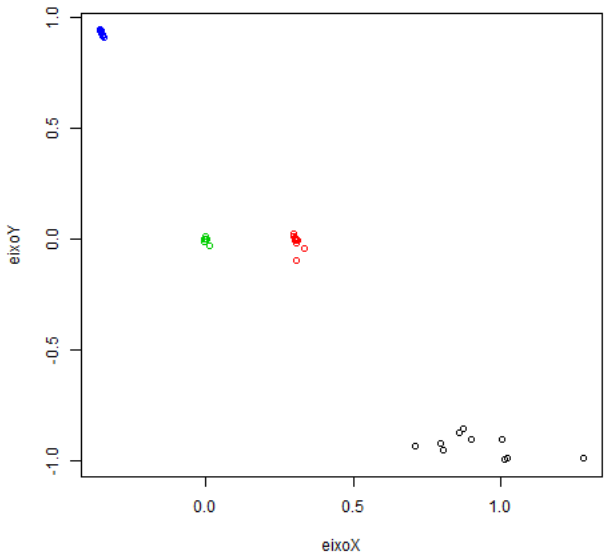

Figura 3.4 *Beta*(IBOV) x *Beta*(PTAX)

Tabela 3.4 *Beta*(IBOV) x *Beta*(PTAX)

Grupo1	Grupo2	Grupo3	Grupo4
ACOES1	MULTIM1	RF1	CAMBIAL1
ACOES2	MULTIM2	RF2	CAMBIAL2
ACOES3	MULTIM3	RF3	CAMBIAL3
ACOES4	MULTIM4	RF4	CAMBIAL4
ACOES5	MULTIM5	RF5	CAMBIAL5
ACOES6	MULTIM6	RF6	CAMBIAL6
ACOES7	MULTIM7	RF7	CAMBIAL7
ACOES8	MULTIM8	RF8	CAMBIAL8
ACOES9	MULTIM9	RF9	CAMBIAL9
ACOES10	MULTIM10	RF10	CAMBIAL10

CAPÍTULO 4

Remoção de *Outliers*

4.1 Distância de Mahalanobis

Ao realizar análises em grupos (categorias) com um razoável número de fundos, é comum que seja detectada a presença de *outliers*, isto é, observações numericamente distantes da tendência do grupo analisado. Este tipo de situação prejudica demasiadamente análises gráficas, como gráficos de dispersão ou *box-plot*, por exemplo, e até mesmo indicadores estatísticos de um determinado grupo.

Diante disso, torna-se interessante examinar as categorias obtidas em particular, buscando melhorar a qualidade de dados dentro de cada grupo. A intenção é que estes *outliers* sejam detectados, de acordo com alguma parametrização pré-estabelecida, e o próprio usuário defina quais deles deverão ser removidos do estudo apresentado. Vale ressaltar que um *outlier* de uma determinada análise pode não necessariamente ser *outlier* em uma análise com parâmetros distintos, porém para o mesmo grupo. A variação de parâmetros como período e janela de cálculo podem influenciar significativamente as saídas geradas para um mesmo grupo de fundos.

Uma solução proposta neste projeto é a utilização da distância de Mahalanobis[4] como ferramenta para detecção de *outliers*. Ela difere da distância Euclidiana por considerar a matriz de correlação entre os elementos envolvidos na análise. No caso particular em que esta matriz de correlação é a matriz identidade, a distância de Mahalanobis é a própria distância Euclidiana.

Neste projeto, utilizamos a distância de Mahalanobis para 2 (duas) variáveis, que serão representadas por 2 (dois) indicadores definidos pelo próprio usuário. Além disso, utilizamos um fator de multiplicação que define a sensibilidade na remoção dos *outliers*.

4.2 Estudos e resultados obtidos

Para estudar o comportamento do algoritmo de detecção e remoção de *outliers*, foi realizado um estudo com 472 fundos, representados inicialmente pela figura 4.1. Foram utilizados como indicadores a volatilidade e o retorno dos fundos, num período de 6 meses.

Para ilustrar a percepção de remoção obtida graficamente e a sensibilidade de remoção gerada pelo fator de multiplicação, foram adotados os seguintes fatores:

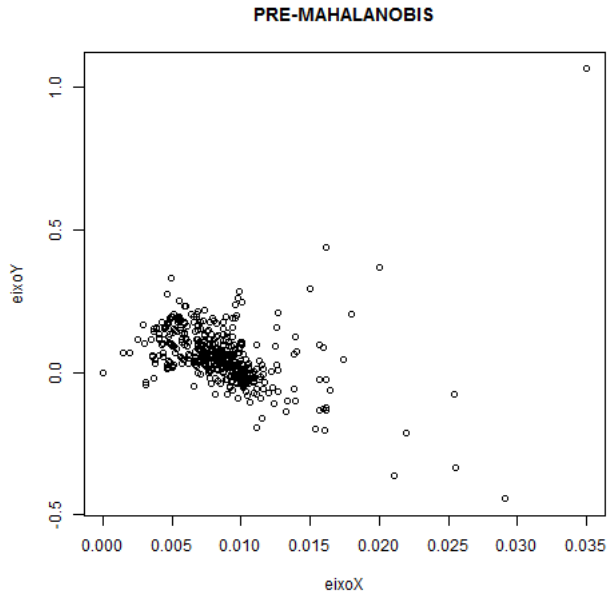

Figura 4.1 Fundos antes da aplicação do algoritmo

- Fator 10, na Figura 4.2
- Fator 5, na Figura 4.3
- Fator 2, na Figura 4.4
- Fator 1, na Figura 4.5

Os dados foram calculados com valores reais e com data de referência para 30 de Setembro de 2010.

