

# Classificação automática de requisições: uma aplicação de aprendizado de máquina sobre Cidades Inteligentes

Ivan Caramello de Andrade  
RA 181.777  
i.caramello@gmail.com

Rodney Rick  
RA 181.814  
rodneyrick@gmail.com

**Resumo**—Muito comentado nos dias atuais e com a principal finalidade de melhor explorar os recursos, o tema Cidades Inteligentes recebe cada vez mais investimentos para análises estatísticas, reconhecimento de padrões e aprendizado de máquina; esse trabalho abordará a automatização de um serviço de atendimento a requisições da cidade de Nova York, utilizando dados existentes de sua utilização passada e com a aplicação de algoritmos de aprendizado de máquina para agrupamento e reconhecimento.

## I. A PROPOSTA

Toda grande cidade possui uma infinidade de problemas associados, passando por uma grande quantidade de gravidades e categorias, os quais precisam de soluções específicas em lugares específicos para que todo o sistema continue funcionando corretamente. Mesmo em lugares modernos, onde o registro de tais problemas pode ser feito e acessado de maneira eletrônica, ainda é necessária a intervenção humana para receber, interpretar e categorizar corretamente cada reclamação, para apenas então endereçar o problema ao departamento e agência pertinentes. Neste trabalho, a base de dados de ocorrências de Nova York foi analisada com o intuito de se obter um classificador automático de reclamações, usando como entrada o local de origem, o horário, descrição da ocorrência, agência de abertura do chamado, bairro e endereço. Tal aplicação permitiria uma melhor alocação de recursos e um tempo de resposta melhor para os inúmeros incidentes que acontecem diariamente nessa cidade, e um modelo semelhante poderia ser gerado para outras cidades que possuam dados semelhantes sobre o histórico de reclamações.

## II. O CONJUNTO DE DADOS

O conjunto de dados utilizado para essa atividade vem do projeto *Open Data*<sup>1</sup> da cidade de Nova York, mais exatamente da base de dados "311 Service Requests"<sup>2</sup>. Essa base consiste de centenas de milhares de chamados relatando eventos problemáticos pela cidade, de infestações de roedores a vizinhos barulhentos, passando por problemas de aquecimento, iluminação pública e fluxo de veículos. Dados o tamanho e

a diversidade de uma como essa, a base de dados apresenta uma multiplicidade de classes e características para descrever esses chamados e problemas, bem como uma pluralidade de departamentos responsáveis por tomar as medidas necessárias.

**Features:** para cada ocorrência registrada, o horário, data, coordenadas e endereço do chamado são descritos, bem como o tipo de construção que originou a mesma, o tipo de ocorrência relatado, a descrição dada pelo requisitante, a agência alocada para resolver o problema, e a divisão político-geográfica (cidade e bairro) do chamado. Esses dados apresentam um misto de informações numéricas, textuais e categóricas, de um total de 53 *features*, foram consideradas 15 *features* com as diversas naturezas citadas, tamanhos e maneiras de se descrever.

**Desbalanceamento:** o dado que se propõe prever, dada uma chamada, é o tipo de reclamação que a originou. Mesmo com um grande número de chamados caindo no mesmo conjunto de categorias - ruas obstruídas e iluminação pública defeituosa são dois dos mais comuns - há uma grande quantidade de ocorrências possíveis que, mesmo sendo incomuns, estão presentes na cidade e precisam ser avaliadas e corrigidas. Esse tipo de variação na distribuição leva a um grande desbalanceamento nas classes-alvo, com algumas tendo menos do que 0.01% de todos os dados. Esse comportamento pode ser visto na Figura 1. Caso os chamados sejam ainda divididos por região espacial ou por tempo, essas disparidades podem ser ainda maiores, levando a um problema adicional quando tenta-se aprender e prever comportamentos: muitas das técnicas de aprendizado de máquina sofrem um bias relativo a classes mais comuns, ao mesmo tempo que o fato de algumas classes serem mais frequentes é um dado relevante da base.

