



Fernando Mauro Buleo Barbosa

*Previsão automática com redes neurais da
probabilidade por bairros de serem solicitados
transportes por aplicativos*

Monografia de Final de Curso

31/08/2020

***Monografia apresentada ao Departamento de Engenharia Elétrica da
PUC/Rio como parte dos requisitos para a obtenção do título de
Especialização em Business Intelligence.***

Orientadores:

Profa. Manoela Kohler

Prof. Leonardo Forero Mendoza

Dedicatória

Aos meus filhos, Daniel e Guilherme, minhas maiores alegrias, orgulhos, motivações.

À minha esposa Moira, companheira de jornada.

Agradecimentos

Ao Bruno Muniz da empresa Gaudium, por compartilhar sua experiência, necessidades de negócio e a base de dados para realização deste trabalho.

Aos professores Manoela Kohler e Leonardo Forero, pelas aulas e pela orientação desse TCC.

À minha família pela compreensão e investimento conjunto em tempo, nos sábados, domingos e noites de estudo.

O melhor momento para plantar uma árvore foi há 20 anos atrás.
O segundo melhor é agora.

Provérbio Zen

RESUMO

O modelo de negócio dos transportes por aplicativos é relativamente recente e viabilizado essencialmente pelo uso da tecnologia. O aplicativo em si permite que o usuário solicite transporte identificando transportadores disponíveis localizados próximo ao local de partida. Integrado ao aplicativo, são agregadas diversas soluções, como aplicativos de navegação, capazes, não só de identificar a melhor rota a seguir, mas também de viabilizar que transportadores atuem na área mesmo sem saber como chegar por conta própria a origem e ao destino do transporte.

No presente trabalho foi desenvolvido modelo preditivo visando identificar as áreas de cidades com maior probabilidade de demandarem transporte por aplicativo, a cada dia e hora. O modelo preditivo foi desenvolvido em uma rede neural e treinado para prever a probabilidade de transportes serem demandados em 3 (três) cidades brasileiras. Para o treinamento foi utilizada a base completa de 1 (um) mês de 2020 de transportes realizados por aplicativo.

Os melhores resultados obtidos indicaram que, em 84% dos casos, o modelo consegue prever a probabilidade de bairros demandarem transporte com uma distância de até 3 posições em relação a ordem ("*ranking*") real de bairros, quando consideramos os 5 que mais demandam transportes. Expandindo a análise para os 10 bairros que mais solicitam transportes, foi verificado que, em 77% dos casos, o modelo consegue prever a probabilidade de bairros demandarem transportes com uma distância de até 3 posições em relação ao "*ranking*" real.

Essa é, portanto, mais uma solução tecnológica que pode ser agregada ao negócio de modo a facilitar o trabalho dos transportadores, reduzir custos e elevar a satisfação de clientes.

ABSTRACT

The business model of transport by applications is relatively recent and made possible essentially by the use of technology. The application itself allows the user to request transportation by identifying available carriers located close to the party location. Integrated into the application, various solutions are added, such as navigation applications, capable of not only identifying the best route to follow, but also enabling carriers to operate in the area without even knowing how to reach the customer's origin and destination on their own.

In this work, a predictive model was developed to identify the areas of cities most likely to require transportation, every day and hour. The predictive model was developed in a neural network and trained to predict the likelihood of a demand for transport in 3 (three) Brazilian cities. For the training of the model, the complete base of transports performed in 1 (one) month of the year 2020 of a transport application was used.

The best results obtained indicated that, in 84% of the cases, the model is able to predict the probability of neighborhoods requiring transport with a distance of up to 3 positions in relation to the real "*ranking*", when we consider the 5 neighborhoods that most originate races. Expanding the analysis to the 10 neighborhoods that demand the most transport, It was found that, in 77% of the cases, the model is able to predict the probability of neighborhoods requiring transport with a distance of up to 3 positions in relation to the real "*ranking*".

This is, therefore, another technological solution that can be added to the business in order to facilitate the work of the transporters, reduce costs and increase customer satisfaction.