O número de fundos removidos em cada um dos casos pode ser visualizado na tabela 4.1.

Tabela 4.1 Quantidade de fundos removidos

Fator=10	Fator=5	Fator=2	Fator=1
1	4	32	190

Notamos que, quanto mais próximo de zero, mais o fator de remoção tende a detectar a presença de *outliers*. A idéia é que o usuário utilize um critério de ajustes no fator, de acordo com os resultados visuais apresentados, ou até mesmo com base em indicadores estatísticos

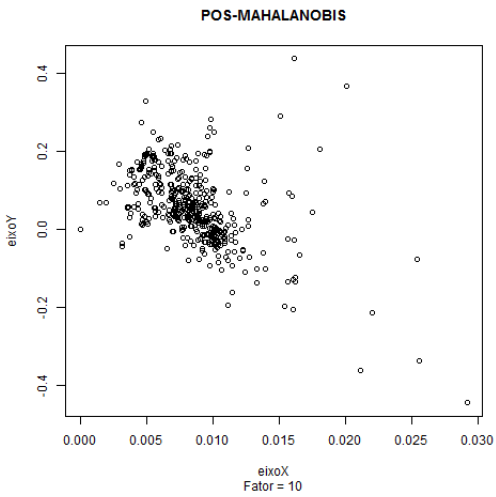

Figura 4.2 Remoção com fator = 10

por ele determinados. A princípio, este processo de aprendizado poderia ser realizado por um processo de *Machine Learning*, citado na seção "Conclusão".

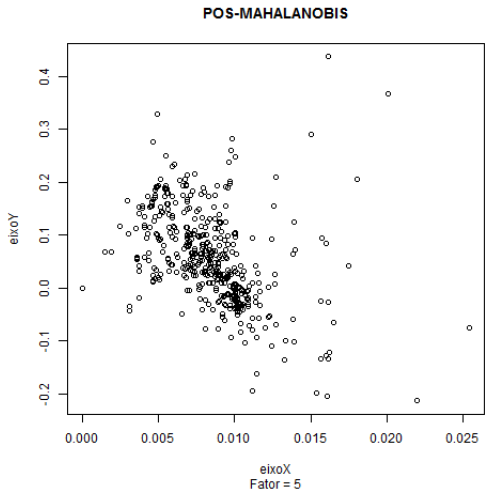

Figura 4.3 Remoção com fator = 5

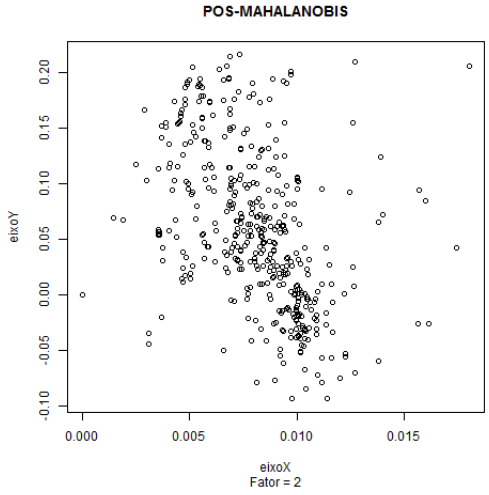

Figura 4.4 Remoção com fator = 2

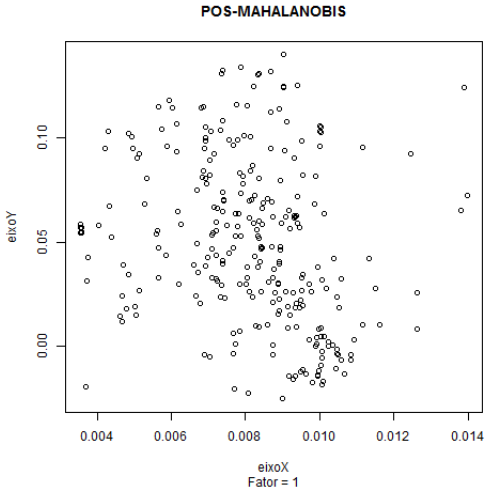

Figura 4.5 Remoção com fator = 1

CAPÍTULO 5

Sistema: *Fund Clustering*

Neste capítulo serão abordadas detalhadamente as estruturas envolvidas em cada um dos módulos do projeto, desde a alimentação da base de dados, até a interface disponibilizada para o usuário final.