**Extensão:** a quantidade de dados apresentados é, como se esperaria, tão grande quanto a frequência de reclamações sobre serviços públicos que acontecem numa das maiores cidades do mundo. Os dados são distribuídos não apenas entre uma grande quantidade de classes, mas também sobre muitos locais, períodos de tempo, endereços e bairros, o que pode levar a dificuldades quando todo esse conjunto é processado: dados categóricos precisarão de mais espaço para serem representados, dados textuais terão corpus de palavras muito grandes etc. A grande variação de valores nos dados numéricos pode até ajudar a separar eventos muito diferentes, mas a presença de comportamentos que mudam conforme o

<sup>1</sup><https://nycopendata.socrata.com/>

<sup>2</sup><https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9>

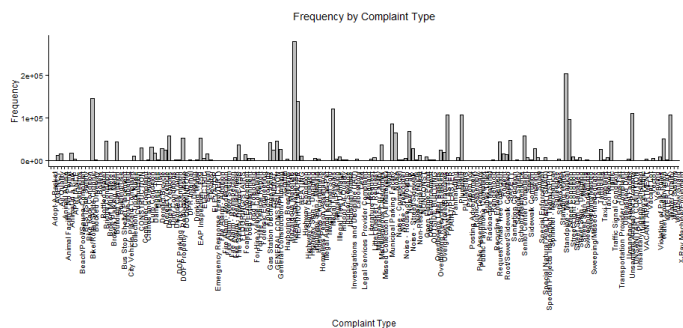


Figura 1: Classes descritas pela base de dados.

tempo podem levar modelos estritamente lineares a um tempo maior de treinamento ou mesmo a situações que não podem ser descritas por tal tipo de técnica.

**Dados não-numéricos:** a maior parte dos algoritmos de aprendizado de máquina depende de dados numéricos ou mesmo binários para analisar e aprender. No entanto, um problema que envolve endereços, cidades e descrições exige que técnicas de conversão sejam empregadas. Mesmo em atributos categóricos, como departamento ou bairro, uma simples enumeração leva a problemas quando se considera a distância entre elementos distintos, exigindo que tais transformações sejam adequadas para o modelo de dados sendo utilizado.

### III. EXPLORAÇÃO DOS DADOS

Trabalhando a modelagem de dados para disponibilizar a base com a melhor exploração para os algoritmos de aprendizado de máquina, considerando os tratamento e processamento das informações em dados processáveis por um algoritmo são etapas extremamente importantes para a solução do problema, uma vez que a representação do mundo real em termos computacionais costuma ser menos genérica e mais dependente do problema em questão do que as técnicas e algoritmos, estes já descritos na literatura. Um conjunto de técnicas foram explorados para tentar aproveitar as informações disponíveis da melhor maneira possível, tendo em vista a natureza dos dados existentes e as necessidades dos algoritmos a serem utilizados em seguida.

Os próximos tópicos, tem-se a exploração dados numéricos e textuais:

- **Discretization:** nem toda a informação disponível numa base de dados contribui para a compreensão ou descrição de um problema, seja por apresentar ruído ou por agregar pouca informação relevante para a resolução do problema. No caso de dados numéricos, informações onde a precisão representa meramente variações aleatórias no espaço ou tempo podem ser discretizadas, levando a menores tempos computacionais de execução e mesmo convergências mais rápidas, uma vez que os algoritmos não precisarão aprender quão relevantes ou não são essas variações. Um exemplo clássico desse caso são as casas mais baixas de tempo num *timestamp*: o minuto exato de uma ocorrência traz poucas informações a respeito da

mesma, apesar da hora ou do período do dia poderem ser mais relevantes.

- **Normalization:** dados numéricos podem ter diversas formas e origens e, ao tratarem de tipos de dado diferentes, raramente compartilham uma escala. Ainda assim, com o intuito de computá-los todos num mesmo algoritmo, a normalização dos mesmos se faz necessária, especialmente quando as ordens de grandeza e precisão são muito diferentes. As informações de coordenada, hora e dia da semana, por exemplo, podem ser normalizadas utilizando-se min-max para que possam ser comparadas lado a lado.
- **PCA/LDA:** dada a grande quantidade de dados apresentados e os diferentes significados entre eles, é possível aplicar técnicas matemáticas de redução de dimensionalidade, rotacionando os mesmos no espaço com o intuito de evidenciar as dimensões mais descritivas sobre o problema. Essas técnicas também auxiliam na comparação de medidas diferentes, uma vez que elementos comuns de atributos diferentes serão devidamente agrupados, auxiliando os algoritmos de aprendizado a trabalharem sobre novas *features* que se complementam ao descrever o problema em questão.
- **Textual data:** boa parte dos dados apresentados dá-se de maneira não numérica, seja por linguagem natural (como nas descrições) ou por palavras e frases que representam classes (como nas agências e bairros). Para tratá-los, é importante transformar tais informações em dados computáveis, extraíndo-se um corpus sobre o qual os textos possam ser mapeados. Enquanto a diversidade de vocabulário é relativamente extensa como um todo, cada um dos "textos" é relativamente pequeno, possuindo apenas uma ou duas frases. Assim, análises focadas na simples ocorrência de palavras são mais relevantes do que análises mais complexas, considerando estruturas textuais ou ignorando palavras auxiliares.

