Gustavo Kendi Tsuji

Segmentação e classificação de Big Data

São Paulo

2016

Gustavo Kendi Tsuji

Segmentação e classificação de Big Data

Experimento quantitativo sobre algoritmos de segmentação e classificação para grandes bases de dados

Universidade de São Paulo - USP

Faculdade de Economia, Administração e Ciências Contábeis - FEAUSP

Bacharelado em Administração

Orientador: Alessandra Montini

São Paulo

2016

Agradecimentos

Gostaria de agradecer a professora Alessandra Montini pelo apoio durante os estudos sobre um tema que tenho muito interesse em continuar aprendendo. Também gostaria de agradecer ao meu amigo Paulo Haddad que deu dicas e explicações em momentos chaves do estudo.

Além disso, quero agradecer a minha família que sem dúvidas me deram o suporte para que pudesse estudar e em especial, a minha noiva Eliana que sempre esteve ao meu lado mesmo neste ano cheio de imprevistos.

"One's mind, once stretched by a new idea,
never regains its original dimensions.
(HOMES, Oliver Wendell)

Resumo

Nos dias atuais, a informação se tornou um recurso estratégico, crucial para toda empresa que busque competitividade no mercado. A sistematização e estudos de indicadores estão cada vez mais viavéis por conta da evolução tecnológica, que facilitou a acessibilidade de dessas informações. É a era do *Big Data*. Com os dados internos da empresa e externos do mercado é possível construir indicadores que podem dar norte a decisões . Em paralelo a isso, algoritmos de *Machine Learning* auxiliam a tarefa de compreender diversos aspectos explícitos e implícitos da empresa.

Palavras-chave: *Big Data*. K Médias. Regressão Logística. *Random Forest*

Abstract

In nowadays, the information has become a strategic resource, crucial to every company which seeks market competitiveness. The systematization and KPI studies are more viable because of technological evolution which has created facilities to retrieve those information. It is the Big Data era. With internal company and market external data it is possible to build KPI that can guide every decision. At the same time, machine learning algorithms aid to understand several explicit and implict company's aspects

Keywords: : Big Data. K Means. logistic Regression. Random Forest

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Na *internet*, é possível encontrar mais de 60 trilhões de páginas indexadas pelo Google (SMITH, 2016). Só o Facebook possui *data warehouses*[1] com mais de 300 *petabytes*, tendo um tráfego de mais de 600 *terabytes* diários (VAGATA; WILFONG, 2014). Como é possível notar, a tecnologia evoluiu a ponto de tornar o armazenamento de volume de dados um grande desafio. Mas, ao mesmo tempo, abriu portas para novas possibilidades.

O conteúdo dessas páginas ou operações executadas freneticamente todos os dias geram dados que, segundo Baeza-Yates e Ribeiro-Neto (1999), são objetos brutos que trazem pouco ou nenhum significado, armazenados de forma estruturada ou não. Por trás desses dados existe o que o autor chama de informação. A informação, então, seria uma interpretação do dado dentro de um contexto com um ganho cognitivo. A utilização dela produz o conhecimento e a aprendizagem, o que permite o desenvolvimento de novos produtos ou serviços como um melhor entendimento sobre comportamento dos clientes e do funcionamento de processos, otimização da produção, entre outras melhorias.

Por conta da dificuldade computacional em não só armazenar como também analisar e monitorar esse volume de dados que nasceu o *Big Data*. A interpretação e extração de informações possibilitam uma melhor compreensão de vários aspectos, tanto nas esferas micro e macro na empresa. A utilização de *Big Data* a favor da companhia pode conferir uma grande vantagem competitiva, bem como uma diferenciação, sendo considerado um ativo estratégico muito valioso.

Este trabalho visa, portanto, estudar conceitos teóricos estatísticos que analisam os dados, os algoritmos que criam as informações, bem como tecnologias que auxiliam o processamento e execução do algoritmo em larga escala.

1.1 Objetivo

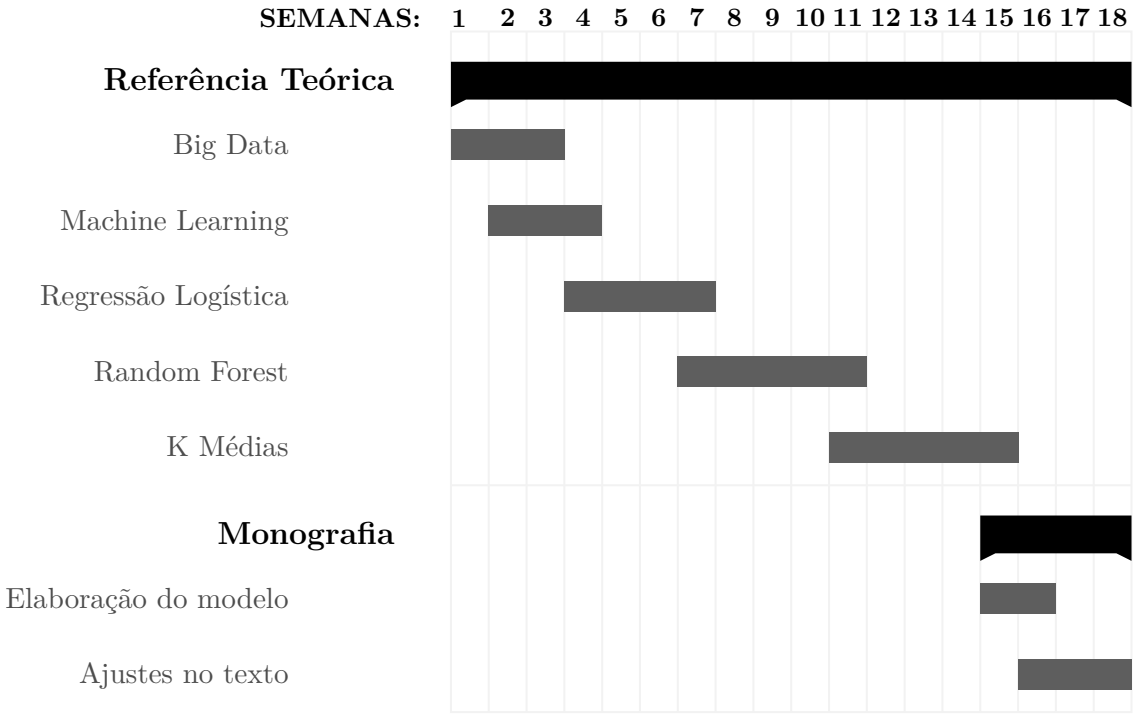
Este trabalho se baseia em um experimento quantitativo sobre problemas relacionados a segmentação e classificação de dados de uma base de dados grande, utilizando algoritmos de aprendizagem supervisionada e não supervisionada. Para resolver o problema de segmentação, este trabalho irá abordar o algoritmo de K Médias (não supervisionado) e para os casos de classificação, Regressão Logística e *Random Forest* (supervisionados). Também serão feitas interpretações, análises e comparações, levantando aspectos positivos e negativos de cada metodologia.

[1] *data warehouses* são banco de dados otimizados para consultas e análise de dados e geração de relatório em detrimento de operações transacionais

1.2 Cronograma

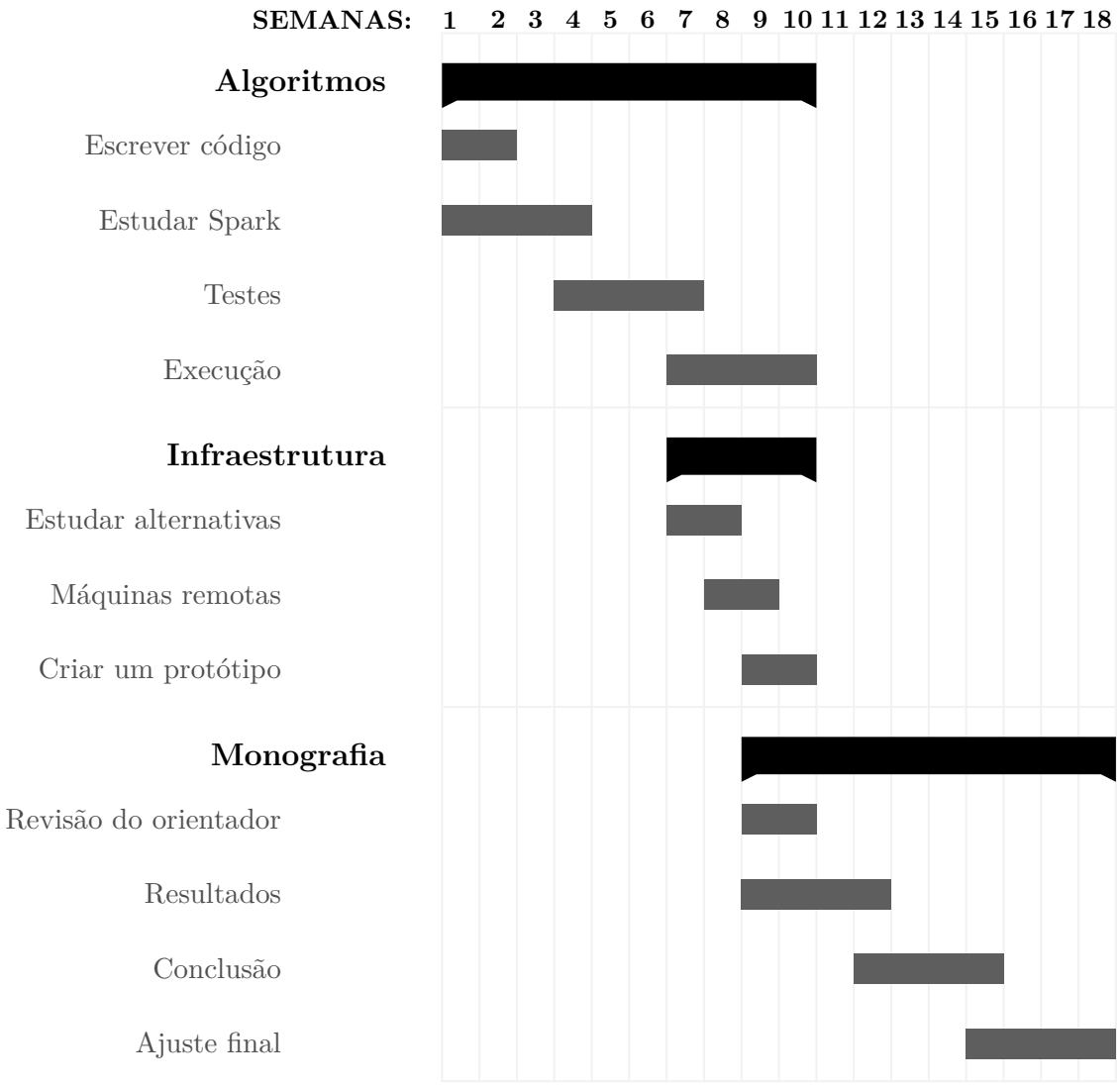
1.2.1 Primeiro semestre

Para este período, ficou decidido estudar o contexto atual de como as empresas estão lidando com os horizontes abertos pelo *Big Data* e seus desafios. Também foi definido que é necessário estudar sobre princípios básicos de *Machine Learning* para a resolução de problemas de clusterização e classificação, aprofundando sobre detalhes dos conceitos por trás dos algoritmos e de sua implementação.



1.2.2 Segundo semestre

Para este período, será feito uma revisão sobre o conteúdo apresentado na primeira versão da monografia, ajustes em relação a proposta do trabalho e aplicação dos algoritmos. A infraestrutura e tecnologia a serem utlizadas serão pensados e revistos durante o segundo semestre, visto que a execução sobre uma grande base de dados é complexa e custosa. Também iremos escolher uma base grande apropriada para a execução dos algoritmos.


SEMANAS: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Algoritmos
Escrever código
Estudar Spark
Testes
Execução
Infraestrutura
Estudar alternativas
Máquinas remotas
Criar um protótipo
Monografia
Revisão do orientador
Resultados
Conclusão
Ajuste final

2 *Referencial Teórico*

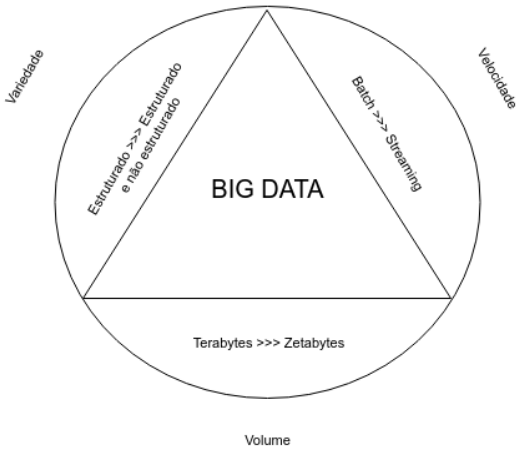
2.1 *Big Data*

Não existe uma definição clara do que é *Big Data*. Para Mayer-Schönberger e Cukier (2013, p. 6), trata-se de um volume de informações que cresceu de tal forma que simples computadores não são capazes de processar com ferramentas tradicionais. Pode referir-se a qualquer ação ou evento que necessite ser executado em larga escala, inviável de ser realizado numa estrutura menor, para extrair novos *insights* ou criar novas formas de valor de tal forma que provoque mudanças nos mercados, organizações, na relação entre as pessoas e os governos, entre outros.

Big data é a habilidade da sociedade em aproveitar informações de diversas maneiras para produzir novas ideias ou bens e serviços com um valor significativo(MAYER-SCHöNBERGER; CUKIER, 2013, p. 2). Para os autores, os dados não ficam mais num estado estático, ou seja, depois de sua coleta, existe uma utilidade além do seu simples armazenamento. Caso sejam analisados da forma correta, os dados podem ser reutilizados, tornando-se uma fonte de inovação e novos serviços.

Para (ZIKOPOULOS; EATON, 2011, p. 5), *Big Data* pode ser caracterizado por:

Figura 1 – Características de *Big Data*.



Fonte – Extraído e traduzido de (ZIKOPOULOS; EATON, 2011, p. 5)

- Volume: refere-se a grande volume de dados gerados. Dentre os desafios, em grande parte resolvidos, estão que tipo de tecnologia a ser utilizada para guardar um volume grande de dados, uma vez que as tradicionais são incapazes ou ineficientes para lidar com essa questão, além do custos de armazenagem.

- Velocidade: relacionado a rapidez com que os dados são gerados. Com a evolução da tecnologia, tudo está cada vez mais interconectado. As bandas largas possibilitam um tráfego de dados cada vez maior e mais eficiente. Sistemas de tempo real passaram a ter maior relevância para as empresas que podem tomar decisões cada vez mais rápida.

- Variedade: trata-se dos diferentes dispositivos que podem gerar dados passíveis de extração de informação. *Smartphones*, *tables*, *internet* das coisas(IOT), computadores, sensores podem produzir dados em diferentes formatos que precisam ser interpretados e armazenados.

2.2 *Algoritmos*

2.2.1 Segmentação

Com um volume de dados grande disponível, uma possibilidade para as empresas é conseguir reconhecer certos padrões. Por exemplo, conhecendo o perfil dos clientes, é possível adotar estratégias adequadas para cada segmento, principalmente quando o público alvo é composto por clientes muito heterogêneos. Segundo Kotler (1992, p. 257), "segmentação de mercado é o ato de dividir um mercado em grupos distintos de compradores com diferentes necessidades e respostas".

> Um bom agrupamento exibe a característica de que objetos associados ao mesmo grupo são bastante similares, ao mesmo tempo em que objetos associados a grupos diferentes exibem uma baixa similaridade. Aplicações diretas da análise de grupos incluem segmentação de clientes ou de produtos, agrupamento de genes em um experimento de micro-array, organização dos resultados de uma consulta enviada a um mecanismo de busca da WEB, etc. (BEZERRA, 2006)

Com a segmentação e os grupos definidos, é possível realizar uma análise descritiva para traçar um padrão no comportamento dos dados.