5.1 Mapa do Projeto

Na figura 5.1 pode-se visualizar a estrutura geral do projeto:

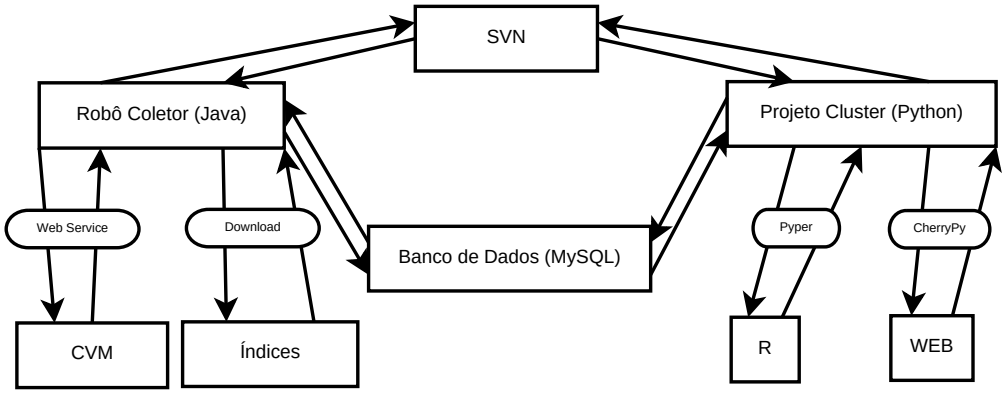

Figura 5.1 Diagrama do Projeto

5.2 Coleta de Dados: Robôs

Para a coleta dos dados de mercado foram criados robôs automatizados na linguagem *Java* [1]. A princípio, o ideal é que a coleta seja realizada diariamente, utilizando a biblioteca *Quartz* [2] para agendamento de tarefas. Para a modelagem das entidades e controle de persistência, foi utilizado o framework *Hibernate/JPA* [3].

[1] http://www.java.com
[2] http://www.quartz-scheduler.org/
[3] http://www.hibernate.org/

Os dados cadastrais dos fundos do mercado e suas respectivas séries de cota e patrimônio líquido foram coletados da *CVM* [4], através de *Web Services*. Já a coleta das séries de índices do sistema varia conforme a fonte. Foi criada uma estrutura que permite facilmente a inclusão de novos índices e robôs responsáveis pela coleta dos mesmos. Neste trabalho, foi implementado como exemplo um coletor do valor diário de fechamento do índice Ibovespa, divulgado pela BMF&BOVESPA [5]. Neste caso, é realizado o *download* de um arquivo texto disponibilizado diariamente com as informações de ações, índices e outros elementos de mercado.

5.3 Estrutura Interna

O sistema *Fund Clustering* foi desenvolvido na linguagem *Python* [6]. Sua estrutura interna está segmentada em diversos módulos, sendo eles responsáveis pela comunicação com a base de dados, cálculo de indicadores, classificação de fundos, tratamento de *outliers*, análises, administração e camada *web*, detalhados a seguir. A maioria dos módulos possuem classes que tem como objetivo encapsular as tarefas a serem realizadas, como *input*(entrada), processamento e *output*(saída) de dados.

5.4 Módulo: comunicação com a base de dados

No módulo de comunicação com a base de dados, foram criadas inicialmente todas as classes que representam as entidades utilizadas no sistema. Foram implementados dois módulos para a realização de consultas e persistência. Uma das preocupações foi realizar transações rápidas e que realizassem os *commits* sempre que possível.

5.5 Módulo: Interface *WEB*

O sistema possui uma interface *WEB*, permitindo a navegação e utilização por qualquer usuário que possua uma conexão e um navegador.Foi implementado um módulo responsável pelo gerenciamento de navegação e controle de informações de sessões de usuários. Ele também é responsável por chamar as principais funções dos outros módulos que compõem o sistema, apresentados a seguir.

Abaixo, vemos na figura 5.2 a tela de *login* do sistema.

Na tela inicial, representada pela figura 5.3, temos acesso aos principais módulos do sis-

[4] http://www.cvm.gov.br/
[5] http://www.bmfbovespa.com.br/ibovespa
[6] http://www.python.org/

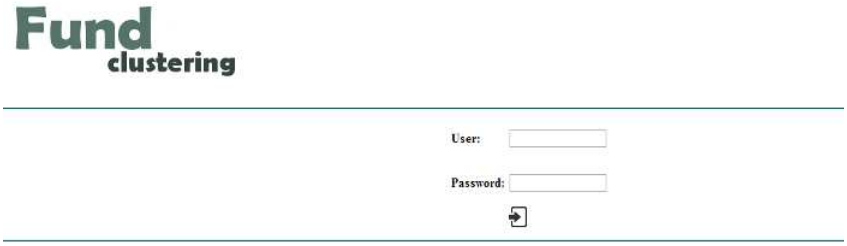

Figura 5.2 Tela de *login*

tema. Este acesso também é facilitado por um menu superior à direita, onde o nome do usuário que está logado é apresentado.

Figura 5.3 Tela inicial

5.6 Módulo: classificação de fundos

Para a classificação de fundos de investimento, foi implementado um módulo responsável pela aplicação de algoritmos de agrupamento em um conjunto de informações quantitativas passadas como parâmetros. Nesta versão do projeto, o algoritmo proposto foi o *k-means*[2]. O módulo responsável pela aplicação deste algoritmo recebe como parâmetros de entrada uma lista de fundos e um número de grupos a serem definidos. Para cada fundo, são informados dois valores referentes a dois indicadores previamente definidos pelo usuário final, para que o *k-means* seja realizado bidimensionalmente.

Como ferramenta para aplicação deste algoritmo, utilizamos o auxílio da plataforma *R* [7]. Para isto, foi necessário utilizar uma integração direta entre o *Python* e o *R*. Esta integração foi facilitada pela utilização de uma biblioteca chamada *PypeR* [8], a qual permite a realização de chamadas diretas no ambiente *R*. O interessante de trabalhar com esta biblioteca é a possibilidade do aproveitamento de recursos e algoritmos de otimização já implementados em *R*, o qual possui uma vasta comunidade de colaboradores envolvidos.