### IV. AS TÉCNICAS

Para os experimentos desta seção, trabalha-se a identificação e atendimento de um novo chamado, através da utilização de algumas etapas:

Para a Localização dos eventos:

**Clustering:** uma vez que o problema apresentado possui dados espaciais, o agrupamento de eventos traz uma informação relevante: como distribuir um conjunto de pontos de tal maneira que a distância de atendimento seja mínima. Nesse caso em específico, esses pontos - representantes de grupos formados por um algoritmo de agrupamento - serviriam como referência para o estabelecimento de bases de atendimento, minimizando a distância às ocorrências e, consequentemente, o tempo de atendimento e os custos de deslocamento. Uma análise mais fina do problema levou à observação de que, dada a variada distribuição de comportamentos sobre o tempo, estabelecer centroides 'médios' no tempo teria pouca relevância prática caso essas janelas de tempo fossem muito extensas. Assim, os eventos foram agrupados sobre o espaço em janelas de 2 horas, propondo assim um conjunto de  $N$  bases móveis

de atendimento que, ao longo do dia, se deslocariam para otimizar sua operação. Com o tipo de evento alocado a cada uma delas, seria também possível especializá-las, colocando equipes focadas nos problemas mais comuns da região - fiscais de poluição sonora, detetizadores, equipes de reparos elétricos etc.

Para a Classificação:

**Logistic Regression:** é um dos algoritmos mais clássicos para classificação, mas não necessariamente um dos mais adequados para o problema sendo considerado. Dada a pluralidade de dados categóricos e textuais, a precisão de um classificador essencialmente linear diminui com a introdução de uma grande quantidade de variáveis binárias. Mais do que isso, um problema multi-classe com mais de uma centena delas leva a um set extenso e de difícil treinamento, seja através de *one-versus-all* ou *one-versus-one*. Finalmente, com um mesmo tipo de ocorrência podendo ocorrer em regiões geograficamente distintas da cidade, classificadores lineares apresentam dificuldades em corretamente resolver tais situações, exigindo ou modelagens de dados muito específicas, ou aumento de complexidade.

**Neural Networks:** sendo modelos naturalmente não-lineares, redes neurais são uma técnica promissora para se solucionar um problema com tamanha distribuição de classes sobre os dados. A natureza distinta dos dados também se adequa bem à estrutura dessas redes, uma vez que as funções de ativação operam sobre tanto valores contínuos como discretos e, uma vez dentro da rede, tais dados são transformados em sinais de ativação semelhantes. No entanto, a grande quantidade de classes não-sequenciais leva a camadas de saída muito extensas e propensas a erros, uma vez que a quantidade de bits errados será, muitas vezes, uma ordem de grandeza maior do que o de certos, o que limita a velocidade de aprendizado do *backpropagation*. É possível descrever as classes como combinações aleatórias de bits de saída, reduzindo parcialmente esse problema ao aumentar o tamanho de representação e a diferença absoluta entre classes de saída.

**Decision Trees:** árvores de decisão trazem duas principais vantagens, como classificadores: permitem soluções não-continuas no espaço de dados e podem ser naturalmente multi-classe. Ambas essas características são compatíveis com o problema sendo tratado, fazendo com que as mesmas sejam uma solução potencialmente atrativa para se solucionar o problema em questão. A principal limitação das mesmas, que diz respeito a valores contínuos cujos *thresholds* precisam ser encontrados para cada divisão, é limitada quando se considera que boa parte dos (já poucos) atributos contínuos são transformáveis em classes discretas sem maiores perdas de informação. Dada a complexidade do problema, é possível que o tamanho final das arvores seja muito grande, porém, além de serem passíveis de *overfitting*. Podas podem ser aplicadas para reduzir ambos os problemas, mas com uma quantidade tão grande de classes, o tamanho em memória das mesmas ainda pode ser muito grande.