2.2.1.1 K Médias

O K Médias é um algoritmo de *machine learning* não supervisionado relativamente simples, podendo ser utilizado para resolver problemas de clusterização. Para MacQueen (1967), trata-se de um método que tem para uma quantidade k de *clusters* pré definida, o objetivo de definir k centróides[1], um para cada *cluster*, tal que o conjunto de dados possa

[1] centróide é um conceito muito utilizado em geometria e física e representa um ponto médio ou um centro de massa de uma representação. No caso de K Médias, considerando que as informações são transformadas em vetores, seria um ponto médio da informação

ser repartido de forma eficiente. Para um conjunto de observações $(x_1, x_2, ..., x_n)$, onde

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 , \tag{2.1}$$

onde μ_i é a média dos pontos em S_i

O algoritmo minimiza a função objetiva usando o princípio dos mínimos quadrados. Por conta disso, é sensível a *ouliers* e ruídos. O pseudo algoritmo do K Médias seria estes passos:

1. Defina uma inicialização inicial aleatória usando k *clusters*;
while *Não houve convergência* **do**
    2. Atribua para cada ponto do conjunto de dados um *cluster* mais próximo;
    3. Redefina a posição do centróide de cada *cluster* como um ponto médio de todos os pontos do *cluster*;
end

Algoritmo 1: K Médias

A localização desses centróides deve ser o mais afastado entre si possível. A partir de uma posição inicial dos centróides, o próximo passo é, então, associar todos pontos do conjunto de dados com o centróide mais próximo. Com os pontos associados, recalcula-se k novos centróides como baricentros dos *clusters* anteriores, repetindo esses passos até que os novos centróides sejam gerados muito próximos do passo anterior.

2.2.2 Classificação

A classificação é uma análise preditiva com o objetivo de estabelecer modelos que possbilitem uma previsão do futuro, permitindo estudar tendências por meio do uso de técnicas estatísticas sobre dados históricos.

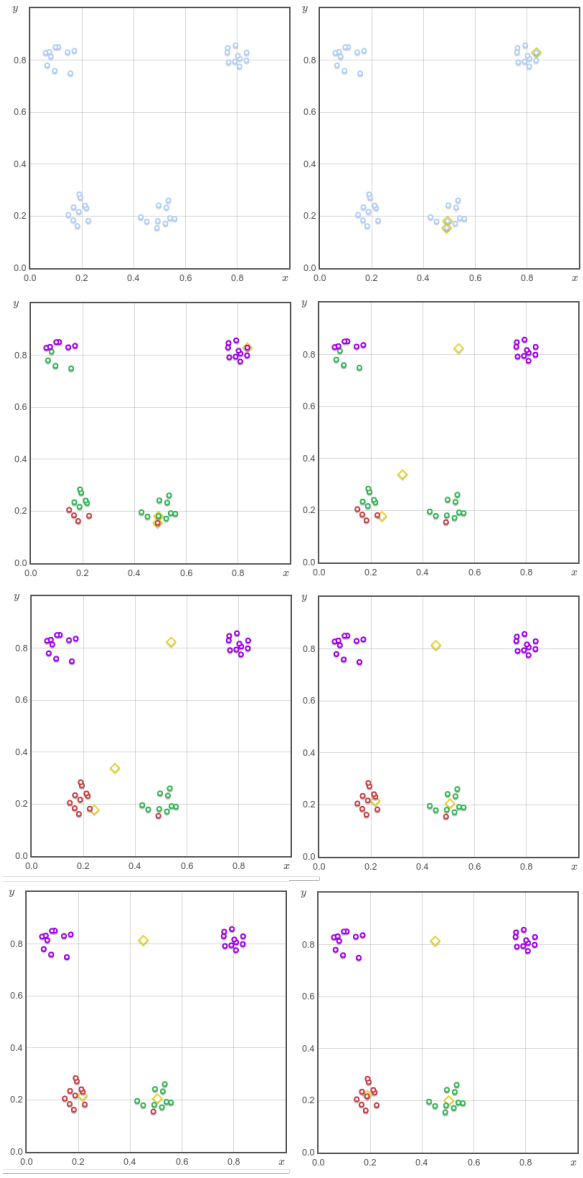
2.2.2.1 Regressão Linear

A regressão linear é uma modelagem matemática [2] que permite descrever variáveis em função de outras. Segundo Hastie, Tibshirani e Friedman (2011, p. 44) ela pode ser representada como

$$Y = \hat{\beta}_0 + \sum_{j=1}^{p} (X_j \hat{\beta}_j), \tag{2.2}$$

[2] modelagem matemática é uma representação em fórmulas matemáticas que tentam descrever ou simular eventos e sistemas reais com o propósito de prever comportamentos

Figura 2 – Evolução da execução do algoritmo de K Médias



Fonte – Extraído de (CONCEIÇÃO,)

onde Y representa uma variável dependente contínua, X_j as variáveis independentes (contínuas, discretas ou binárias). Isto significa que uma certa característica (variável dependente) pode ser descrita por outras (variável independente).

Para ajustar o modelo linear ao conjunto de dados, é possível utilizar diferentes maneiras. Um deles é método dos mínimos quadrados, uma técnica de otimização matemática que visa encontrar ajuste ótimo para um conjunto de dados por meio da minimização a soma dos quadrados das diferenças entre o valor estimado e os dados

observados, representado por:

$$G(\beta) = \sum_{i=1}^{N}(y_i - x_i^T\beta)^2, \tag{2.3}$$

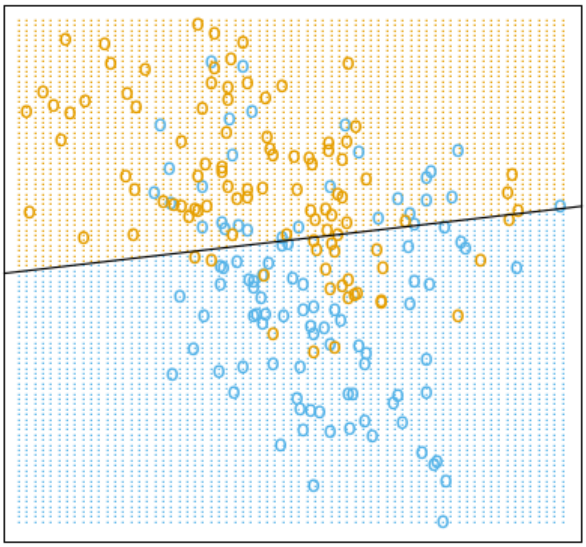
Assim, nosso problema passar a ser como descobrir $\hat{\beta}$ que minimize 2.3. Para calcular $\hat{\beta}$, é possível utilizar a equação:

$$\hat{\beta} = (X^TX)^{-1}X^Ty, \tag{2.4}$$

onde $X \in \mathbb{R}^{N,p}$ X representando uma matriz com cada linha sendo um vetor do conjunto de dados de entrada e $y \in \mathbb{R}^N$ um vetor que representa os dados de saída [3]

Com a definição de $\hat{\beta}$, é possível determinar uma reta que separa o conjunto de dados.

Figura 3 – Exemplo de classificação em 2 dimensões.



Fonte – Extraído de (HASTIE; TIBSHIRANI; FRIEDMAN, 2011)

As classes estão representadas um variável binária (AZUL = 0, LARANJA = 1), ajustadas por uma regressão linear. A linha que separa os grupos foi definido por $x^T\hat{\beta} = 0,5$. A área hachurada em laranja representa o espaço classificado por LARANJA enquanto a área em azul, classificado por AZUL

2.2.2.2 Regressão Logística

Segundo Hastie, Tibshirani e Friedman (2011, p. 119), a regressão logística, assim como a regressão linear, também é um modelo matemático de predição de eventos usada

[3] inicialmente, esses dados devem vir do conjunto de treino

para descrever dados e explicar a relação entre um conjunto de variáveis independentes e uma variável dependente. Contudo, enquanto na regressão linear a variável dependente é contínua, na regressão logística, ela é considerada uma variável categórica. Ela segue uma distribuição Bernoulli[4] com uma probabilidade p desconhecida. Assim, a regressão logística tem como objetivo estimar essa probabilidade p desconhecida.

Para estimar essa probabilidade, a regressão logística usa as chances (*odds*) do evento ocorrer em cada variável independente, calculando a taxa dessas chances, dada pela equação:

$$OR = \frac{P(sucesso)}{P(fracasso)} = \frac{p}{1-p}. \tag{2.5}$$

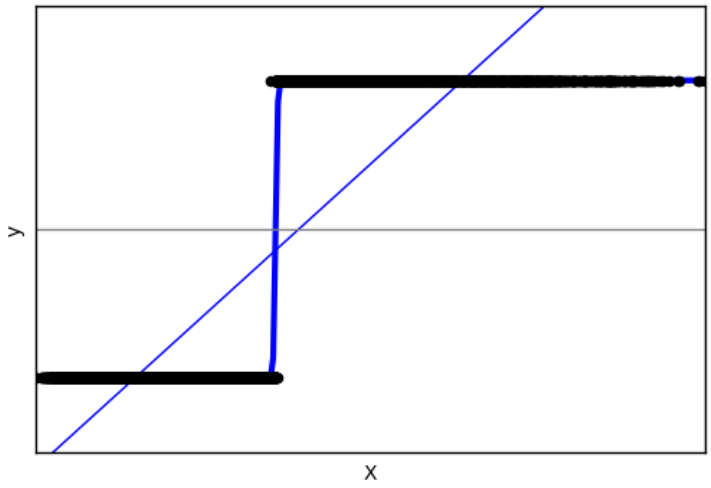
Utilizando inferência de estatística, podemos aplicar log em 2.5, ficando com:

$$logit(p) = ln\left(\frac{p}{1-p}\right). \tag{2.6}$$

Tal transformação recebe o nome de logit. Ela é ajustada a função de predição, como numa análise de regressão linear, visto anteriormente. O valor final obtido a partir da função logit é convertido novamente para as chances via a função inversa do logaritmo natural (ou uma função exponencial).

$$logit^{-1}(\alpha) = \frac{1}{1+e^{-\alpha}} = \frac{e^{\alpha}}{1+e^{\alpha}}. \tag{2.7}$$

Figura 4 – Função logit



Fonte – Gerado a partir do script LogisticRegression.ipynb

[4] distribuição de Bernoulli é uma modelagem de probabilidade que representa eventos binários cujas ocorrências são tratados como sucesso ou falha. Considerando que a probabilidade de ocorrer um sucesso é p, então a probabilidade de ocorrer uma falha é $1-p$

Generalizando com 2.8, 2.9 e 2.10 , temos:

$$\log\left(\frac{P(G=1|X=x)}{P(G=K|X=x)}\right) = \beta_{10} + \beta_1^T x \tag{2.8}$$

$$\log\left(\frac{P(G=2|X=x)}{P(G=K|X=x)}\right) = \beta_{20} + \beta_2^T x \tag{2.9}$$

$$\log\left(\frac{P(G=K-1|X=x)}{P(G=K|X=x)}\right) = \beta_{(k-1)0} + \beta_{k-1}^T x, \tag{2.10}$$

onde o modelo é composto por K classes e K - 1 transformações logit. Utilizando a inversa da logit, temos:

$$\begin{aligned} P(G=k|X=x) &= \frac{\exp\left(\beta_{k0} + \beta_k^T x\right)}{1 + \sum_{\ell=1}^{K-1} \exp\left(\beta_{\ell 0} + \beta_\ell^T x\right)}, k = 1, ..., K-1 \\ P(G=K|X=x) &= \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp\left(\beta_{\ell 0} + \beta_\ell^T x\right)} \end{aligned} \tag{2.11}$$

Dessa forma, temos com 2.11 que a regressão logística estima as changes (odds) como uma variável contínua, mesmo quando a variável dependente que está sendo o objeto de estudo seja uma variável binária.

Assim, a regressão logística viabiliza a classificação das observações por meio da probabilidade estimada na categoria estudada.

2.2.3 *Random Forest*

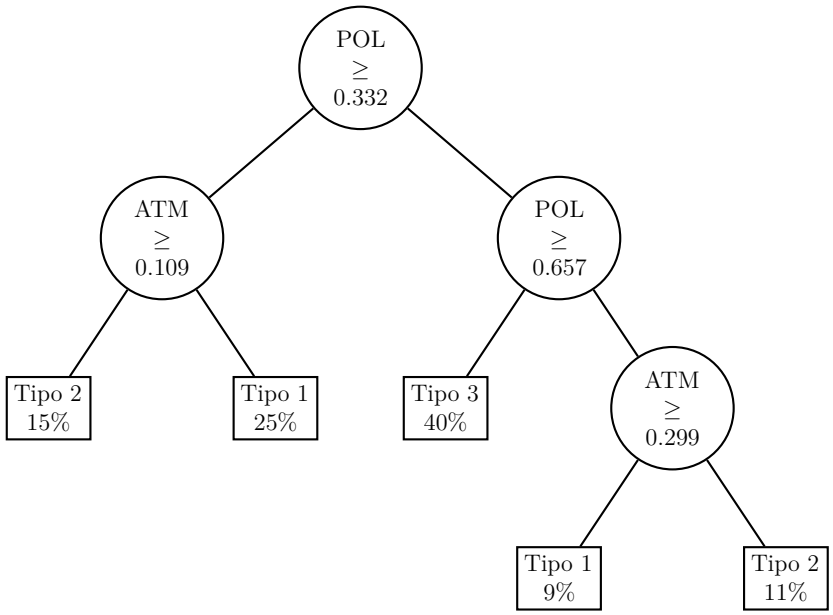
2.2.3.1 Árvores

Árvores são modelos presentes tanto em computação como estrutura de dados e em estatística como estrutura para tomadas de decisão. No contexto de *Machine Learning*, a árvore de decisão refere-se a uma estrutura de modelo preditivo, um método de aprendizagem supervisionada não parametrizada utilizada para classificação (para variáveis categóricas) e regressão (variáveis métricas). Trata-se de um modelo de conjunto de decisões ou regras na qual estabelece um fluxo dentro de sua estrutura, definindo uma classificação ou uma predição. Para Hastie, Tibshirani e Friedman (2011, p. 305), as árvores permitem um particionamento do espaço em um conjunto regiões. Suponha que exista M partição que possa ser divida em regiões $R_1, R_2, ..., R_M$ e que seja possível modelar a resposta para cada região com a constante c_m,

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{2.12}$$

Para a repartição dessas regiões, são definidos critérios para decidir de qual lado o dado irá ficar na árvore. São estruturas de fácil interpretação pois, uma vez criado o modelo, percorrer os nós da árvore indica as características do dado.

Figura 5 – Exemplo de árvore classificadora



Nota – Cada nó representa um atributo de um elemento da amostra. As folhas são consideradas a representação da classe a que uma observação pertence. Já o ramo é um conjunto de valores que reflete todas suas características e detalhes de um elemento

Fonte – Criado pelo autor

2.2.3.2 Árvore de regressão

Para Hastie, Tibshirani e Friedman (2011, p. 307), árvores de regressão são utilizadas em problemas de predição. Assim, a variável de saída refere-se a uma variável numérica e contínua.

Assim como foi visto anteriormente, o algoritmo deve conseseguir decidir em quais variáveis e quais pontos as decisões serão tomadas, construindo a topologia da árvore.

No caso de uma regressão, poderia utiliza-se como critério de minimização a soma do mínimos quadrados $\sum (y_i - f(x_i))^2$, temos que o melhor $\hat{c}_m$ é exatamente a média para y_i na região R_m:

$$\hat{c}_m = média(y_i | x_i \in R_m) \tag{2.13}$$

Para encontrar a melhor partição, é necessário recorrer a um algoritmo guloso[5],

[5] *algoritmo guloso* é um algoritmo que busca a resolução de um problema elegendo sempre uma solução localmente ótima. Isto significa que, dentro de um conjunto de soluções possíveis numa determinada etapa da solução do problema, escolhe-se sempre a que traz o melhor resultado naquela situação, o que muitas vezes pode não levar a uma solução ótima global.

uma vez que calcular utilizando os mínimos quadrados é computacionalmente inviável.