Como *input*(entrada), o sistema aceita um arquivo texto com as informações a serem processadas. Este *upload*(carregamento) é realizado pelo usuário, conforme visto na figura 5.4.

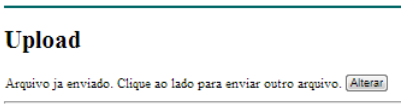

Figura 5.4 Upload de arquivo

O *output*(saída) pode ser visualizado em gráfico e valores, disponibilizados em tela e em arquivo texto, respectivamente, com as informações detalhadas dos grupos obtidos e cada um dos fundos enquadrados. A figura 5.5 representa parte da tela deste módulo, após o processamento de informações.

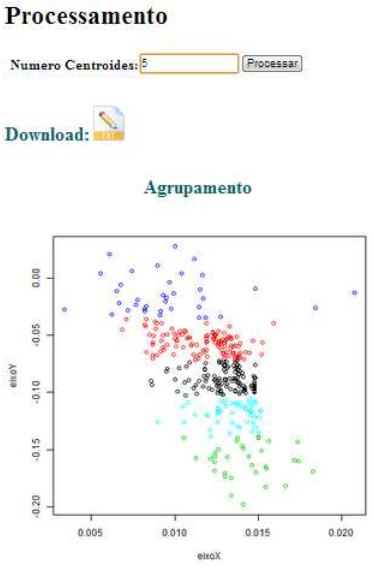

Figura 5.5 Parte da tela de classificação

[7] http://www.r-project.org/
[8] http://www.webarray.org/softwares/PypeR/

Já na figura 5.6 vemos parte do arquivo texto gerado como *output*(saída) na análise, com o detalhamento dos grupos obtidos e seus respectivos centróides.

```
#########################################################
####  ANALISE GERADA PELO SISTEMA FUND CLUSTERING  ####
#########################################################


#########################################################
####                    PARAMETROS                   ####
#########################################################

CENTROIDES: 4


#########################################################
####                GRUPOS DEFINIDOS                 ####
#########################################################



---------------------------------------------------------
GRUPO: 0
TAMANHO: 2 itens
CENTROIDE: (0.0086452;0.01894295)
---------------------------------------------------------

FUNDO;VALOR_X;VALOR_Y
5594765000110;0.0061051;0.0212265
5775774000108;0.0111853;0.0166594


---------------------------------------------------------
GRUPO: 1
TAMANHO: 4 itens
CENTROIDE: (0.010680725;-0.007147525)
```

Figura 5.6 Saída em texto gerada pelo *k-means*

Ainda falando sobre o módulo de classificação, vale a pena ressaltar a flexibilidade de serem acrescentados algoritmos de classificação em versões posteriores do projeto. A idéia é que o usuário final possa escolher o algoritmo que julgue mais adequado para determinada análise, ou até mesmo realizar estudos de variabilidade de classificação para um determinado grupo de fundos. Basicamente, a forma em que este módulo foi construído permite ao desenvolvedor implementar apenas um módulo para um novo algoritmo e realizar pequenos ajustes na interface *web*, a fim de possibilitar a escolha do mesmo.

5.7 Módulo: remoção de *outliers*

Seguindo os mesmos moldes do módulo de classificação de fundos, o módulo desenvolvido para o estudo da remoção de *outliers* teve como objetivo ser extremamente flexível à inclusão de novos algoritmos para este tipo de ferramenta. Neste versão do projeto, utilizamos como suporte o conceito da Distância de Mahalanobis para a remoção de *outliers*[4].

Novamente, são passados como parâmetros de entrada uma lista de fundos e um fator de corte para a aplicação do algoritmo, além de dois valores referentes aos indicadores previamente definidos pelo usuário final. Um módulo responsável pela aplicação do algoritmo recebe

estas informações e aplica um filtro, marcando os elementos que foram julgados como *outliers*, para posterior análise e exportação.

Neste caso, também utilizamos uma pequena integração com o ambiente *R* para a geração dos gráficos de saída, gerados pela dispersão e colorização das informações processadas. Como *input*(entrada), o sistema aceita um arquivo texto com as informações a serem processadas. O *output*(saída) pode ser visualizado em gráfico e valores, disponibilizados em tela e em arquivo texto, respectivamente, com os fundos que foram mantidos no grupo após a aplicação do algoritmo.

A figura 5.7 representa parte da tela visualizada após a aplicação do algoritmo de remoção de *outliers*.