**Random Forests:** dada a extensão do problema, o uso do algoritmo *Random Forests* baseadas em árvores de decisão é uma continuação previsível da abordagem anterior, aplicando-se cortes aleatórios que, selecionados e combinados, levam a

uma descrição fiel do problema. No entanto, essa abertura de possibilidades dada às *Random Forests*, quando aplicada a uma base em que a quantidade de dados utilizados já é limitada pela capacidade computacional, limita-se muito o potencial de tal técnica. A utilização de árvores de decisão tradicionais, quando combinadas como *Random Forests*, é um meio termo de aproximar e agregar as qualidades de ambas as abordagens sem escapar dos limites físicos encontrados durante a execução. Considerando-se que a quantidade de *features* é muito menor do que a de dados, o ganho proporcionado na seleção das *features* também é limitado.

**Support Vector Machines (SVM):** em um primeiro momento, ao transformar dados textuais, o problema ganhou um escopo de "big data" mais interessante. Por esse motivo entendeu-se que seria adequado a aplicação de *SVM*, pelos seguintes motivos: (a) eficiência na abordagem de altas dimensões, e melhor abordagem com maior amostragem; (b) treinamento da função de decisão quando combinado com a exploração *one-versus-all*. No entanto existe um custo: (a) se o número de características é muito maior do que o número de amostras, o método é susceptível a ter maus resultados; (b) necessita-se de mais validações cruzadas que os outros algoritmos utilizados; (c) o custo de execução é alto, devido ao alto processamento e tempo consideravelmente alto também.

**Naive Bayes:** o método de Naive Bayes descreve um conjunto de algoritmos de aprendizado supervisionado com o qual trabalha com as proporcionalidades dos itens abordados no contexto como um todo. Baseado nas *features* em no contexto desse trabalho, torna-se interessante o uso deste algoritmo. Característico de rápida execução, usar Naive Bayes como classificador pode ser extremamente rápido para a obtenção de proporcionalidade perante o problema abordado.

Para melhor exploração de algumas abordagens:

**More features:** a adição de *features* é uma das primeiras abordagens a ser trabalhada, com o intuito de adicionar ou tornar mais claras as informações que relacionam os dados. No entanto, tais técnicas são, frequentemente, baseadas em processos matemáticos (elevação de ordem, gaussianas etc), cuja aplicabilidade em dados textuais e categóricos é pequena. A outra possibilidade é a adição de dados ao se agregar novas informações, baseando-se em informações de especialistas. Enquanto essa possibilidade era real, uma vez que o projeto Open Data possui outras bases referentes a Nova York, a dificuldade em relacionar eventos ou dados distintos adiciona um grande peso no pré-processamento, potencialmente explodindo o número de entradas. Considerando que a quantidade de dados presente já superava o potencial computacional disponível, a adição de novos atributos foi considerada apenas para trabalhos futuros.

**Divide-and-conquer:** como já foi listado, um dos problemas da base de dados utilizada é a cobertura de uma grande extensão espaço-temporal, sendo que certos comportamentos movimentam-se no espaço conforme o tempo. Isso causa não só dificuldades para se aprender um comportamento que ocorre por toda a extensão, mas em momentos diferentes, como também leva a conclusões ou resultados errôneo, especialmente quando se considera valores médios - nesses casos, análises em áreas ou períodos menores apresentam uma descrição

mais fiel do problema e, com isso, levam a modelos com maior acerto e precisão. Uma das estratégias adotadas foi dividir a base em janelas de duas horas, de modo que apenas eventos naquele intervalo de tempo são considerados para o aprendizado. Apesar da informação nos horários limítrofes ser perdida e da necessidade de se treinar um modelo para cada janela, eles não são limitantes no aprendizado, e os resultados obtidos apresentam qualidade consideravelmente maior.

**Feature pruning:** alguns dados da base original não apresentam informações relevantes para o problema, enquanto outros apresentam informações tão infrequentes que o peso adicionado no processamento de um atributo adicional não é compensado pela pouca informação agregada. Outras *features*, como a agência responsável e a data de resolução, foram propositalmente omitidas para permitir que o preditor seja aplicado em casos reais, onde tais dados ainda não estariam disponíveis no momento da abertura do chamado.