Seja j uma variável de reparticionamento, s um ponto de divisão. É possível definir um par de semi planos

$$R_1(j,s) = X|X_j \leq s \quad \text{e} \quad R_2(j,s) = X|X_j > s). \tag{2.14}$$

Resultando a busca pela de s e j que resolva

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2 + \min_{c_1}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \tag{2.15}$$

Mas como visto em 2.13, temos que para qualquer j e s, a minimização interna pode ser resolvida por:

$$\hat{c}_1 = \textit{média}(y_i|x_i \in R_1(j,s)) \quad \text{e} \quad \hat{c}_2 = \textit{média}(y_i|x_i \in R_2(j,s)). \tag{2.16}$$

Este processo é repetido até que todas as regiões sejam descobertas.

2.2.3.2.1 Árvore de classificação

Para o caso de uma árvore de classificação, Hastie, Tibshirani e Friedman (2011, p. 308) explica que o objetivo principal concentra-se em conseguir, a partir das variáveis independentes, efetuar a decisão em definir um classe para uma determinada entrada de dados. Logo, a saída é uma variável categórica. Neste caso, utiliza-se como critério de decisão o índice de Gini[6] $Gini(T) = 1 - \sum_{i=1}^{n} p_i^2$, onde p é a proporção de observações de uma determinada classe para um dado nó.

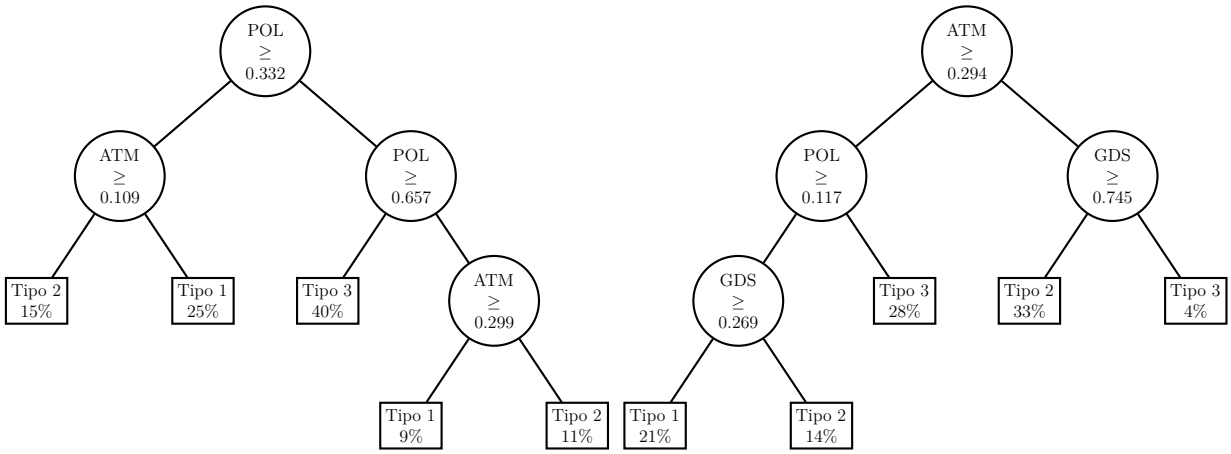
2.2.3.3 *Bagging*

As árvores vistas anteriormente são modelos interessantes de classificação mas possuem um problema: o resultado gerado por elas em geral possuem uma acurácia muito baixa quando a estrutura da árvore começa a crescer muito (HASTIE; TIBSHIRANI; FRIEDMAN, 2011, p. 312). A proposta do *Bagging* é criar subconjuntos dos dados, gerando diversas árvores que deverão executar a classificação. Esses modelos subconjuntos são gerados com reposição, ou seja, um mesmo dado pode estar presente em mais de um modelo no momento do treino. Cada árvore gerada fornecerá um modelo para a *Random Forest* e um valor intermediário será adotado para os nós.

Dessa forma, a solução para contornar esse problema foi a utilização dessa técnica nas árvores de decisões. Na *Random Forest*, constrói-se uma árvore de classificação

[6] Índice de Gini mede a frequencia de que um elemento selecionado aleatoriamente de um conjunto é marcado de forma errada

Figura 6 – Exemplo de composição da *Random Forest*



(a) Configuração 1 da árvore de decisão **(b)** Configuração 2 da árvore de decisão

Fonte – Criado pelo autor

repetidamente usando amostras aleatórias do conjunto de dados. Por fim, o modelo final é recalculado.

2.3 Tecnologias

2.3.1 *Scikit Learn*

O *Scikit Learn* é um *framework* desenvolvido em *Python* voltado para *Data Science*. Com mais de 30 contribuidores ativos, possui implementação de diversos algoritmos e de ferramentas que auxiliam desde a análise dos dados até a execução de tarefas como clusterização e classificação, além de disponibilizar integrações com programas visuais que facilitam muito a compreensão.

2.3.2 *Apache Spark*

O *Apache Spark* é uma biblioteca com diversas ferramentas e viabiliza a escalabilidade da execução de algoritmos de *Machine Learning*. É possível utilizar mais de linguagem de programação para desenvolver os programas. Neste trabalho, foi escolhido Scala. Os algoritmos estudados (em especial o K Médias) são robustos e possuem uma grande carga de cálculos. Para bases grandes, o *Scikit Learn* não é capaz de executar em tempo satisfatório. O *Apache Spark*, por outro lado, oferece uma solução que possibilita uma maior rapidez na execução. Também é possível configurá-lo para que os algoritimos sejam executados em tempo real com alimentação contínua de dados. Contudo tal abordagem não será contemplada por não se tratar do foco do estudo.

2.4 Avaliação dos algoritmos

2.4.1 Segmentação

2.4.1.1 Análise Descritiva

A análise descritiva possibilita a descrição e sumarização dos dados de um conjunto. Há 2 tipos de medidas: a de tendência central e a de variabilidade ou dispersão. Entre as medidas de tendência central estão média, mediana e moda. Já para as medidas de variabilidade, existem o desvio padrão, variância, o valor máximo e mínimo e quantis. Para a análise descritiva, pode-se utilizar gráficos como histogramas e boxplots que ajudam visualizar com clareza e caracterizar dos dados.

2.4.2 Classificação

2.4.2.1 *Cross Validation*

A *Cross Validation* é uma técnica que permite verificar a eficiência de um modelo preditivo (HASTIE; TIBSHIRANI; FRIEDMAN, 2011, p. 241). A idéia é particionar a base para se criar um modelo de treinamento e executar um teste. Essa partição é mutualmente exclusiva, ou seja, os dados que são usados para o treinamento não são reutilizados no teste. Neste caso, a *Cross Validation* será executada nos algoritmos de classificação.

2.4.2.2 Métricas

O cálculo de métricas para os algoritmos são importantes para avaliar o desempenho da classificação. Cada métrica pode ser mais conveniente para cada tipo de problema ou natureza dos dados. Neste trabalho iremos abordar algumas métricas simples mas iremos fazer uma análise profunda.

2.4.2.2.1 *Precision*

É calculado a partir da proporção dos verdadeiros positivos sobre a quantidade de todos elementos que foram classificados na mesma classe (soma dos verdadeiros positivos e falsos positivos)

$$Precision = \frac{VP}{VP + FP} \tag{2.17}$$

2.4.2.2.2 *Recall*

É a proporção de elementos foram classificados corretamente em uma classe sobre a quantidade de todos elementos que realmente pertencem a esta classe mesmo que estejam

em outras (soma dos verdadeiros positivos e falsos negativos).

$$Recall = \frac{VP}{VP + FN} \tag{2.18}$$

2.4.2.2.3 Falso Positivo

O Falso Positivo corresponde aos elementos que foram classificados como pertencentes a uma dada classe mas que, quando na verdade, não pertencem a essa classe.

2.4.2.2.4 *F-measure*

O F-measure é a média harmônica entre *precision* e *recall*. Trata-se de uma medida que consegue avaliar melhor conjunto de dados que possuem uma distribuição das classes desproporcionais.

$$F - measure = (1 + \beta^2)\frac{precision * recall}{\beta^2 * precision + recall} \tag{2.19}$$

2.4.2.3 *Confusion Matrix*

Trata-se de uma matriz quadrada na qual as colunas representam as classes as quais o elementos realmente pertencem enquanto as linhas representa a saída do classificador. Cada célula fornece a quantidade de elementos classificados.

3 Resultados

3.1 *Loan Club*

A *Loan Club* é um *marketplace* de créditos *online*. Trata-se de uma plataforma que reune pessoas que gostariam de realizar um empréstimo e outras que possuem um capital a investir. Em geral, são empréstimos pessoais, de negócios ou para procedimentos médicos. A proposta é oferecer um serviço de fácil acesso via dispositivos mobiles a taxa baixas, além de garantir um retorno que se ajuste a expectativa do investidor. Dessa forma, operam de forma menos burocrática do que um banco mas assumindo os riscos de uma instituição financeira.

3.2 *Kaggle*

Kaggle é uma plataforma criada em 2010 que reune ferramentas e estudos sobre modelagem preditiva. Também viabiliza competições de temas relacionados a análise preditiva. Com o intuito de compartilhar conhecimento, qualquer interessado pode acessar dados de empresas para aplicar estudos ou gerar novos modelos. Além disso, o *Kaggle* também tem sido usado como uma forma de recrutar cientistas de dados. A base da *Loan Club* foi disponibilizado no Kaggle e usado como objeto de estudo deste trabalho.

3.3 Preparação da base

Como uma base de uma empresa, a *Loan Club* apresenta diversos problemas como:

3.3.1 *Missing values*

Inerente a uma base comum de qualquer empresa, *missing values* ocorrem quando uma informação não consta na base. Assim como exemplifica Hastie, Tibshirani e Friedman (2011, p. 311) esse fato pode influenciar a conclusão de uma análise porque a ausência de dados pode distorcê-la. Para diminuir o impacto desse tipo de problema, é necessário verificar a natureza da informação. Esses erros são oriundos desde problemas do armazenamento da informação, provenientes de erros humanos, falta de informação ou em decorrência de alguma falha ou inconsistência no sistema de informação.

Para os algoritmos, é necessário que algum valor seja colocado para a realização dos cálculos. Há diversas abordagens, como substituição por um valor padrão que faça sentido para cada informação. Em alguns casos, são usados valores como 0, média ou

moda dos campos respectivos das observações. Contudo, é evidente que cada uma dessas técnicas enviesa e modifica o resultado. Neste estudo, desprezamos as informações que possuiam altas ocorrências de missing values e para as que possuiam menos de 1% de missing values, foi preenchido com 0.

3.3.2 Conversão de tipo de dados

Alguns dados como int_rate ou o term estavam armazenados como tipo texto, sendo inviável a execução do algoritmo do K Médias. Para isso, foram removidos termos de texto e converteu-se os valores em número.

3.3.3 Variáveis categóricas

O algoritmo de K Médias, a princípio, não é executado com uma base que contenha variáveis categóricas. É possível adaptar a base executando uma transformação, tornando-as variáveis *dummies*. Contudo, não iremos considerar as variáveis categóricas por haver uma carga considerável e inviável de trabalho no escopo deste estudo. A *Random Forest* consegue lidar com variáveis categóricas mas também necessita de algumas transformações. Contudo, elas não foram feitas neste trabalho.

3.4 Algoritmos

Todos os scripts dos algoritmos foram desenvolvidos baseado nos códigos que estão no site do Scikit Learn e do Apache Spark. Encontram-se noo GitHub (TSUJI,).

3.4.1 K Médias

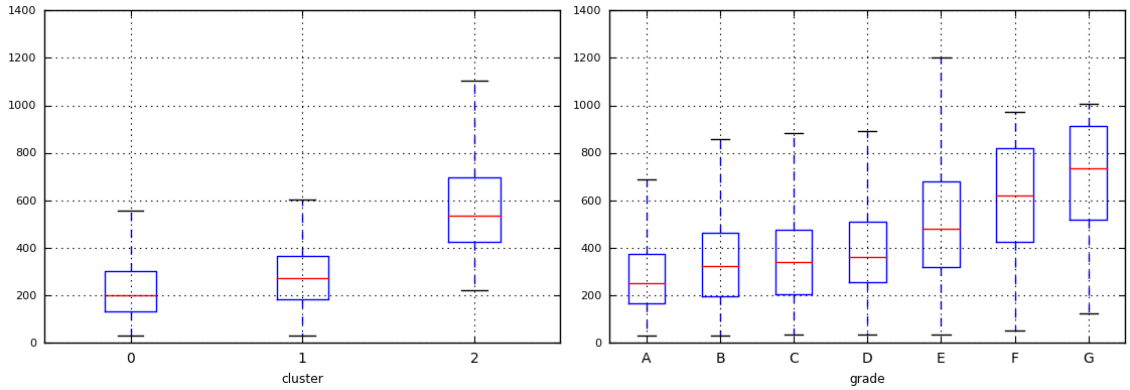
O objetivo de uma clusterização é para dado um conjunto de dados e agrupar os elementos em grupos baseados na similaridade entre eles. Com a base de dados da *Loan Club*, já contamos com uma divisão feita por ela mesma. Dessa forma, uma hipótese a ser testada é constatar alguma relação nos grupos gerados pelo K Médias com alguma característica. Neste trabalho, iremos usar as notas atribuídas por ela mesma, armazenada no campo *grade*. Para a execução do K Médias, executou-se uma normalização dos dados. Ela é recomendada pois como os dados possuem diversas informações, é possível que haja grandes disparidades de intervalos. Assim quando se normaliza todos os dados, a idéia é uniformizar esses intervalos para que não haja muita interferência nos cálculos do algoritmo.

Pelo *Scikit Learn*, é possível criar gráficos de *boxplot* para visualizar em quais campos houve uma separação mais clara do banco. Inicialmente foi feito criado uma amostra de 8000 registros na qual foi executado o K Médias. Esses gráficos estão disponível em Anexos

na seção Análise dos clusters vs grades. Junto desses gráficos, há um comparativo com *boxplots* gerados a partir dos grades estabelecidos pela *Loan Club*. Dessa forma, pode-se ver que existe uma relação entre os *clusters* e os *grades*.

Campos como o funded_amnt, installment, loan_amnt, total_pymnt deixam muito claro a segregação dos *clusters*, enquanto que os outros não foram capazes de mostrar a diferença entre os grupos. Não foi possível criar esses gráficos para a base toda pois seria necessário uma integração entre as ferramentas do *Scikit Learn* e do *Apache Spark* uma vez que não foi possível clusterizar toda a base via *Scikit Learn*. Contudo, devemos ressaltar que dentro de um contexto de *Big Data* nem sempre será possível utilizar as ferramentas de visualização por conta da quantidade de dados.