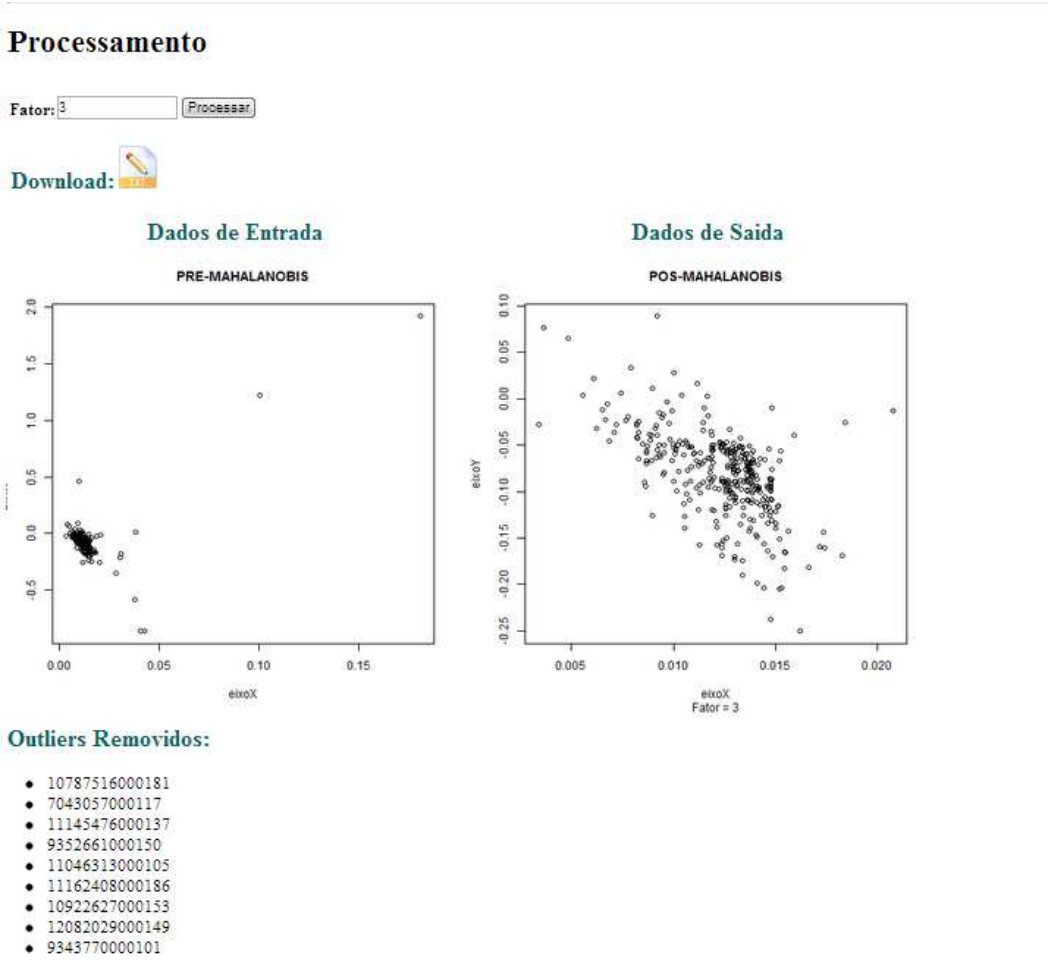

Figura 5.7 Parte da tela de remoção de *outliers*

Já na figura 5.8 vemos parte do arquivo texto gerado como *output* na análise, com o

detalhamento de fundos que foram mantidos e removidos para os parâmetros fornecidos.

```
#########################################################
####  ANALISE GERADA PELO SISTEMA FUND CLUSTERING  ####
#########################################################


#########################################################
####               FUNDOS PRESERVADOS                ####
#########################################################

000000000001
000000000002
000000000003
000000000004
000000000005
000000000006
000000000007


#########################################################
####          FUNDOS REMOVIDOS (OUTLIERS)           ####
#########################################################

000000000010
000000000021
000000000061
000000000063
000000000087
000000000102
000000000119
000000000150
```

Figura 5.8 Saída em texto gerado pela remoção

5.8 Módulo: cálculo de indicadores

Já o módulo de cálculo de indicadores foi criado de forma a permitir facilmente a inclusão de novos indicadores para versões posteriores do sistema, bastando criar um módulo novo para cada indicador acrescentado, com sua respectiva metodologia. Esta facilidade foi proporcionada com a criação de classes que encapsulam as séries dos fundos, indicadores e parâmetros utilizados em uma determinada análise. Nesta versão, foram disponibilizados os cálculos referentes ao retorno contínuo, volatilidade (desvio padrão) e *beta*, explicados a seguir, com base na leitura [3].

Seja c o vetor que representa a série de cotas de um determinado fundo de investimento. Tomemos j um número inteiro, $j > 1$, como a janela de cálculo da série de retornos contínuos deste fundo, a qual chamaremos de r. A construção de r é dada por:

$$r_i = ln(\frac{c_i}{c_{i-j}})$$

Analogamente, podemos calcular a série de retornos contínuos de um *benchmark* a partir da sua série diária de índices.

A volatilidade é uma das medidas de risco mais utilizadas no mercado financeiro, que pode ser representada pelo cálculo do desvio padrão amostral da série de retornos contínuos do ativo r, isto é:

$$volatilidade = \sigma(r) = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^{n} (r_i - \bar{r})^2}$$

Sejam r_a e r_m as séries de retorno contínuo do ativo e da carteira de mercado (a qual trataremos como *benchmark*), respectivamente, para um determinado período. Utilizando os conceitos estatísticos de covariância e variância, o *Beta* pode ser calculado da seguinte forma:

$$\beta = \frac{Cov(r_a, r_m)}{Var(r_m)}$$

5.9 Recuperação de Dados: valores pré-calculados

Como citado na introdução, um dos grandes desafios é a manipulação do grande volume de dados em um relativo curto espaço de tempo. Ao realizarmos qualquer análise quantitativa de um fundo, o primeiro passo se dá na recuperação de duas séries de cotas e patrimônios líquidos, o qual já demanda um certo custo. Posteriormente, de acordo com os parâmetros utilizados, tais como tipo de indicador, período, dentre outros, estes dados são processados através das calculadoras do sistema, gerando os valores finais dos indicadores, também demandando um certo custo.