**Combination, Aggregation and Boosting:** dada a pluralidade de modelos possíveis para se descrever a base, bem como a grande quantidade de dados - os quais frequentemente ultrapassam a capacidade computacional disponível - é possível e necessário combinar modelos e preditores distintos, tentando-se obter um mais preciso. De fato, essas técnicas garantem uma melhor cobertura da base de dados, utilizando mais informações e agregando mais conhecimento aos algoritmos. Finalmente, agregar diversas técnicas diminui a influência de particularidades de conjuntos específicos de testes, diminuindo o *overfitting* e ajudando na generalização do problema. Essa última característica é especialmente relevante para árvores de decisão, diminuindo assim uma das poucas características negativas de um modelo que se adequa, em vários pontos, ao problema sendo considerado.

## V. OS RESULTADOS

Trabalhando a primeira parte da identificação dos possíveis locais de atendimento, e posteriormente dado a predição da classificação, pode-se explorar qual seria o ponto de atendimento mais preparado e próximo para averiguar o caso de *service request* recebido. Para este tópico, a figura 2 ilustra os possível 250 pontos de atendimentos, mais interessantes para o intervalo entre 12:00 e 14:00 de um dia qualquer da semana.

Alguns exemplos de atendimentos especializados dos *clusters* podem ser é descrito na tabela II (presente no apêndice A), referente ao intervalo entre 12:00 e 14:00.

Perante os algoritmos aplicados e descritos, a tabela I exibe os valores de acurácia disponíveis por intervalo de tempo de 2 horas o algoritmo aplicado (combinado ou não).

Somente no caso da abordagem da técnica de *SVM*, foram utilizados os dados numéricos de Latitude, Longitude, dia da semana (entre 1 para Domingo e 7 para Sábado), hora e minuto. Convertidos usando *PCA* para as novas dimensionalidades e, posteriormente a isso, aplicados alguns conceitos simples de estatísticas, para obtenção dos novos valores utilizando máximo, mínimo, média, curtose e assimetria de uma amostra.

A abordagem mais adequada de acordo com os vários modelos aplicados para a classificação descreve-se em 2 passos:

### 1) Modelo: atributos textuais concatenados e binarizados

Dentre os modelos testados, o que apresentou melhores

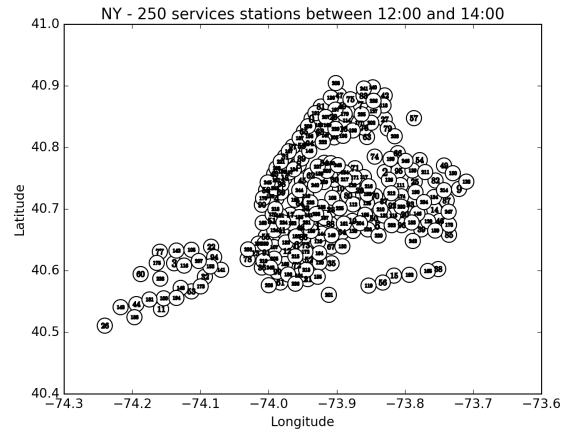


Figura 2: 250 *Clusters* melhores posicionados entre 12:00 e 14:00 de um dia qualquer da semana.

	AdaBoost e Decision Tree	Decision Tree	Random Forest	SVM	Naïve Bayes
0:00 - 2:00	73.61%	53.51%	9.13%	14.12%	5.10%
2:00 - 4:00	36.65%	40.04%	9.79%	0.89%	27.94%
4:00 - 6:00	63.54%	32.10%	6.33%	9.39%	14.63%
6:00 - 8:00	80.78%	48.71%	10.22%	2.69%	9.30%
8:00 - 10:00	87.98%	69.51%	20.50%	2.94%	12.83%
10:00 - 12:00	86.93%	71.41%	19.04%	0.00%	8.46%
12:00 - 14:00	87.44%	70.21%	23.67%	0.29%	9.62%
14:00 - 16:00	87.25%	73.93%	17.33%	1.92%	11.97%
16:00 - 18:00	85.30%	66.86%	13.72%	1.00%	11.12%
18:00 - 20:00	82.43%	60.35%	13.77%	3.66%	12.27%
20:00 - 22:00	83.09%	69.14%	16.52%	2.57%	15.78%