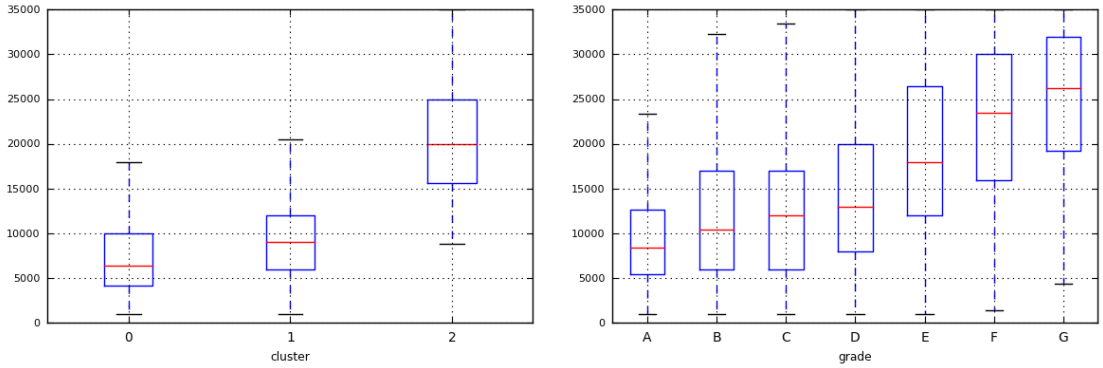
Figura 7 – *Boxplots* de installment



Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 8 – *Boxplots* de loan_amnt

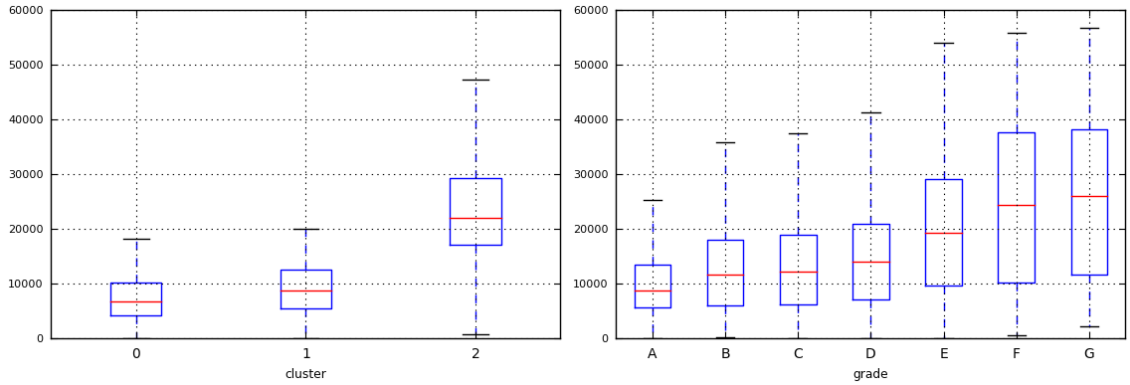
(a) Cluster **(b)** Grade



Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Pelo *Apache Spark*, é possível calcular algumas váriaveis como a média, o desvio padrão, mínimo e máximo sobre a base completa. Na tabela abaixo, comparamos os 4 componentes de cada *cluster*.

Figura 9 – *Boxplots* de total_pymnt



Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Tabela 1 – Média, Desvio Padrão, Mínimo, Máximo dos *clusters*

Variável	Média			Desvio Padrão			Mínimo			Máximo		
	1	2	3	1	2	3	1	2	3	1	2	3
1	1.103	1.038	-0.598	0.730	0.741	0.513	-0.562	-1.639	-1.689	2.403	2.403	2.403
2	1.099	1.039	-0.598	0.733	0.742	0.514	-1.729	-1.742	-1.742	2.404	2.404	1.812
3	1.106	1.037	-0.598	0.729	0.741	0.514	-0.563	-1.639	-1.690	2.399	2.399	2.399
4	0.331	0.659	-0.311	1.085	1.067	0.794	-0.654	-0.654	-0.654	1.527	1.527	1.527
5	0.469	0.259	-0.185	1.063	1.074	0.903	-1.809	-1.809	-1.809	3.592	3.592	3.592
6	1.164	0.892	-0.554	0.885	0.852	0.532	-0.748	-1.665	-1.724	3.985	4.131	2.880
7	0.338	0.372	-0.204	0.956	1.546	0.609	-0.788	-1.159	-1.141	76.122	145.675	91.578
8	-0.071	0.115	-0.030	0.430	1.833	0.497	-1.056	-1.056	-1.056	1.268	580.557	62.492
9	-0.036	0.034	-0.006	0.905	1.024	1.006	-0.364	-0.364	-0.364	27.471	29.791	44.869
10	0.121	0.053	-0.042	1.081	1.057	0.958	-0.696	-0.696	-0.696	10.331	32.385	26.370
11	0.003	0.461	-0.175	0.882	1.139	0.903	-2.172	-2.172	-2.172	7.983	14.754	9.488
12	-0.159	0.060	0.006	0.698	1.341	0.886	-0.335	-0.335	-0.335	35.739	147.398	16.842
13	0.295	0.429	-0.217	1.133	1.528	0.553	-0.754	-0.754	-0.754	52.312	128.779	21.998
14	0.088	0.533	-0.219	0.896	1.083	0.901	-1.965	-1.880	-2.049	7.663	12.139	7.071
15	-0.483	1.243	-0.383	0.842	0.928	0.573	-0.989	-0.989	-0.989	2.713	4.823	1.763
16	-0.483	1.243	-0.383	0.842	0.928	0.573	-0.989	-0.989	-0.989	2.714	4.825	1.764
17	2.129	-0.218	-0.310	0.938	0.573	0.569	0.121	-0.960	-0.960	6.379	4.054	2.658
18	2.131	-0.215	-0.311	0.941	0.573	0.567	-0.943	-0.958	-0.958	6.404	3.855	1.837
19	2.052	-0.332	-0.253	1.114	0.494	0.597	-0.869	-0.869	-0.869	4.413	4.413	2.904
20	1.507	0.216	-0.360	1.657	0.894	0.430	-0.837	-0.837	-0.837	10.713	6.837	2.927
21	0.112	-0.005	-0.018	1.757	1.076	0.742	-0.097	-0.097	-0.097	87.747	72.073	59.306
22	0.015	0.071	-0.029	1.474	1.360	0.676	-0.112	-0.112	-0.112	81.709	67.624	53.445
23	0.027	0.046	-0.022	1.488	1.286	0.724	-0.077	-0.077	-0.077	110.893	82.221	91.441
24	1.469	-0.281	-0.164	2.008	0.326	0.567	-0.451	-0.451	-0.451	7.155	7.006	5.194

Fonte – Gerado a partir do script spark.scala

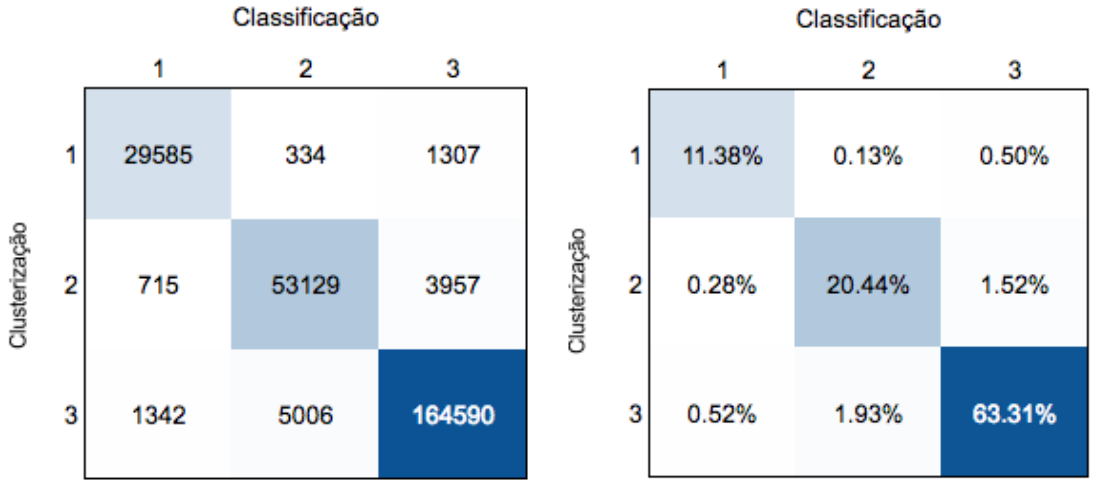
Assim como nos *boxplots* do Anexo, é possível notar pela tabela que algumas variáveis não possuem valores que viabilize a criação de *clusters* por si só. Podemos observar uma nítida diferença entre os *clusters* 1 e 3, sendo que o *cluster* 2 possui algumas variáveis mais próximas do *cluster 1* e para outras, mais próxima do *cluster* 3.

3.4.2 Regressão Logística

No *Apache Spark*, a execução da Regressão Logística foi dividida em 2 fases: treinamento do modelo e teste. A base completa foi dividida em 80% para treino e 20 % para testes. Para verificar a clssificação, foi considerado a clusterização realizada

previamente pelo K Médias.

Figura 10 – Resultados da clusterização da Regressão Logística



(a) Classificação em valores absolutos **(b)** Classificação em percentual

Fonte – Dados gerados a partir do script spark.scala

Podemos notar que houve uma classificação com baixos índices de erros, isto é, com menos de 5% de predições em classes não esperadas. A classificação dos registros seguiu uma proporção de 11,38% para a classe 1, 20,44% para a classe 2 e 63,31% para a classe 3.

Tabela 2 – Métricas para a Regressão Logística

Classe	Precisão	Recall	Falso Positivo	F-measure
1	0,93499	0,94744	0,00877	0,94117
2	0,91679	0,92643	0,02641	0,95234
3	0,96900	0,96286	0,05556	0,95241

Fonte – Gerado a partir do script spark.scala

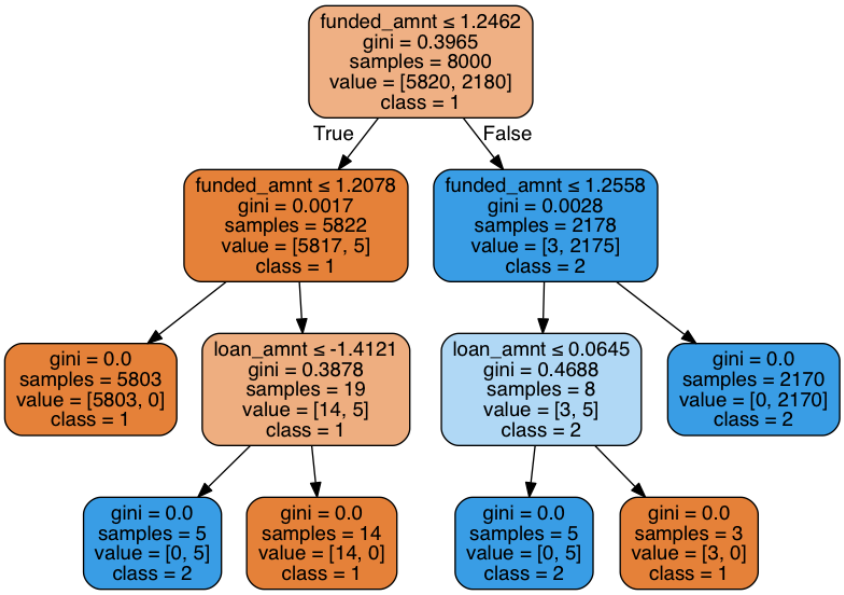
3.4.3 *Random Forest*

Com uma árvore de decisão tem-se uma representação visual, de fácil interpretação de como é feito a classificação. No *Scikit Learn*, foi possível construir uma árvore para ilustrar a árvore do *Loan Club*. Neste caso, foi feito uma redução de dimensionalidade para 2 *features*, considerando-se 3 *clusters*.

A árvore para mais de 20 *features* fica muito maior por conta da quantidade de informações.

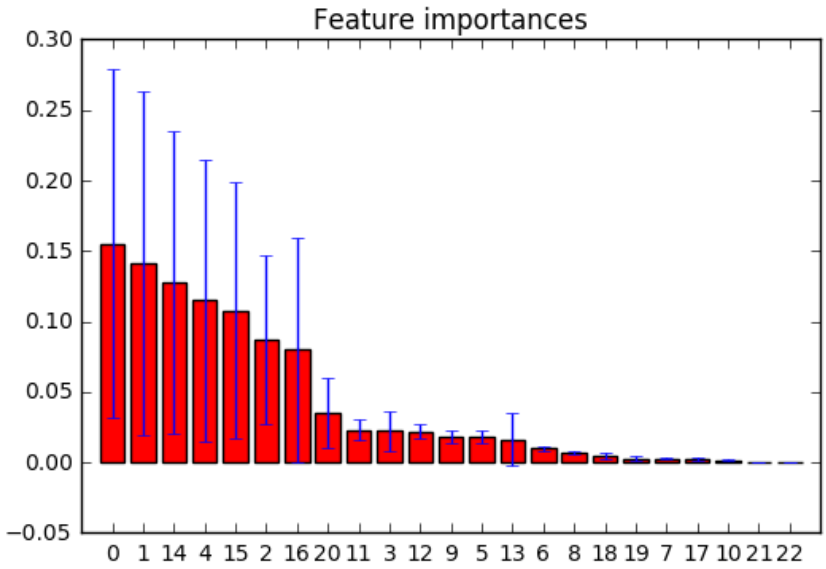
Ao gerar uma árvore de classificação, o algoritmo também disponibiliza a relevância das *features*.

Figura 11 – Estrutura da árvore de decisão reduzida a 2 *features*



Fonte – Gerado a partir do script RandomForest.ipynb

Figura 12 – *Features* mais relevantes



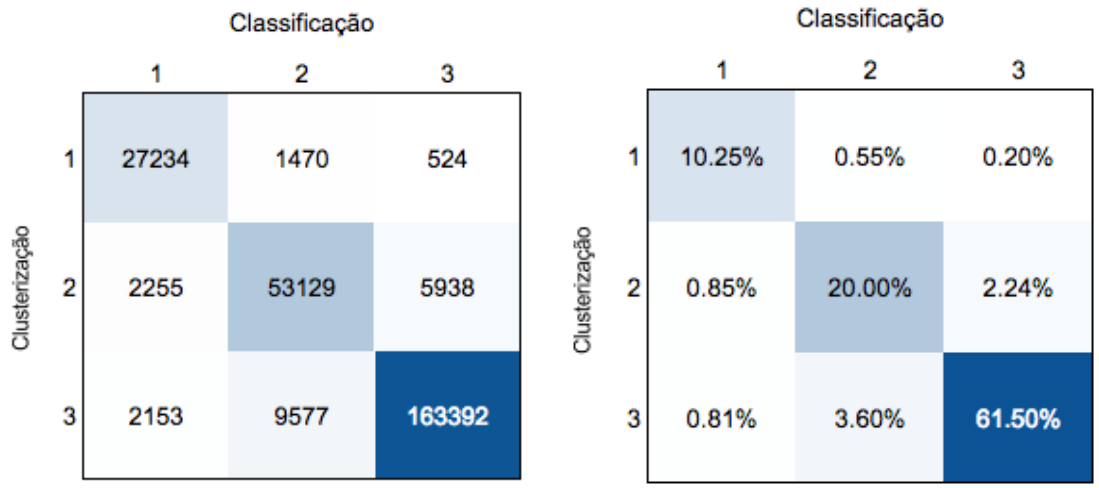
Fonte – Gerado a partir do script RandomForest.ipynb

Desta base, é possível notar que 8 destas *features* possuem maior relevância frente as demais. Isso pode levar a uma redução de dimensionalidade, isto é, as features de menor relevância passam a ser descartadas no treinamento do modelo, o que pode gerar economia de processamento de dados. Por outro lado, isso é válido partindo-se do pressuposto de que a natureza das informações se manterão, o que pode não ser verdade.

No *Apache Spark*, assim como na Regressão Logística, a classificação com *Random Forest* também foi dividida em 2 fases: treinamento do modelo (80 %) e teste (20 %).

Para verificar a classificação, foi considerado a clusterização realizada previamente pelo K Médias.

Figura 13 – Resultados da clusterização da *Random Forest*



(a) Classificação em valores absolutos **(b)** Classificação em percentual

Fonte – Gerado a partir do script RandomForest.ipynb

Ao compararmos com a classificação realizada pela Regressão Logística, percebe-se que também houve baixos índices de erros, isto é, com menos de 7% de predições em classes não esperadas. Manteve-se uma proporção das classes muito próxima ao da Regressão Logística: 10,25% para a classe 1, 20,00% para a classe 2 e 61,50% para a classe 3.