Imaginemos um cenário em que um sistema desses, em ambiente de produção, estivesse atendendo um grande número de usuários que, por sua vez, realizariam de forma disjunta análises de mesma característica. Além do tempo de espera já previsto no parágrafo anterior, teríamos as questões de concorrência e, especificamente neste caso, o reprocessamento desnecessário de análises muito próximas ou, até mesmo, iguais.

Dado que os valores históricos do mercado não são alterados, exceto por correções de erros de divulgação, pois trabalham com o fechamento dos dias de divulgação, um dos recursos implementados foi a utilização de indicadores pré-calculados para as análises. A idéia é que, determinada uma matriz de indicadores e períodos pré-definidos, além de uma data de referência, cada um dos elementos desta matriz sejam calculados e persistidos na base de dados, para todos os fundos disponíveis na base de dados, permitindo uma posterior recuperação destas

informações por parte dos usuários. Assim, a grande maioria dos procedimentos de cálculo citados anteriormente pode ser evitada. Para isto, foram criadas duas tabelas que representam os fundos e seus indicadores calculados para uma determinada data e parâmetros, como o período, por exemplo.

Com o objetivo de atender grande parte das análises mais comuns realizadas pelos usuários de mercado financeiro, basta que o administrador do sistema escolha datas consideradas como "chave"para as análises de mercado, como fechamentos de mês, inícios de quinzenas, dentre outros, e pré-processar estes valores para os usuários finais.

Vale ressaltar que uma alternativa que poderia ser aplicada na recuperação e tratamento de séries é a utilização de *caches*, onde as séries podem ser guardadas de acordo com critérios de definição de quais são os fundos mais consultados.

5.10 Módulo: Análises

Para que o usuário possa realizar as consultas dos instrumentos disponibilizados no sistema, foi criado um módulo de consulta e análise. Para a parte de fundos, por exemplo, o usuário poderá buscar tanto informações qualitativas, como categoria *CVM*, data de início, dentre outros, como informações quantitativas, como as séries de cotas e patrimônios líquidos, por exemplo, além dos indicadores pré-calculados, se estes existirem. A figura 5.9 apresenta os dados obtidos após a realização de uma determinada consulta.

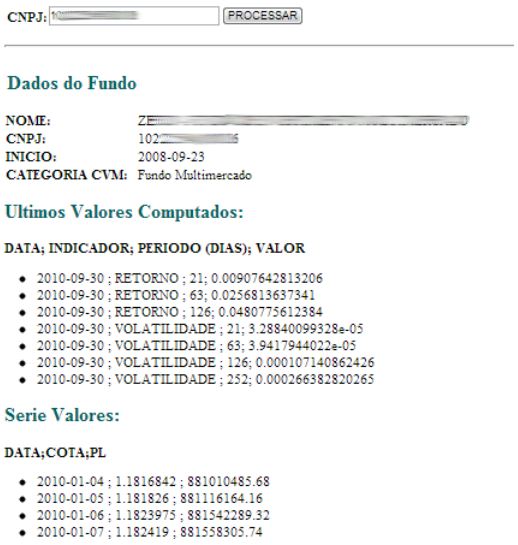

Figura 5.9 Parte da tela de consulta de fundos

Já para a parte de índices é possível realizar a consulta do cadastro de índices e suas respectivas séries de valores diários divulgados, como vemos na figura 5.10.

Figura 5.10 Parte da tela de consulta de índices

Além disso, o usuário poderá processar relatórios com os indicadores pré-processados pelo administrador do sistema, bastando realizar um *upload* de arquivo num formato especificado.

Figura 5.11 Parte da tela de relatórios

Já na figura 5.12 vemos parte do arquivo texto gerado como *output* na análise, com o detalhamento de todos os parâmetros utilizados, além dos valores referentes aos fundos, períodos e indicadores escolhidos.

```
##########################################################
####  ANALISE GERADA PELO SISTEMA FUND CLUSTERING  ####
##########################################################


##########################################################
####                    PARAMETROS                    ####
##########################################################

DATA: 2010-09-30;

PERIODOS: 1M;3M;6M;

INDICADORES: RETORNO;VOLATILIDADE;


##########################################################
####                VALORES OBTIDOS                   ####
##########################################################


FUNDO;INDICADOR;PERIODO;VALOR
[illegible]1;RETORNO;1M;0.0101197563783
[illegible]1;RETORNO;3M;0.0288427490735
[illegible]1;RETORNO;6M;0.0501897062512
[illegible]0;RETORNO;1M;0.112846269556
[illegible]0;RETORNO;3M;0.159619826813
[illegible]0;RETORNO;6M;0.0910464373292
[illegible]47;RETORNO;1M;0.0704061641316
```

Figura 5.12 Output em texto do relatório

5.11 Módulo: Administrador

Este módulo tem como objetivo fornecer as principais ferramentas de controle para um administrador do sistema. Nesta versão do projeto, foram disponibilizadas algumas ferramentas que representam estas funções, como a inserção de novos índices, o upload de séries e o pré-processamento de indicadores. Parte desta tela pode ser visualizada na figura 5.13.

Figura 5.13 Parte da tela de administrador

5.12 Manuais

A fim de facilitar a utilização do sistema, foi criado um manual de usuário. Este documento explica detalhadamente quais os objetivos de cada módulo, além de como o usuário poderá utilizar cada funcionalidade do sistema. O ideal é que este documento seja atualizado a cada versão disponbilizada.