Tabela I: Acurácia obtidas nos modelos

resultados foi uma composição de *AdaBoost* sobre Árvores de Decisão, aplicado sobre entradas formadas pela descrição textual, o endereço e a localização político-geográfica agregados, dividido em janelas de duas horas, totalizando 12 classificadores. Essa aplicação toma os atributos listados e os concatena, gerando um corpus unificado que é, por sua vez, transformado num conjunto de *features* binárias por *tf-idf*. Uma vez que os textos originais eram curtos, a planificação dessas *features* num mesmo texto para cada chamado (e a unificação de todos num mesmo corpus, ao executar *Tf-idf* - *term-frequency inverse document-frequency*) é uma maneira de agregar os atributos, de maneira análoga à utilização de *PCA*, que trabalha com diferentes *features* de origens distintas. Com atributos binarizados, a eficiência das árvores de decisão aumenta, uma vez que não é necessário encontrar os valores limítrofes para variáveis contínuas nem abrir em mais do que dois ramos cada ponto de decisão.

### 2) AdaBoost combinado com Decision Trees

Com os dados descritos por esses conjuntos de variáveis binárias, uma sequência em cascata de árvores de decisão é gerada, sendo que a base de treino de cada árvore em

sucessão é ponderada de maneira a se focar nos casos que foram classificados incorretamente nas anteriores. Dessa maneira, o conjunto de árvores tende progressivamente a acertar os casos mais difíceis, mas preservando os casos encontrados pelas árvores iniciais. Nos nós-folha das árvores, todas as  $x$  possibilidades de classe são abertas, de modo que cada entrada pode ser classificada em qualquer uma das classes possíveis. Apesar da extensão de cada árvore tornar a portabilidade desses classificadores praticamente proibitiva (o conjunto ocupa pouco menos de 1.5 GB em disco), a velocidade de execução para cada caso é praticamente instantâneo (comparado aos outros modelos abordados), de maneira que a disponibilização do conjunto como um serviço é viável e eficiente.

## VI. CONCLUSÕES

Para este trabalho, houve um grande aprendizado em como explorar a combinação dos atributos para novas componentes como:

- i Trabalhar com etapas de processamento, aplicando dois caminhos exploratórios: *Clustering* e a predição das categorias (*Complaint Types*);
- ii Aplicação de varias técnicas a fim de investigar dados numéricos (com redução de dimensionalidade) e categóricos (com de *tf-idf*);
- iii Exploração das técnicas *one-class* (como *Logistic Regression*), com abordagens como *one-versus-all* ou *one-versus-one*.

Pode-se concluir, também, que a disponibilidade de grande quantidade de dados pode não apenas levar a como também exigir modelos mais complexos, seja para descrever corretamente particularidades incomuns, seja para representar comportamentos que, apesar de infrequentes, são parte integrante da solução. A quantidade de classes presentes nesse problema, por exemplo, tornou inviáveis abordagens que, num cenário mais convencional, teriam sido capazes de resolver o problema.

A necessidade de controlar a quantidade de informação considerada, levando em conta as limitações de poder computacional e mesmo de modelagem dos modelos escolhidos, foi outra observação obtida através desse trabalho. Com bases extensas e complexas como essa, é necessário considerar o custo-benefício de cada decisão, tanto em qualidade quanto em tempo ou memória, para se obter um resultado viável em todos os quesitos.

O modelo selecionado como de maior qualidade ainda apresenta uma taxa de erro observável, de pouco mais de 10% para a maior parte das janelas de tempo. Para reduzi-la, seria possível agregar novos dados relativos à origem da ligação, como proximidade de áreas comerciais, fluxo de veículos na região ou uma relação de proximidade entre ruas. Apesar dessas informações existirem, a junção das mesmas aos dados já obtidos é de difícil execução, sendo necessária uma nova etapa de estudos para que as informações agregadas sejam ao mesmo tempo relevantes e fiéis ao sistema modelado.

## APÊNDICE

As figuras 3, 4, 5, 6, 7, 8 descrevem o uso da técnica *Elbow Rule* aplicado a soma da distância total das centróides até seus

pontos pertencentes a cada um dos *clusters* (linha em azul) para identificação do "cotovelo" para o valor  $k$  assim adequado ao considerar intervalos de 2 horas e o tempo de execução (linha vermelha) descrita em segundos.