Tabela 3 – Métricas para a *Random Forest*

Classe	Precisão	Recall	Falso Positivo	F-measure
1	0,86069	0,93177	0,01864	0,89482
2	0,82786	0,86639	0,05405	0,84669
3	0,96195	0,93301	0,07136	0,94726

Fonte – Gerado a partir do script spark.scala

4 Considerações Finais

4.1 Segmentação

O K Médias é um algoritmo que depende de uma quantidade inicial de *clusters*. Sem uma análise dos dados ou alguma proposta e/ou planejamento inicial, a única alternativa é definir algum valor aleatório e fazer uma análise sobre os *clusters* gerados. Quando se trabalha com poucos dados, seria possível calcular o *silhoutte score* para ter um embasamento sobre a quantidade de *cluster* inicial. Contudo, ao se trabalhar com um grande volume, uma possível alternativa poderia ser retirar amostras para calcular um valor médio. Neste trabalho, apesar da escolha inicial não ser o foco do estudo, utilizou-se uma análise superficial a partir de uma pequena amostra na qual gerou-se *boxplots* para visualizar as características dos dados em seus respectivos segmentos. Já na execução sobre a base completa, a verificação foi feita com a descrição de algumas métricas (média, desvio padrão, valor mínimo e máximo). Como o resultado da clusterização do K Médias serviu de entrada de informação para os algoritmos de classificação, não foi considerado necessário uma verificação mais profunda sobre os *clusters*. Se houvesse algum resultado discrepante, isso se verificaria na classificação dos dados.

4.2 Classificação

Ambos os algoritmos conseguiram classificar os registros com sucesso. Dado a segmentação gerada pela K Médias, tanto a Regressão Logística como a *Random Forest* obtiveram bons índices.

A Regressão Logística, quando executado em bases com dados que possuem um comportamento mais linear, tende a ser mais estável. Quando a base não possui tal característica, um dos problemas que pode ocorrer é o *overfitting*. Há alguma saídas para se contornar isso, que é a adicionando parâmetros l2 ou l1 na execução da Regressão Logística. Na base da *Loan Club*, verificou-se que isto não representou um problema muito significativo mas no caso de tentar melhorar os resultados, seria interessante um estudo em cada parâmetro disponível na execução do algoritmo. Há dois pontos de atenção para a regressão logística: caso seja intrínsico trabalhar com as variáveis categóricas, é necessário a transformação dessas *features* em variáveis *dummies*. Além disso, a normalização dos dados é recomendável para que não haja uma distorção muito grande nos cálculos do K Médias.

O *Random Forest* possui vantagens como a possibilidade de trabalhar com base de

dados que não tenham um comportamento linear. Além disso, elas podem tratar variáveis categóricas, fato que é complicado quando se trabalha com regressão logística, já que a *Random Forest* são árvores de decisões. Com o uso de *bagging* ou *boosting*, é possível trabalhar com uma grande quantidade de variáveis, além de realizar diversos treinos. Além disso, a *Random Forest* pode selecionar quais são as variáveis mais relevantes independente da quantidade de váriaveis que ela receba inicialmente.

Por fim, é importante ressaltar que os resultados provenientes da execução dos algoritmos não representam uma verdade absoluta e inquestionável. Trata-se de uma modelagem que busca se aproximar da realidade mas que inerentemente fica condicionada a limitações. Devem ser compreendidos como um possível direcionamento ou solução que possui um embasamento científico. Todavia, os algoritmos de *Machine Learning* vem se mostrando ferramentas poderosas e que podem ser úteis em diversas áreas de setores distintos.

4.3 Sugestão para trabalhos futuros

Este trabalhou limitou-se a estudar apenas 1 algoritmo de segmentação e 2 de classificação. Existem outros modelos que se adequam melhor a cada situação. Seria interessante que fossem estudados para ter um parâmetro de comparação. Além disso, para cada algoritmo também existem ajustes que podem ser feitos para se obter um resultado ainda melhor. Não obstante a esses pontos, também seria possível explorar ainda mais o banco de dados com os dados que foram excluídos propositalmente para reduzir o escopo do trabalho.

Referências

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. 1. ed. [S.l.]: Addison Wesley, 1999.

BEZERRA, E. *Introdução à tarefa de agrupamento: lições de inform*. 2. ed. Rio de Janeiro: UniverCidade Editora, 2006.

CONCEIÇÃO, H. *K-means clustering*. Disponível em: <http://www.onmyphd.com/?p=k-means.clustering>. Acesso em: 21 jun 2016.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*. [S.l.]: Springer, 2011. ISBN 0387848576.

KOTLER, P. *Administração de Marketing: análise, planejamento, implementação e controle*. 2. ed. São Paulo: Atlas, 1992.

MACQUEEN, J. *Some Methods for classification and Analysis of Multivariate Observations*. Berkeley, University of California Press, 1967. Disponível em: <https://projecteuclid.org/download/pdf_1/euclid.bsmsp/1200512992>. Acesso em: 16 jun 2016.

MAYER-SCHöNBERGER, V.; CUKIER, K. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. [S.l.]: Eamon Dolan/Houghton Mifflin Harcourt, 2013. ISBN 0544002695.

SMITH, C. *By the numbers: a gigantic list of google stats and facts*. 2016. Disponível em: <http://expandedramblings.com/index.php/by-the-numbers-a-gigantic-list-of-google-stats-and-facts/>. Acesso em: 07 jun 2016.

TSUJI, G. K. *Repositório de código*. Disponível em: <https://github.com/gustavotsuji/gustavotsuji.github.io/tree/master/tcc/scripts>. Acesso em: 20 nov 2016.

VAGATA, P.; WILFONG, K. *Scaling the Facebook data warehouse to 300 PB*. 2014. Disponível em: <https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/>. Acesso em: 07 jun 2016.

ZIKOPOULOS, P.; EATON, C. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. [S.l.]: McGraw-Hill Osborne Media, 2011. ISBN 0071790535.

Anexos

ANEXO A – Tabela de dados

Para o caso do *Loan Club*, nem todas as variáveis foram utilizadas. Isto se deve ao fato de que algumas das variáveis em questão são categóricas, outras que representam alguma descrição. Tratam-se de campos que provem informações substanciais mais que para o escopo deste trabalho foram desconsideradas. No caso de variáveis categóricas, seria possível uma conversão para variáveis *dummies*, mas que para o caso do K Médias não devem ser consideradas. Já para os campos que possuem uma informação textual, existem algoritmos que podem fazer análises do conteúdo como *word to vec* ou *bag of words*.

Tabela 4 – Tabela de campos utilizados para a análise do banco de dados *Loan Club*

Coluna	**Descrição**
funded_amnt	Total do montante comprometido para aquele empréstimo até este momento
funded_amnt_inv	Total do montante comprometido pelos investidores para aquele empréstimo até este momento
grade	Nota atribuída pelo *Loan Club*
loan_amnt	A quantia do empréstimo do mutuário, podendo ter eventualmente descontos pelo departamento de crédito
term	Quantidade de pagamentos do empréstimos que pode ser de 36 ou 60 meses
int_rate	Taxa de juros do empréstimo
installment	Pagamento devido pelo tomador de empréstimo
annual_inc	Renda anual relatada pelo devedor durante o cadastro
dti_joint	A razão entre a quantia total de dívidas paga pelos comutuários (desconsiderando hipotecas e o empréstimo feito pelo *Loan Club*) dividido pela renda mensal dos comutuários
delinq_2yrs	Incidências de inadimplência (eventos ocorridos dentro de 30 dias) dentro de um período de 2 anos
inq_last_6mths	Quantidade de pedidos de empréstimos (excluindo carros e hipotecas)
open_acc	Número de linhas de créditos abertas para o mutuário

Tabela 4 – Tabela de campos utilizados para a análise do banco de dados *Loan Club*

Coluna	**Descrição**
pub_rec	Número de d̈erogatory public records(̈trata-se de um registro que pode ser considerado negativo para o mutuário pois indica um risco e denegrindo sua reputação e dificultando a possibilidade de aquisição de outros produtos. Entre os valores, é possível W e F
revol_bal	Total de crédito que não foi pago
total_acc	Número toral de linhas de créditos usada pelo mutuário
out_prncp	Saldo devedor
out_prncp_inv	Saldo devedor sobre a porção de capital financiado por investidores
total_pymnt	Pagamentos recebidos até o momento presente sobre o valor financiado
total_pymnt_inv	Pagamentos recebidos até o momento presente sobre o valor financiado pelo investidor
total_rec_int	Juros recebido até a data presente
total_rec_late_fee	Taxas de atraso recebido até a presente data
total_rec_prncp	Capital financiado recebido até a data presente
collection_recovery_fee	valor de imposto recuperado de c̈harge off(̈declaração de não quitação de uma dívida)
last_pymnt_amnt	Valor total de pagamento recebido
recoveries	valor bruto recuperado de c̈harge offv̈erificados

Tabela 5 – Tabela de campos disponíveis em *Loan Club* e que não foram utilizados

Coluna	**Descrição**
addr_state	Estado de moradia do mutuário
annual_inc_joint	Valor declarado no momento do cadastro da renda anual do co-mutuários (um emprestimo eh realizado com co-mutuarios quando mais de uma pessao eh sao financiadas)
application_type	Indica se o empréstimo será individual ou se será feito por mais de uma pessoa (co-mutuários)
earliest_cr_line	Mês mais recente em que o mutuário abriu uma linha de crédito
emp_length	Período em que o devedor está empregado no trabalho atual, na qual 0 representa menos de 1 ano e 10 pode ser de 10 a mais anos

Tabela 5 – Tabela de campos disponíveis em *Loan Club* e que não foram utilizados

Coluna	**Descrição**
emp_title	Descrição breve do emprego do devedor
fico_range_high	Limite superior do score proveniente do FICO a qual o devedor está enquadrado
fico_range_low	Limite inferior do score proveniente do FICO a qual o devedor está enquadrado
home_ownership	Indica o status do tipo de moradia do tomador de empréstimo RENT (aluguel), OWN (casa própria), MORTGAGE (hipoteca), OTHER (outros)
id	variável de identificação de cada cliente da *Loan Club*
initial_list_status	Status inicial e interno da *Loan Club* do mutuário. Varia entre W e F
is_inc_v	Indica se a renda foi verificada pela *Loan Club* ou não ou se a fonte de renda foi verificada
issue_d	Mês no qual o empréstimo foi financiado
last_credit_pull_d	O mês mais recente desde que o mutuário adquiriu crédito no financiamento
last_fico_range_high	O limite superior do último score do FICO em que o mutuário foi classificado
last_fico_range_low	O limite inferior do último score do FICO em que o mutuário foi classificado
last_pymnt_d	Último mês em que foi recebido um pagamento
loan_status	Status atual do empréstimo
mths_since_last_delinq	Número de meses desde a última inadimplência do mutuário
mths_since_last_major_derog	Meses desde a pior nota (em um período de 90 dias)
mths_since_last_record	Núnmero de meses desde o último registro de informações públicas (antecedência criminal)
next_pymnt_d	Data do próximo pagamento
policy_code	Código interno da *Loan Club*, variando entre 1 e 2
purpose	Categoria escolhida pelo mutuário no momento da solicitação do crédito
pymnt_plan	Indica se o pagamento do plano está em dia
revol_util	Taxa de utilização da linha de crédito (revolving account)
sub_grade	Sub nota atribuída pelo Loan Club
title	Tipo de empréstimo realizado pelo mutuário

Tabela 5 – Tabela de campos disponíveis em *Loan Club* e que não foram utilizados

Coluna	**Descrição**
verified_status_joint	Indica se a renda dos co-mutuários foi verificada pela *Loan Club*, se não foi ou se a fonte de renda foi verificada
zip_code	Os primeiros 3 números de código postal fornecida pelo mutuário
open_acc_6m	Número de negócios abertos nos últimos 6 meses
open_il_6m	Número de operações de financiamento ativos
open_il_12m	Número de contas de operações de financiamento abertas nos últimos 12 meses
open_il_24m	Número de contas de operações de financiamento abertas nos últimos 24 meses
mths_since_rcnt_il	Quantidade de meses desde que a mais recente conta de operações de financiamento aberta
total_bal_il	Saldo total de todas as installment accounts
il_util	Razão entre o total de saldo sobre o limite de crédito para todas as installment accounts
open_rv_12m	Quantidade de contas (revolving accounts) nos últimos 12 meses
open_rv_24m	Quantidade de contas (revolving accounts) abertas nos últimos 24 meses
max_bal_bc	Saldo atual de todas as contas abertas (revolving accounts)
all_util	Saldo de limite de créditos para todas as operações
total_rev_hi_lim	Total limite de revolving créditos
inq_fi	Número de solicitações de crédito para finanças pessoais
total_cu_tl	Número de negócios financeiros
acc_now_delinq	Número de contas nas quais o mutuário está agora inadimplente
tot_coll_amt	Saldo total de collection accounts
tot_cur_bal	Saldo total de todas as contas do mutuário

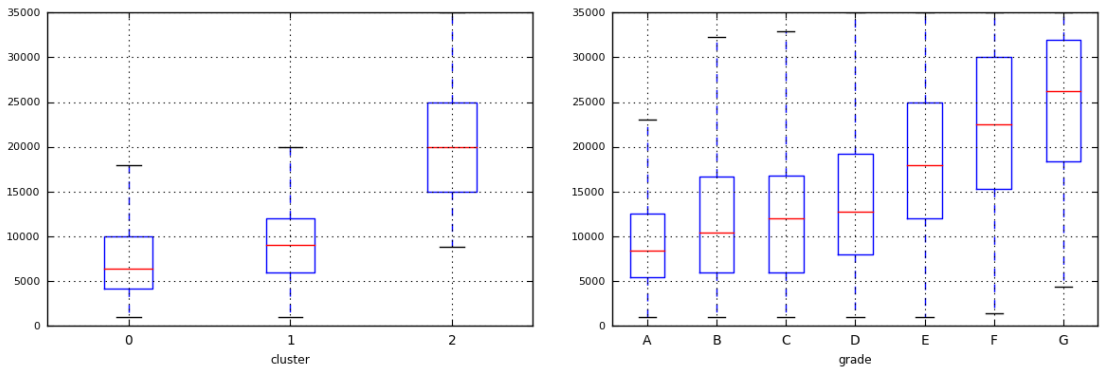
ANEXO B – Análise dos *clusters* vs *grades*

Durante os estudos, foram feitas algumas análises com *boxplots* para entender os *clusters* gerados. Também foram feitos uma comparação com os clientes de acordo com os seus grades para ver se era possível encontrar algum padrão nos registros. Detectar padrões de dados é uma tarefa complexa e visto que não se trata do foco do trabalho, o estudo limitou-se a verificar apenas alguma ocorrência de padrão.

Nos anexos, estão os outros *boxplots* gerados que não constam na parte de Resultados.