Além disso, foi criado um manual para desenvolvedores, que explica como criar o ambiente para utilização do sistema e dos robôs, bem como as tecnologias a serem instaladas para tal funcionamento. Este manual também contém as informações básicas sobre a estrutura do projeto e quais os módulos responsáveis por cada uma das principais funcionalidades do sistema.

CAPÍTULO 6

Conceitos e Tecnologias Utilizadas

6.1 Web Service

Para a coleta do cadastro e dados históricos de fundos, a CVM disponibiliza um *Web Service*. Um *Web Service* é um tipo de serviço que permite a integração de sistemas, em plataformas distintas, através de uma interface normalizada. A comunicação é feita através de arquivos do tipo *XML* [1]. Para descrever os serviços e a interface de utilização de um *Web Service*, um documento WSDL deve ser criado. Este também está escrito em XML e funciona como uma espécie de contrato para que as respectivas implementações de clientes sejam realizadas em cada uma das plataformas.

Atualmente os *Web Services* são muito utilizados em ambientes corporativos, tanto para soluções externas quanto para soluções internas. Além de proporcionar um maior dinamismo em ambientes de desenvolvimento, possui uma grande vantagem quando tratamos de custos financeiros de implementação, já que pode ser facilmente reaproveitado.

6.2 MySQL

O MySQL [2] é um dos mais populares sistemas de gerenciamento de banco de dados (SGBD). Além de ser de código aberto, pode ser utilizado em mais de 20 plataformas diferentes.

6.3 Java

Para a construção dos robôs de coleta de dados, foi escolhida a linguagem de programação Java [3]. Além de já ter familiaridade com esta linguagem, existem diversas bibliotecas que permitem uma rápida implementação e integração de conceitos. Alguns exemplos que podem ser citados são a geração do sistema cliente de *Web Service* (item 2.1), o agendamento de tarefas (biblioteca Quartz), fácil manipulação de dados extraídos do banco de dados (API JPA),

[1] Linguagem utilizada para representação de dados, independente de plataforma.
[2] http://www.mysql.com/
[3] http://www.java.com/

dentre outros.

6.4 Python

Para a implementação do sistema de cálculos e classificação dos fundos, foi escolhida a linguagem Python [4], que é uma linguagem interpretada e que possui uma grande vantagem na questão de performance em relação ao Java. Além de ser muito dinâmica e proporcionar uma rápida implementação, é facilmente integrável à linguagem R, através da biblioteca Pyper, citada posteriormente.

6.5 R

Para análise de dados e realização dos cálculos utilizados neste trabalho, escolhemos a linguagem R [5], uma das mais utilizadas para cálculos estatísticos e geração de gráficos. Além de permitir a inclusão de pacotes com algoritmos avançados para uma melhor utilização, possui certa robustez no processamento e análise dos dados, o que a torna uma boa opção como um módulo de cálculo na integração de sistemas que trabalham com grande volume de dados a serem analisados.

6.6 Pyper

Pyper [6] é uma biblioteca escrita em Python, que permite uma fácil integração com o R. Para sua utilização, basta criar um objeto R em Python e utilizá-lo como a interface para a comunicação com o ambiente R, que ocorre internamente por meio de Pipes.

6.7 Flask

Para o desenvolvimento da camada de interface *WEB* foi utilizado *framework Flask* [7]. Além de ser extremamente leve e de permitir uma rápida e fácil implementação, possui uma documentação simples e com grande volume de informações.

[4] http://www.python.org/
[5] http://www.r-project.org/
[6] http://www.webarray.org/softwares/PypeR/
[7] http://flask.pocoo.org/

6.8 SVN

Como sistema de controle de versão utilizamos o SVN [8], um sistema de código aberto muito utilizado. Além de gerenciar os arquivos cronologicamente, permitindo comparar as diferenças entre as versões, possui algumas vantagens para desenvolvimento em equipes grandes, como a facilidade na realização de *merge* e separação de *commits*.

[8] http://subversion.apache.org/

CAPÍTULO 7

Conclusões

De acordo com a proposta desenvolvida inicialmente, este trabalho mescla conhecimentos de ciência da computação, mercado financeiro e aplicações das ferramentas envolvidas com estas áreas. Desenvolver um produto que atendesse aos requisitos propostos foi um desafio, dado que o projeto deveria ser dividido em camadas e frentes de trabalho distintas.

A parte inicial de estudos foi fundamental para a compreensão das ferramentas a serem exploradas e os conceitos a serem trabalhados. Apesar de consumir um tempo significativo, foi ela que deu a base para o entendimento do projeto como um todo. Além disso, foi de fundamental importância na avaliação das experiências que já haviam sido realizadas, facilitando o caminho de alguns pontos de desenvolvimento.

No que tange à parte técnica, o desenvolvimento das camadas de alimentação de base de dados (robôs automatizados), camadas internas do sistema (como organização dos módulos de cálculo e análises utilizadas, por exemplo) e a camada representada pela interface ao usuário, proporcionou uma visão geral de um projeto completo, imprimindo um produto final que poderá ser explorado no futuro.

Os estudos realizados para a classificação de fundos e remoção de *outliers* apresentaram um resultado paralelo à expectativa inicial. Além de possibilitarem a visualização na prática da aplicação das ferramentas apresentadas, mostraram-se bem ajustados aos dados utilizados como entrada, isto é, dados reais de mercado.