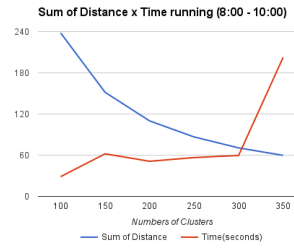


Figura 3

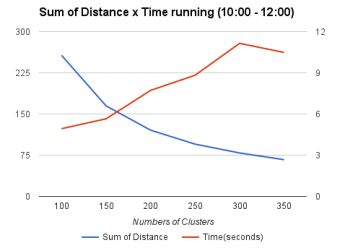


Figura 4

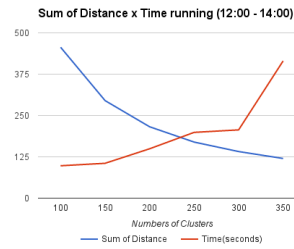


Figura 5

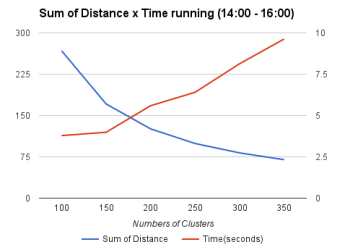


Figura 6

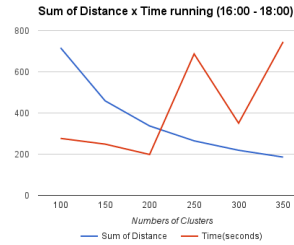


Figura 7

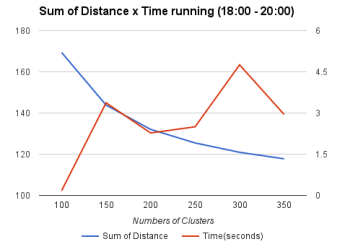


Figura 8

As figuras 9, 10, 11, 12 ilustram a abordagem dos *clusters* aplicados nos intervalos do dia, conforme mencionado anteriormente.

Alguns exemplos de atendimentos especializados dos *clusters* podem ser é descrito na tabela II.

# Cluster	Longitude	Latitude	Atendimentos possíveis		
1	-73.93	40.84	FLOORING/STAIRS	PAINT/PLASTER	DOOR/WINDOW
2	-73.95	40.64	PAINT/PLASTER	UNSANITARY CONDITION	GENERAL
3	-73.95	40.77	Taxi Complaint	DOF Parking - Tax Exemption	Air Quality
4	-73.89	40.73	Root/Sewer/Sidewalk Condition	Derelict Vehicle	Construction
5	-73.81	40.59	Sewer	Water Conservation	Construction
6	-73.87	40.68	Graffiti	Dirty Conditions	Derelict Vehicles

Tabela II: Exemplos de atendimentos especializados por 6 quaisquer *Clusters*, referente ao intervalo entre 12:00 e 14:00

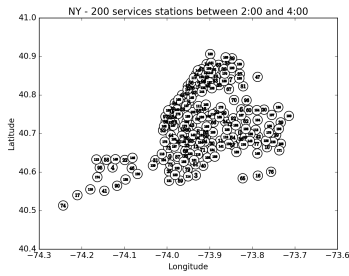


Figura 9

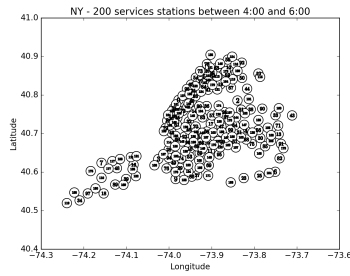


Figura 10

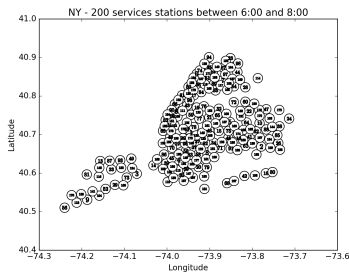


Figura 11

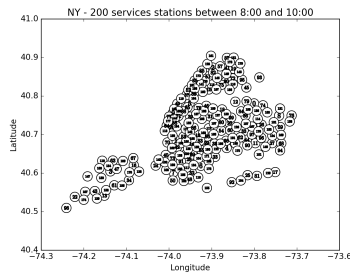


Figura 12