Figura 14 – *Boxplots* de funded_amnt

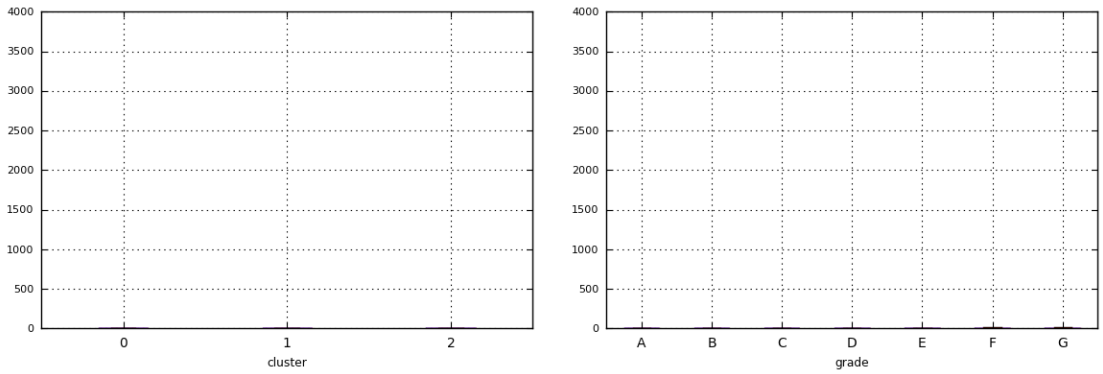
(a) Cluster **(b)** Grade



Fonte – Gerado a partir do script ClusterAnalysis.ipynb

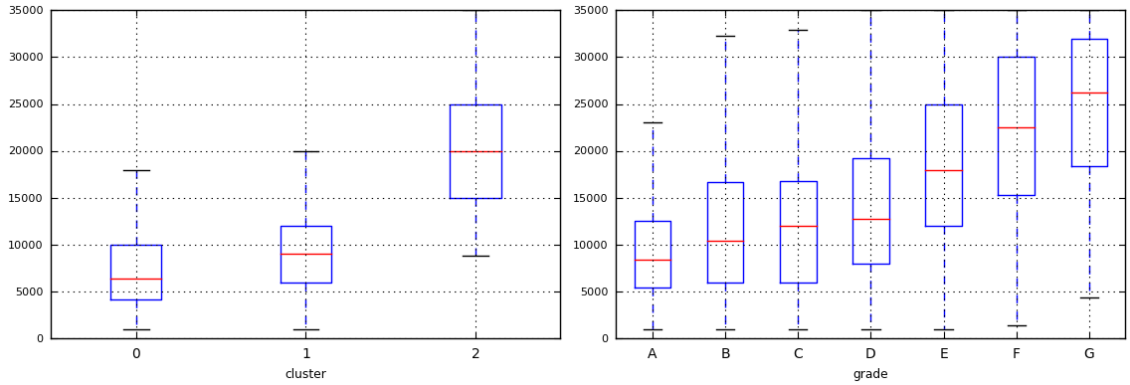
Figura 15 – *Boxplots* de collection_recovery_fee

(a) Cluster **(b)** Grade



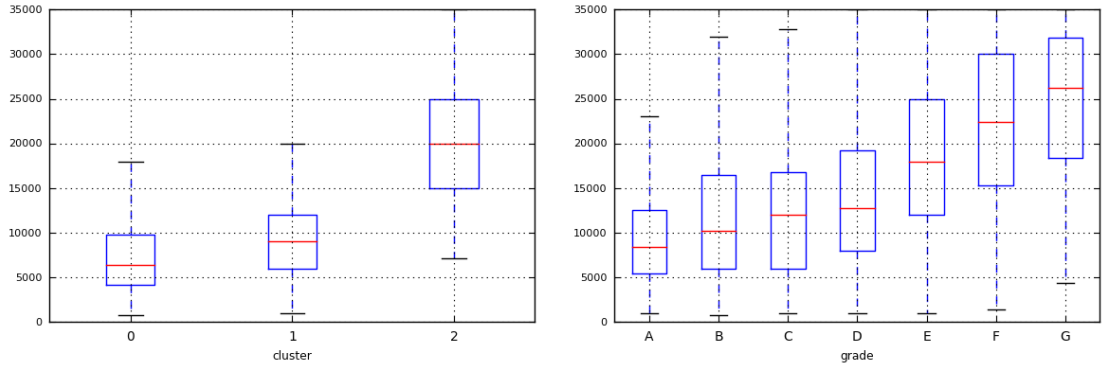
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 16 – *Boxplots* de funded_amnt



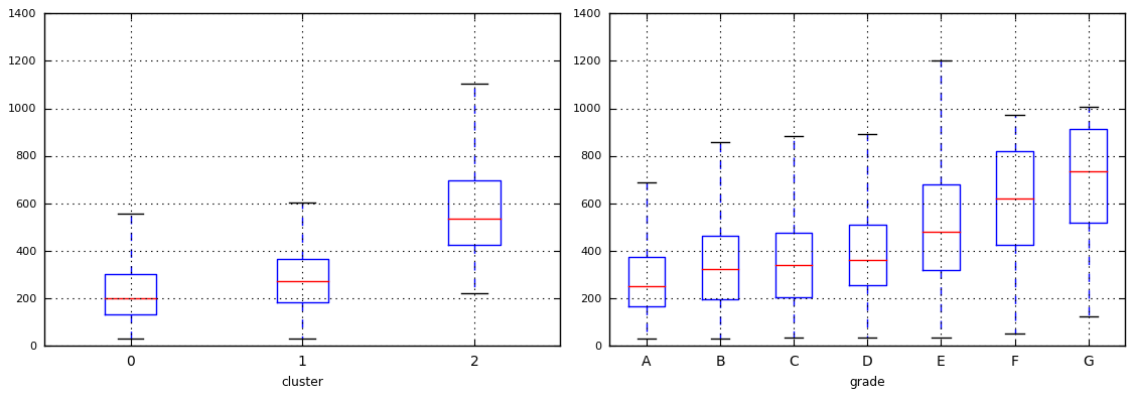
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 17 – *Boxplots* de funded_amnt_inv



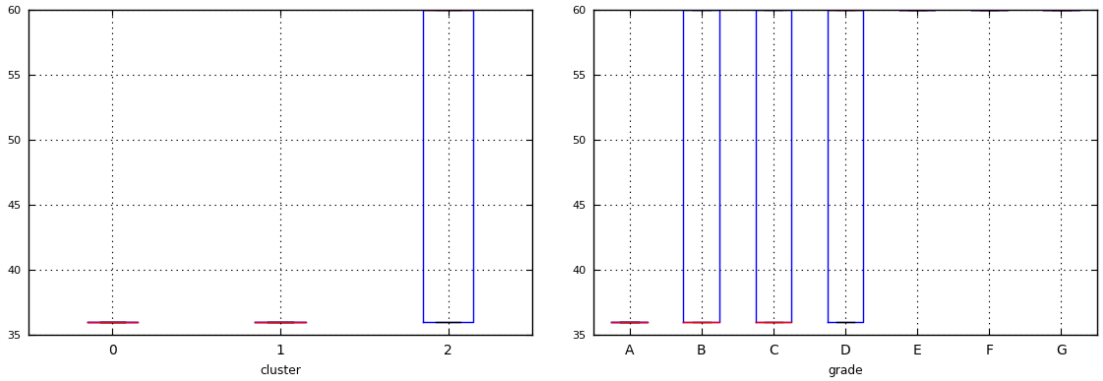
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 18 – *Boxplots* de installment



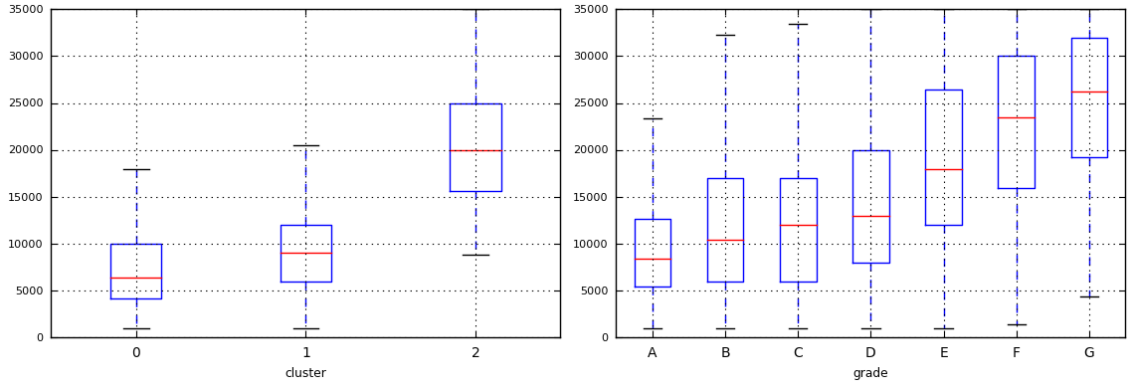
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 19 – *Boxplots* de term_float_fee



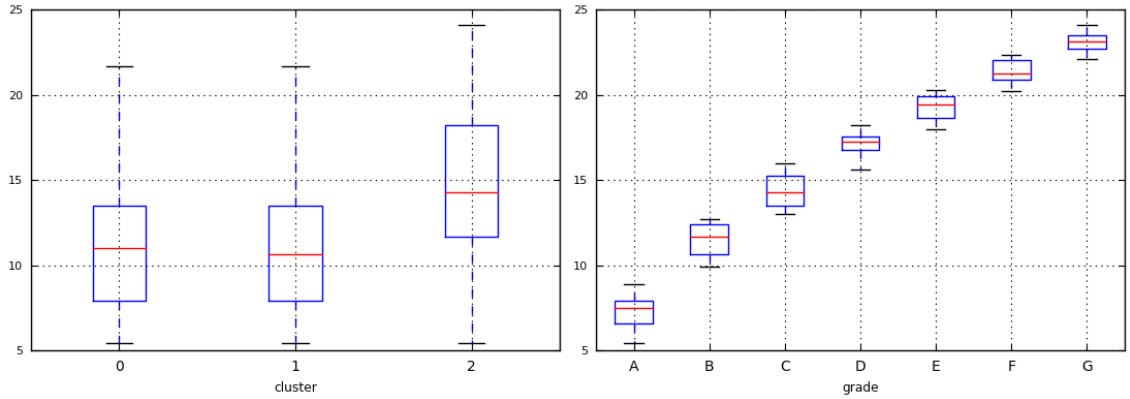
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 20 – *Boxplots* de loan_amnt



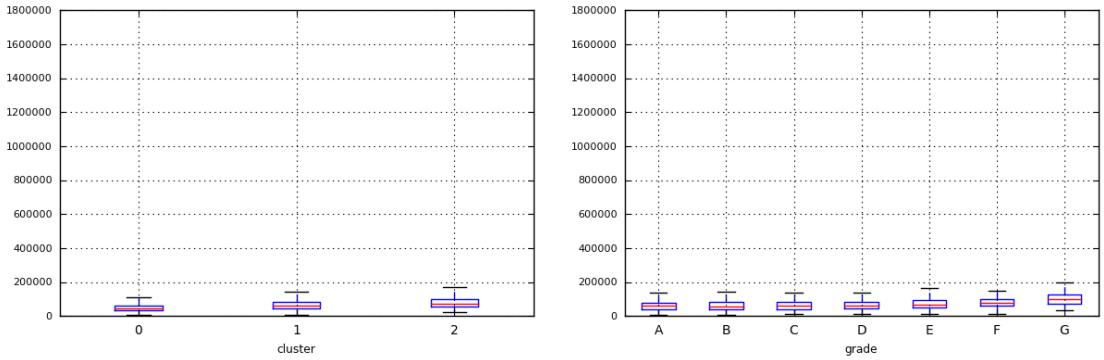
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 21 – *Boxplots* de int_rate_float



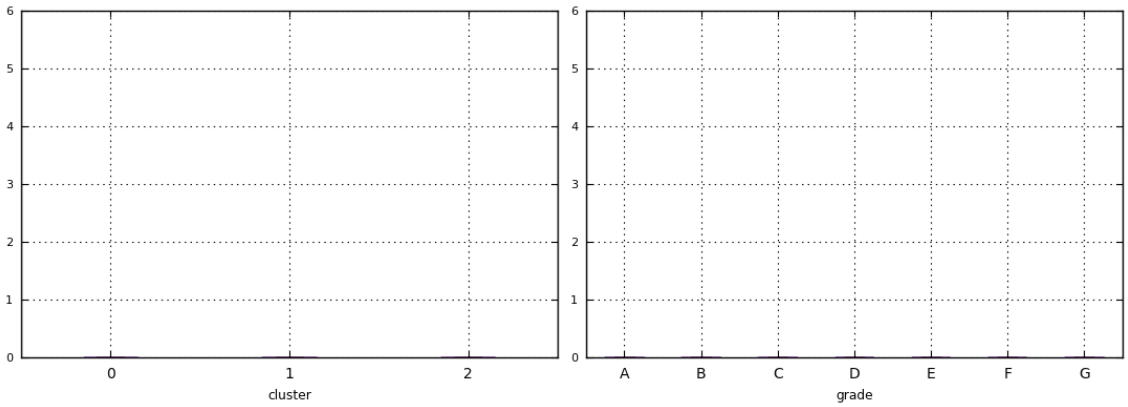
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 22 – *Boxplots* de annual_inc



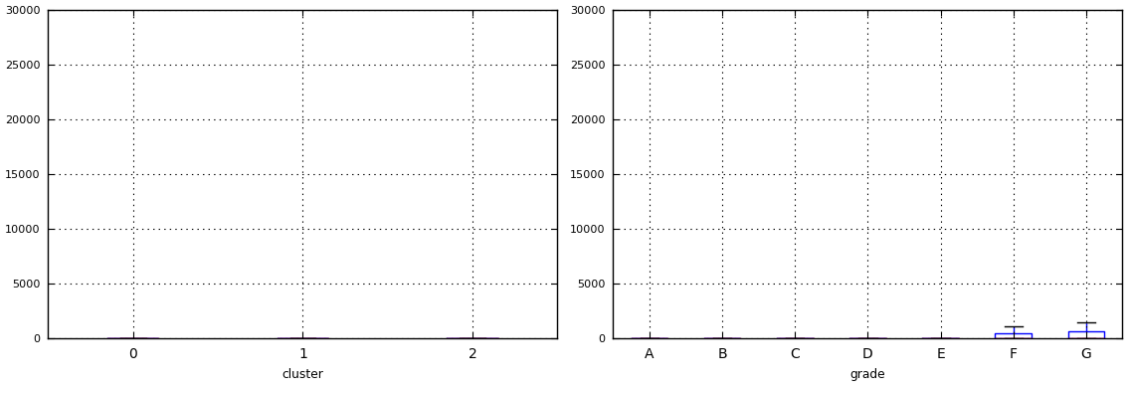
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 23 – *Boxplots* de delinq_2yrs



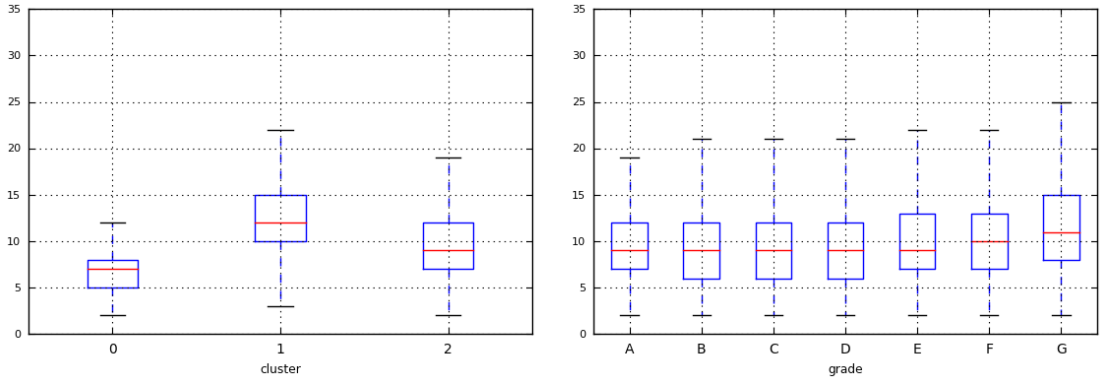
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 24 – *Boxplots* de recoveries



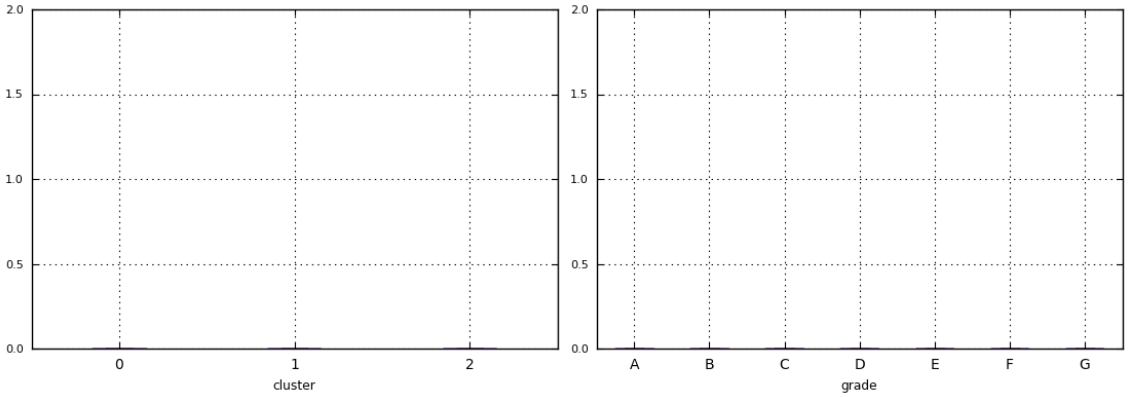
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 25 – *Boxplots* de open_acc