Muitas outras idéias surgiram no decorrer da implantação do projeto. Dentre elas, podemos citar a utilização de *caches* para a recuperação das séries de dados mais utilizadas, implementar um processo de aprendizado *Machine Learning* para a estimação dos fatores de entrada (como o número de centróides e o fator de multiplicação na distância de Mahalanobis), inclusão de indicadores (calculadoras) por parte do usuário (definindo uma linguagem específica a ser interpretada), dentre outros. Acredito que todas estas vertentes possam ser exploradas numa continuação futura deste trabalho, desenvolvendo um sistema mais completo e com grande potencial de utilização.

CAPÍTULO 8

Avaliação Subjetiva

8.1 Dificuldades encontradas

Inicialmente, uma das grandes dificuldades foi trabalhar com o tratamento da base de dados, dado o volume de informações de fundos de investimento e eventuais falhas de divulgação, como valores incorretos e até mesmo a falta de informação para algumas datas. O *Web Service* utilizado no sistema para a busca de dados possui um serviço instável e dificilmente corrige retroativamente erros de divulgação.

Outra dificuldade foi a adaptação ao *Python* e seus *frameworks*. Acostumado a trabalhar com *Java*, tive uma certa dificuldade em assimilar a utilização da linguagem no início, dado o escopo do trabalho que eu pretendia realizar. Porém, acredito que tenha conseguido exercer boas práticas de programação na implementação do projeto.

Podemos citar o fator *tempo* como um dos responsáveis não pela dificuldade, mas sim pelo sentimento de que poderia dar uma continuidade neste projeto que, ao meu ver, poderia ser bem explorado não só no mercado financeiro, como em outros setores de atividades que trabalham com agrupamento e detecção de *outliers*.

8.2 Disciplinas Utilizadas no Trabalho

De forma geral, neste trabalho passamos por várias disciplinas oferecidas durante o curso, como por exemplo:

- **MAC110**(Introdução à Computação) e **MAC122**(Princípios de Desenvolvimento de Algoritmos): acredito que estas matérias foram importates para a criação de uma "cultura"de boas práticas de programação, dado que desde o início do curso temos a preocupação de realizar um desenvolvimento "limpo"e com a preocupação de maior eficiência.
- **MAE121** e **MAE212**(Introdução à Probabilidade e Estatística I e II) : entender os modelos e análises estatísticas é fundamental para qualquer pessoa que deseja trabalhar com ferramentas do mercado financeiro. Podemos detectar diversas vezes a presença de alguns destes conceitos (covariância, desvio padrão, dentre outros) neste trabalho.
- **MAC0323**(Estruturas de Dados): manipular os dados de forma a utilizar as estruturas

mais adequadas é parte fundamental de um projeto bem sucedido. Neste trabalho, por exemplo, trabalhamos com as noções de encapsulamento e manipulação de listas.

- **MAC0426**(Sistemas de Bancos de Dados): a modelagem de uma base de dados e manipulação de consultas e comandos a ela relacionadas exigem um conhecimento que foi apresentado nesta disciplina.

- **MAC0333**(Armazenamento e Recuperação de Informação): o armazenamento e a recuperação de dados do mercado financeiro, processados ou não, podem ser discutidos com base nos conceitos trabalhados por esta disciplina. Um exemplo é a utilização de valores pré-calculados para a redução do tempo de espera para o usuário final.

  Acredito que todas as disciplinas do curso oferecem direta ou indiretamente ferramentas que construirão e consolidarão um bom nível de conhecimento para seus alunos, sendo que encontraríamos, de fato, a presença de cada uma delas neste projeto, mesmo que em outros contextos.

8.3 Como aprimorar os conhecimentos

Ao analisarmos o item anterior, fica evidente que este projeto nos dá abertura a desenvolver diversos conceitos que foram trabalhados superficialmente. Dado que é um projeto que envolve tanto questões técnicas de computação, como soluções para o mercado financeiro, ele proporciona a flexibilidade e a possibilidade de seguir dois caminhos, seja um mais técnico, ou uma mais voltado à parte de aplicação, no caso o próprio ramo das finanças.

Eventualmente, buscar novos conhecimentos sobre técnicas de agrupamento e detecção de *outliers* pode ser um ponto interessante a ser trabalhado pois, conforme citado anteriormente, são temas que podem ser contextualizados na grande maioria de setores que envolvem manipulação de dados.

Tenho muito interesse em Economia e acredito que uma especialização nesta área poderia me render bons frutos de conhecimento, unindo os conhecimentos de computação aos de mercado financeiro. Parece ser uma boa combinação a ser trabalhada.

Referências Bibliográficas

[1] Caio Ramos Casimiro. Desenvolvimento de uma api para estimação de betas variáveis de fundos brasileiros. Trabalho de conclusão de curso, EACH-USP, 2010.

[2] Landau Everitt. *Cluster Analysis*. Wiley, 2011.

[3] Zutter Gitman. *Principles of Managerial Finance*. Prentice Hall, 2011.

[4] A. Hadi. Identifying multiple outliers in multivariate data. *Journal Royal Statistics Society*, 1992.

[5] Roberta Anchieta da Silva. Estimação dinâmica do beta do modelo CAPM em fundos de ações. Dissertação de mestrado, IME/FEA-USP, 2007.