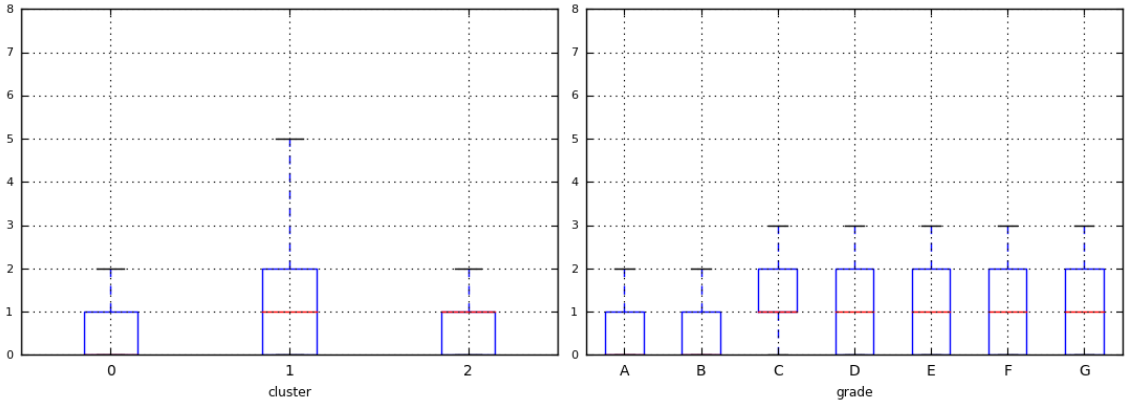
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 26 – *Boxplots* de pub_rec



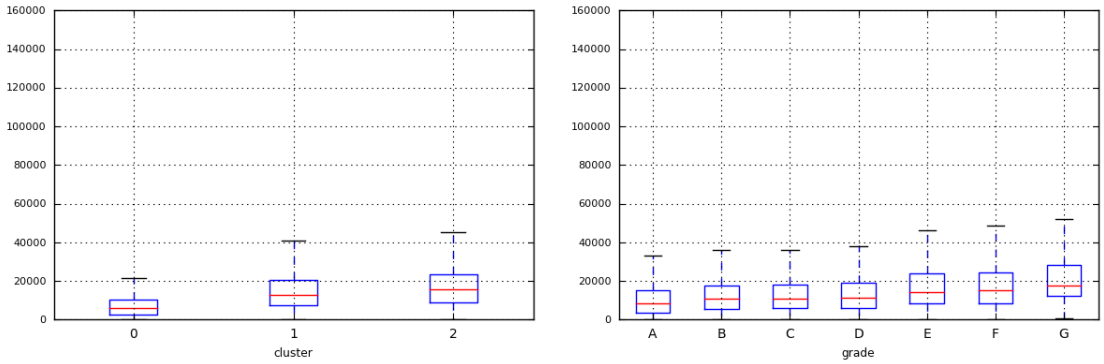
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 27 – *Boxplots* de inq_last_6mths



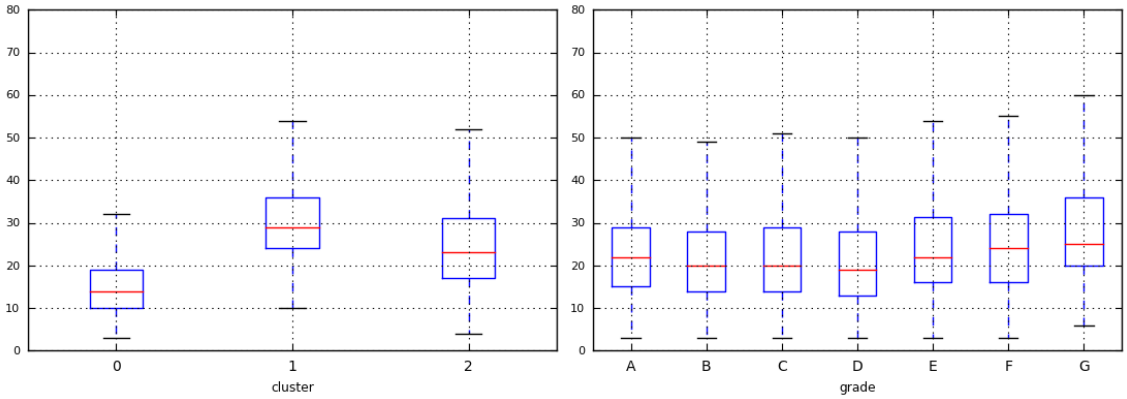
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 28 – *Boxplots* de revol_bal



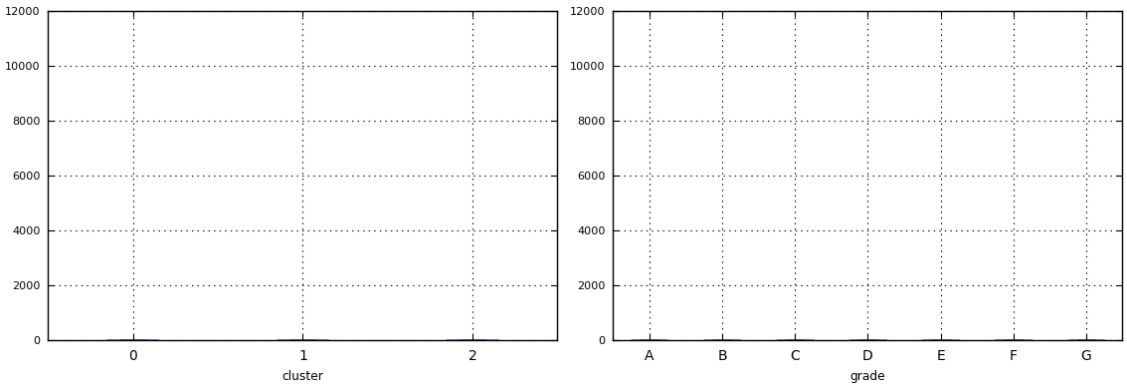
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 29 – *Boxplots* de total_acc



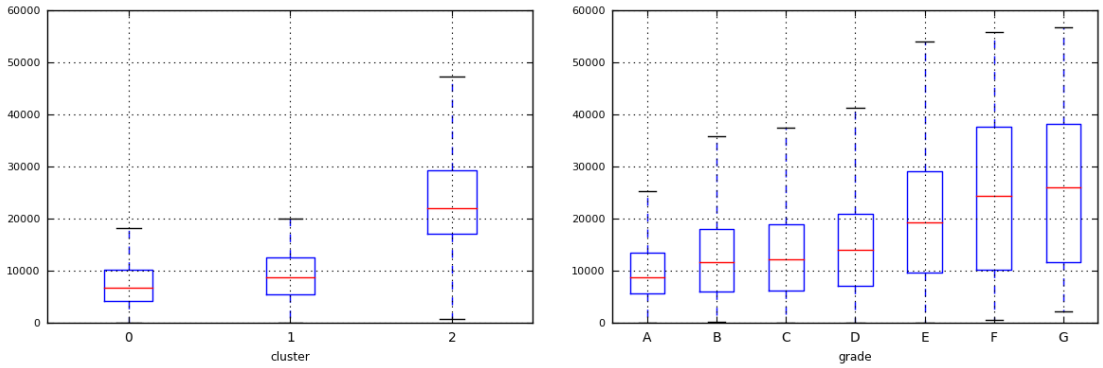
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 30 – *Boxplots* de out_prncp



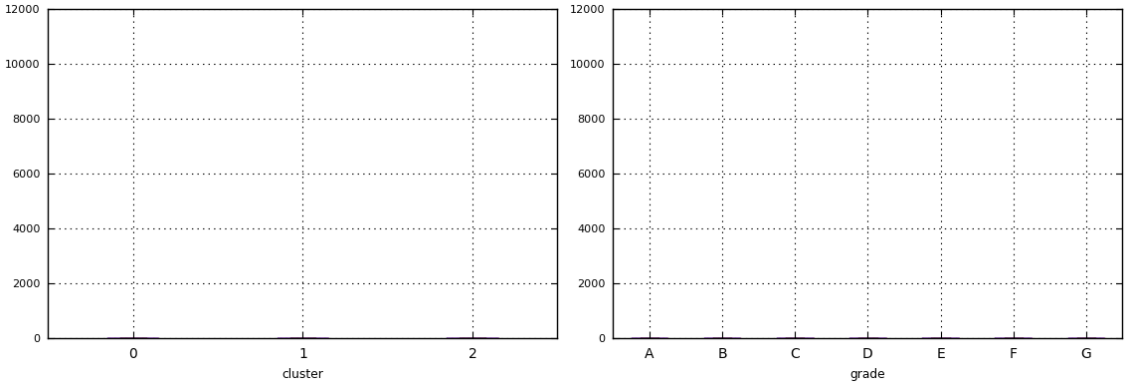
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 31 – *Boxplots* de total_pymnt



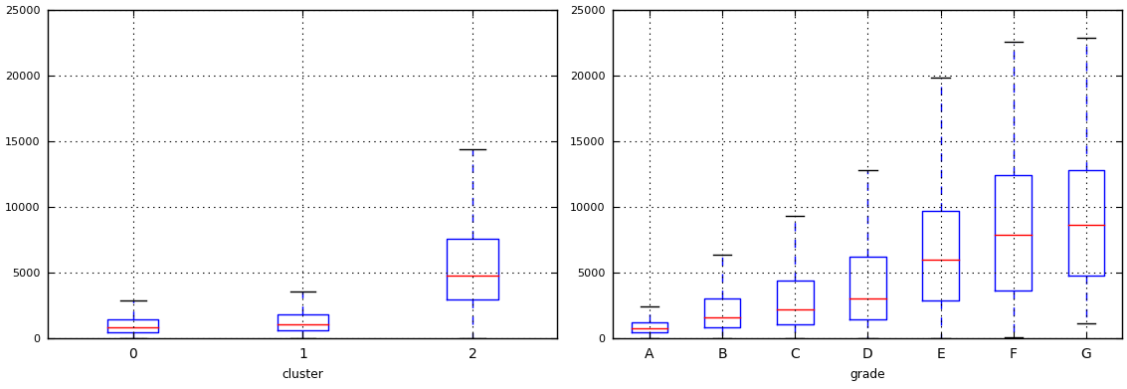
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 32 – *Boxplots* de out_prncp_inv



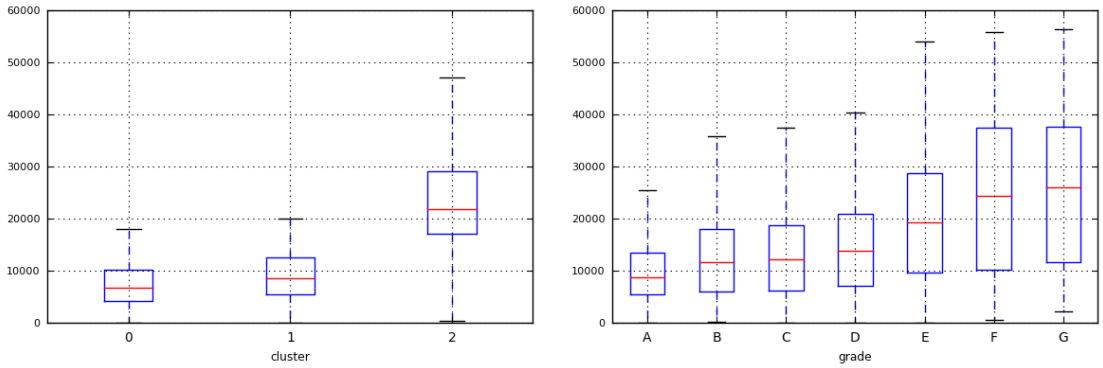
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 33 – *Boxplots* de total_rec_int



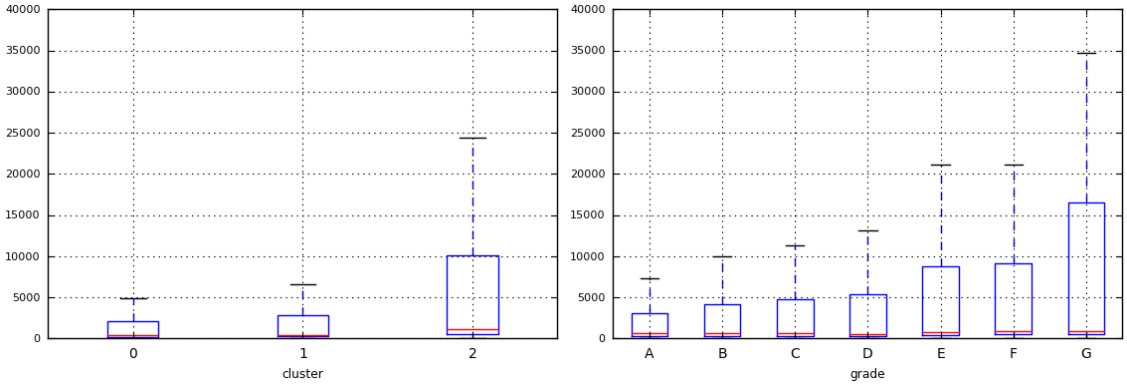
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 34 – *Boxplots* de total_pymnt_inv



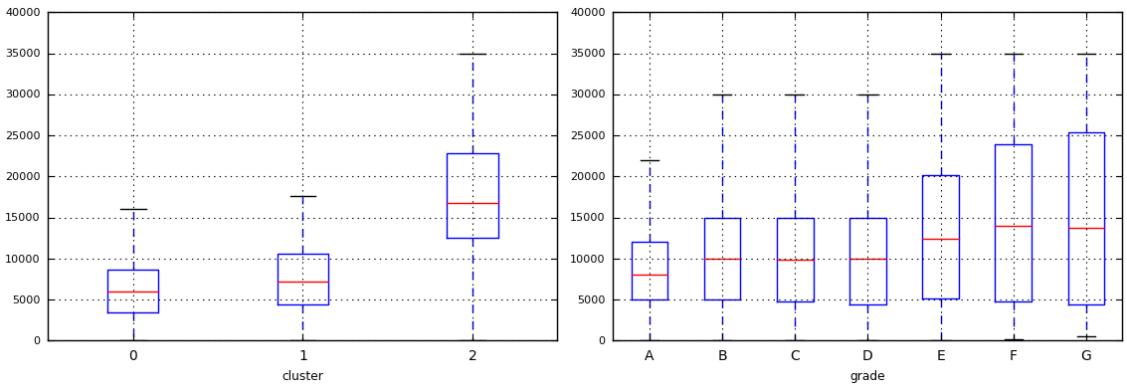
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 35 – *Boxplots* de last_pymnt_amnt



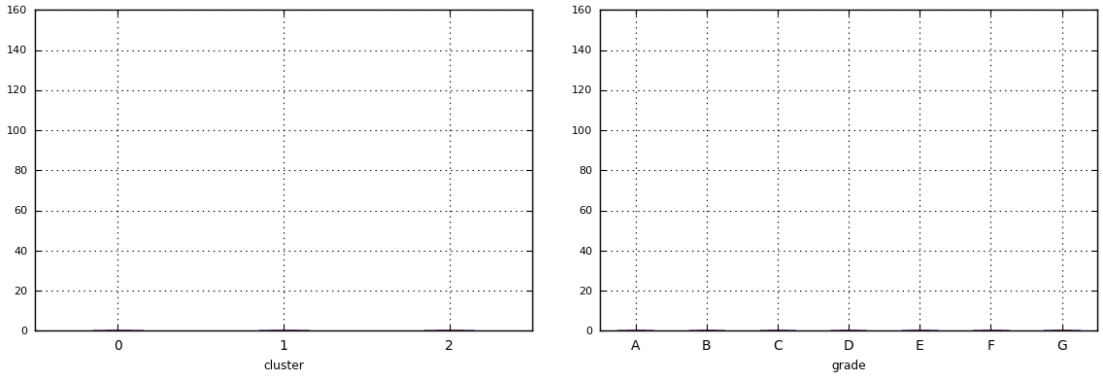
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 36 – *Boxplots* de total_rec_prncp



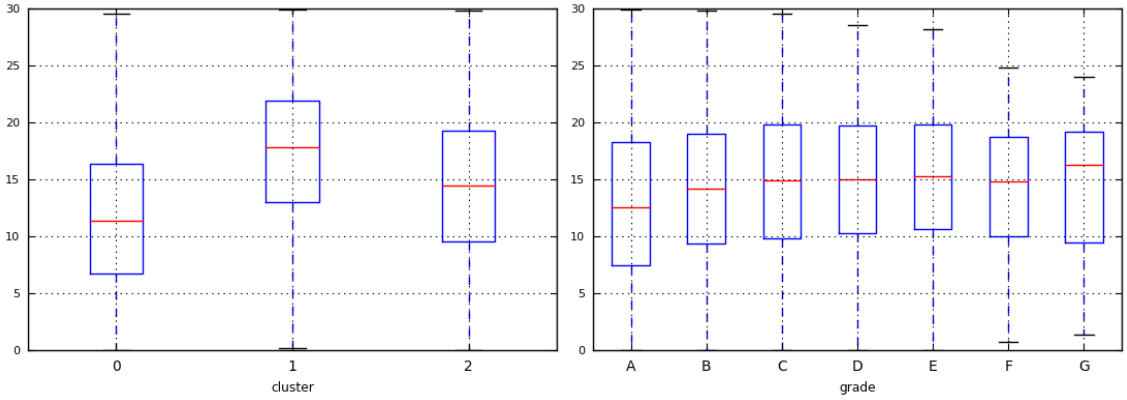
Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 37 – *Boxplots* de total_rec_late_fee



Fonte – Gerado a partir do script ClusterAnalysis.ipynb

Figura 38 – *Boxplots* de dti



Fonte – Gerado a partir do script ClusterAnalysis.ipynb

ANEXO C – Correlacao entre variaveis

Tabela 6 – Correlação entre variáveis

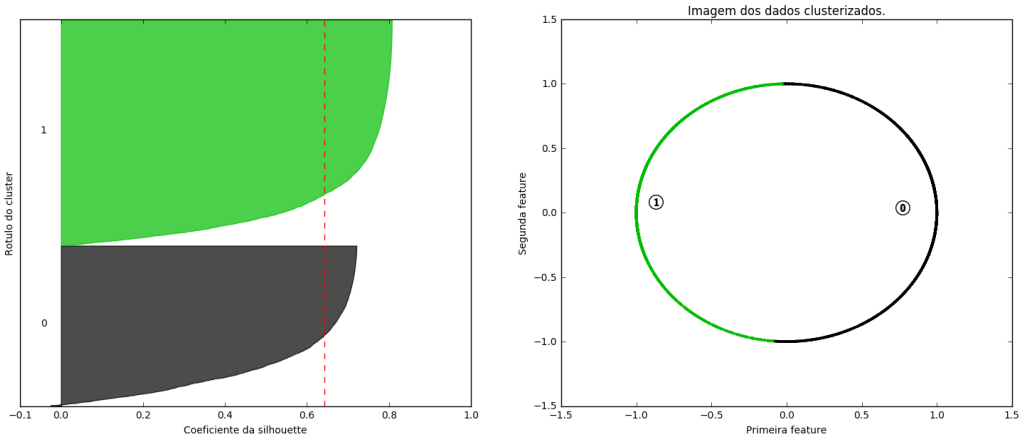
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1.000	0.999	0.993	0.487	0.326	0.956	0.373	0.033	-0.037	0.025	0.164	-0.065	0.337	0.259	0.325	0.325	0.900	0.899	0.839	0.738	0.068	0.169	0.130	0.440
1	0.999	1.000	0.992	0.484	0.326	0.957	0.372	0.032	-0.036	0.026	0.164	-0.066	0.335	0.258	0.324	0.324	0.900	0.900	0.840	0.739	0.068	0.168	0.129	0.439
2	0.993	0.992	1.000	0.498	0.333	0.946	0.373	0.034	-0.036	0.022	0.165	-0.064	0.338	0.262	0.328	0.328	0.893	0.892	0.831	0.737	0.068	0.168	0.129	0.439
3	0.487	0.484	0.498	1.000	0.552	0.264	0.104	0.074	0.012	0.053	0.080	-0.007	0.144	0.134	0.421	0.421	0.429	0.426	0.281	0.614	0.048	0.153	0.122	0.267
4	0.326	0.326	0.333	0.552	1.000	0.291	0.095	0.098	0.149	0.188	0.054	0.082	0.121	-0.002	0.233	0.233	0.305	0.305	0.145	0.554	0.095	0.154	0.124	0.168
5	0.956	0.957	0.946	0.264	0.291	1.000	0.387	0.022	-0.027	0.030	0.164	-0.060	0.334	0.241	0.219	0.219	0.866	0.867	0.836	0.648	0.072	0.144	0.110	0.399
6	0.373	0.372	0.373	0.104	0.095	0.387	1.000	-0.185	0.035	0.046	0.199	-0.029	0.367	0.310	0.094	0.094	0.367	0.367	0.365	0.259	0.034	0.029	0.032	0.200
7	0.033	0.032	0.034	0.074	0.098	0.022	-0.185	1.000	-0.049	0.023	0.280	-0.017	0.221	0.224	0.049	0.049	0.029	0.029	-0.001	0.083	0.005	0.032	0.030	-0.015
8	-0.037	-0.036	-0.036	0.012	0.149	-0.027	0.035	-0.049	1.000	-0.006	0.007	0.003	-0.066	0.069	0.014	0.014	-0.026	-0.026	-0.048	0.031	0.036	0.012	0.015	-0.010
9	0.025	0.026	0.022	0.053	0.188	0.030	0.046	0.023	-0.006	1.000	0.097	0.030	-0.018	0.116	-0.016	-0.015	0.014	0.016	-0.002	0.042	0.013	0.022	0.029	0.057
10	0.164	0.164	0.165	0.080	0.054	0.164	0.199	0.280	0.007	0.097	1.000	-0.006	0.279	0.674	0.046	0.046	0.158	0.158	0.149	0.126	0.008	0.038	0.031	0.080
11	-0.065	-0.066	-0.064	-0.007	0.082	-0.060	-0.029	-0.017	0.003	0.030	-0.006	1.000	-0.077	-0.027	-0.023	-0.023	-0.060	-0.061	-0.069	-0.018	-0.009	-0.011	-0.014	-0.023
12	0.337	0.335	0.338	0.144	0.121	0.334	0.367	0.221	-0.066	-0.018	0.279	-0.077	1.000	0.314	0.130	0.130	0.326	0.325	0.302	0.272	0.016	0.059	0.053	0.140
13	0.259	0.258	0.262	0.134	-0.002	0.241	0.310	0.224	0.069	0.116	0.674	-0.027	0.314	1.000	0.062	0.062	0.237	0.237	0.236	0.160	-0.005	0.045	0.047	0.164
14	0.325	0.324	0.328	0.421	0.233	0.219	0.094	0.049	0.014	-0.016	0.046	-0.023	0.130	0.062	1.000	1.000	0.363	0.361	0.218	0.612	0.036	-0.047	-0.034	-0.153
15	0.325	0.324	0.328	0.421	0.233	0.219	0.094	0.049	0.014	-0.015	0.046	-0.023	0.130	0.062	1.000	1.000	0.363	0.361	0.218	0.612	0.036	-0.047	-0.034	-0.153
16	0.900	0.900	0.893	0.429	0.305	0.866	0.367	0.029	-0.026	0.014	0.158	-0.060	0.326	0.237	0.363	0.363	1.000	1.000	0.957	0.803	0.029	0.030	0.038	0.504
17	0.899	0.900	0.892	0.426	0.305	0.867	0.367	0.029	-0.026	0.016	0.158	-0.061	0.325	0.237	0.361	0.361	1.000	1.000	0.958	0.802	0.029	0.029	0.037	0.503
18	0.839	0.840	0.831	0.281	0.145	0.836	0.365	-0.001	-0.048	-0.002	0.149	-0.069	0.302	0.236	0.218	0.218	0.957	0.958	1.000	0.611	-0.016	-0.113	-0.078	0.594
19	0.738	0.739	0.737	0.614	0.554	0.648	0.259	0.083	0.031	0.042	0.126	-0.018	0.272	0.160	0.612	0.612	0.803	0.802	0.611	1.000	0.108	0.096	0.088	0.168
20	0.068	0.068	0.068	0.048	0.095	0.072	0.034	0.005	0.036	0.013	0.008	-0.009	0.016	-0.005	0.036	0.036	0.029	0.029	-0.016	0.108	1.000	0.065	0.058	-0.067
21	0.169	0.168	0.168	0.153	0.154	0.144	0.029	0.032	0.012	0.022	0.038	-0.011	0.059	0.045	-0.047	-0.047	0.030	0.029	-0.113	0.096	0.065	1.000	0.797	-0.083
22	0.130	0.129	0.129	0.122	0.124	0.110	0.032	0.030	0.015	0.029	0.031	-0.014	0.053	0.047	-0.034	-0.034	0.038	0.037	-0.078	0.088	0.058	0.797	1.000	-0.060
23	0.440	0.439	0.439	0.267	0.168	0.399	0.200	-0.015	-0.010	0.057	0.080	-0.023	0.140	0.164	-0.153	-0.153	0.504	0.503	0.594	0.168	-0.067	-0.083	-0.060	1.000

Fonte – Gerado a partir do script ClusterAnalysis.ipynb

ANEXO D – *Silhoutte score*

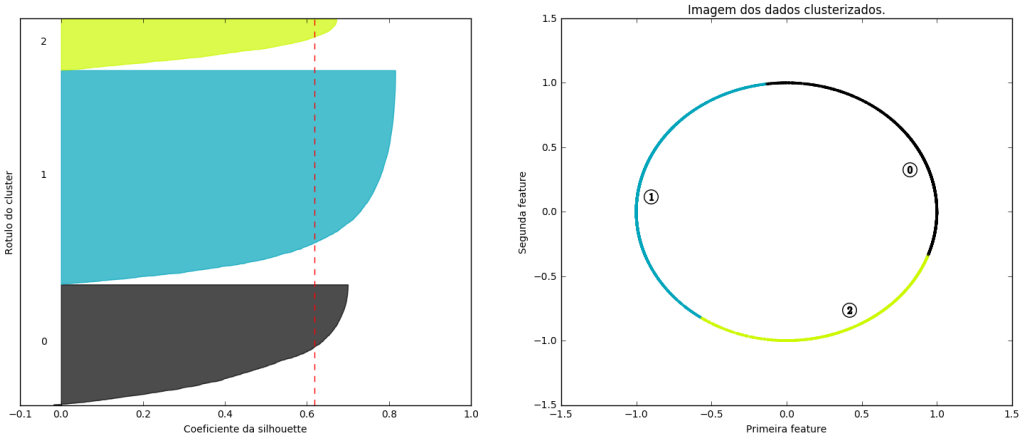
É possível fazer uma análise para verificar a quantidade de *clusters* ideal para a amostragem. A cada execução do algoritmo também calcula-se um *score* chamado *sillhouette*, que é uma forma de interpretar e mensurar a validação da consistência dos *clusters*, além de fornecer uma representação gráfica de quão bem os dados estão inseridos nos *clusters* (coesão) em comparação a outros (separação). Esse *score* varia de -1 a 1, na qual quanto mais próximo de 1, mais bem encaixado encontra-se o elemento com seu respectivo *cluster*.

Figura 39 – Análise de 2 *clusters*



Fonte – Gerado a partir do script KMeans.ipynb

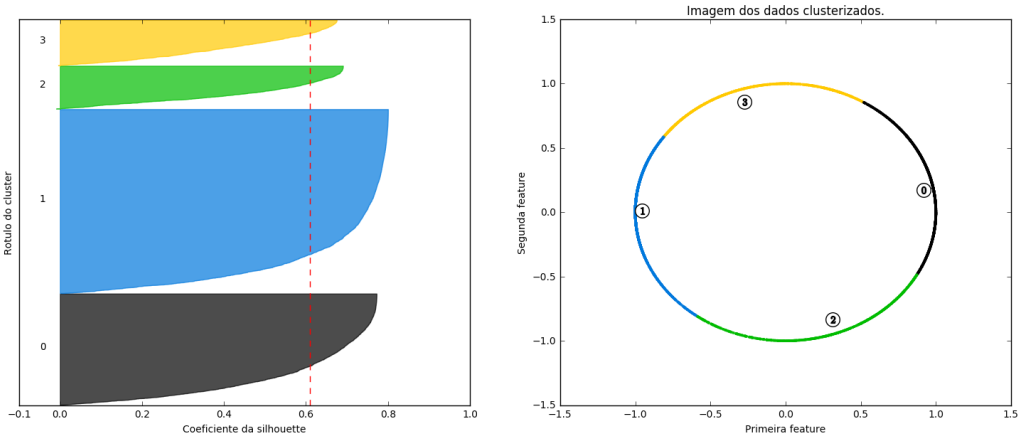
Figura 40 – Análise de 3 *clusters*



Fonte – Gerado a partir do script KMeans.ipynb

A princípio o *silhoutte score* seria uma métrica de avaliação para definir a quantidade ideal de *clusters* para a execução do K Médias. Contudo, essa métrica não foi implementada no *Apache Spark*, sendo útil apenas para uma análise exploratória.

Figura 41 – Análise de 4 *clusters*



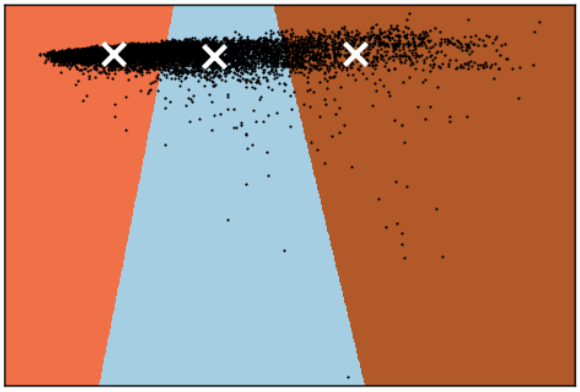
Fonte – Gerado a partir do script KMeans.ipynb

A primeira tentativa foi de separar a amostra em 2 *clusters*. Para essa divisão, foi obtida uma de 0,6428. Para a divisão em 3 *clusters*, foi obtida 0,6188. Já para 4 *clusters*, 0,6099.

ANEXO E – Diagrama de Voronoi

O diagrama de Voronoi é uma ferramenta utilizada para visualizar problemas que envolvem conceito de proximidade em um plano. Baseia-se no fato de que em um plano, existem pontos que estão mais próximos de uma fonte geradora do que de outra fonte, o resultado é um polígono de cujas distâncias entre a fonte e ponto são as menores possíveis. Portanto, é possível ter uma visualização espacial da distribuição dos registros, bem como os centróides.

Figura 42 – Visualização dos pontos no diagrama de Voronoi com os dados sem normalização



Fonte – Gerado a partir do script KMeans.ipynb

A imagem mostra um diagrama de Voronoi. Os polígonos externos se estendem infinitamente no plano, logo são desenhados como figuras abertas. Cada aresta do diagrama constitui um lugar onde os pontos são equidistantes em relação a dois locais. Os vértices dos polígonos estão ligados outras arestas sendo pontos de equidistânciacada região definidas.

Para a visualização desse diagrama, foi necessário realizar uma operação de redução de dimensionalidade de variáveis para 2 *features*. Neste estudo, aplicamos a construção do diagrama para 8000 registros. Embora a redução de dimensionalidade possa distorcer os dados, é possível ter uma visualização de como os dados estão distribuídos num espaço bidimensional. No contexto de *Big Data*, a geração de Diagrama de Voronoi pode ajudar numa análise inicial, sendo inviável gerá-lo para uma base completa e com muitos dados.