Sumário

1. INTRODUÇÃO	10
1.1. MOTIVAÇÃO	10
1.2. OBJETIVOS DO TRABALHO	10
1.3. DESCRIÇÃO DO TRABALHO	11
2. DESCRIÇÃO DO PROBLEMA	12
3. ARQUITETURA DO SISTEMA PROPOSTO	13
3.2. Pré-processamento	16
3.2.1. <i>Seleção das Variáveis</i>	16
3.2.2. <i>Formação da Variável de Resposta</i>	17
3.2.3. <i>Padronização de Informações</i>	21
3.2.4. <i>Base de Informações de Entrada do Modelo de Inferência</i>	22
3.2.5. <i>Seleção de Instâncias</i>	23
3.3. Inferência	26
3.3.1. <i>Carga da Base de Informações e Seleção da Cidade de Partida</i>	27
3.3.2. <i>Conversão para Dummy Variables</i>	27
3.3.3. <i>Separação em Base de Treino e Base de Testes</i>	27
3.3.4. <i>Criação da Rede Neural</i>	27
4. RESULTADOS	30
4.1. <i>Simulações</i>	30
4.2. <i>Melhores Resultados</i>	32
4.2.1. <i>Cidade: Patos de Minas</i>	32
4.2.1.1. <i>Otimizador Nadam – Todas as horas do dia</i>	32
4.2.1.2. <i>Otimizador Nadam – Horário de 8 a 19 horas</i>	33
4.2.1.3. <i>Otimizador SGD – Todas as horas do dia</i>	34
4.2.1.4. <i>Otimizador SGD – Horário de 8 a 19 horas</i>	35
4.2.2. <i>Cidade: Fortaleza</i>	36
4.2.2.1. <i>Otimizador Nadam – Todas as horas do dia</i>	36
4.2.2.2. <i>Otimizador Nadam – Horário de 8 a 19 horas</i>	37
4.2.2.3. <i>Otimizador SGD – Todas as horas do dia</i>	38
4.2.2.4. <i>Otimizador SGD – Horário de 8 a 19 horas</i>	39
4.2.3. <i>Cidade: Petrolina</i>	40
4.2.3.1. <i>Otimizador Nadam – Todas as horas do dia</i>	40
4.2.3.2. <i>Otimizador Nadam – Horário de 8 a 19 horas</i>	41
4.3. <i>Análise da Ordem (“Ranking”) de Bairros com Maior Probabilidade de Demandar Transporte</i>	42
4.3.1. <i>Cidade: Fortaleza</i>	44
4.3.2. <i>Cidade: Patos de Minas</i>	46
4.3.3. <i>Cidade: Petrolina</i>	47
5. CONCLUSÕES E TRABALHOS FUTUROS	50
Referências Bibliográficas	52
Apêndice A – Código fonte do programa de inferência	53
Apêndice B – Relação completa das simulações realizadas	62

Lista de Tabelas

Tabela 1- Relação de Campos da Base de Transportes Demandados em Janeiro/2020 e Variáveis Predictoras Eleitas	16
Tabela 2- Amostra da Base de Informações com Probabilidade Calculada para Input no Modelo de Inferência	19
Tabela 3- Amostra de Padronização da Acentuação no Campo Bairro de Partida	21
Tabela 4- Amostra da Base de Informações com Probabilidade Calculada para Input no Modelo de Inferência	23
Tabela 5- Instâncias (Cidades) selecionadas para Modelagem e Indicadores Utilizados na Seleção	23
Tabela 6- Distribuição do Total de Linhas da Base de Informações por Faixa de Probabilidades nas Cidades Eleitas	30
Tabela 7- Transportes Realizados por Hora de Partida para as Cidades Eleitas.....	31
Tabela 8- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador Nadam	33
Tabela 9- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador Nadam (Horário "Comercial")	34
Tabela 10- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador SGD.....	34
Tabela 11- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador SGD (Horário "Comercial").....	35
Tabela 12- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador Nadam	36
Tabela 13- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador Nadam (Horário "Comercial")	37
Tabela 14- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador SGD	38
Tabela 15- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador SGD (Horário "Comercial").....	39
Tabela 16- Parâmetros e Resultados da Melhor Simulação de Petrolina com Otimizador Nadam	41
Tabela 17- Parâmetros e Resultados da Melhor Simulação de Petrolina com Otimizador Nadam (Horário "Comercial")	42
Tabela 18- Ordem ("Ranking") dos Bairros com maior Probabilidade (real) de Solicitarem Transporte em Fortaleza, em 15/Janeiro às 7:00hs	43
Tabela 19- Ordem ("Ranking") dos Bairros com maior Probabilidade (estimada) de Solicitarem Transporte em Fortaleza, em 15/Janeiro às 7:00hs.....	43
Tabela 20- Distância entre Ranking Real e Estimado dos Bairros com maior Probabilidade de Solicitarem Transporte em Fortaleza, em 15/Janeiro às 7:00hs.....	44
Tabela 21- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 10 Primeiros Bairros do Ranking	44
Tabela 22- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 5 Primeiros Bairros do Ranking	44
Tabela 23- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 20 Primeiros Bairros do Ranking	45
Tabela 24- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 10 Primeiros Bairros do Ranking (Horário "Comercial")	45
Tabela 25- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 5 Primeiros Bairros do Ranking (Horário "Comercial")	45

Tabela 26- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 20 Primeiros Bairros do Ranking (Horário "Comercial")	46
Tabela 27- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 10 Primeiros Bairros do Ranking	46
Tabela 28- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 5 Primeiros Bairros do Ranking	46
Tabela 29- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 10 Primeiros Bairros do Ranking (Horário "Comercial")	47
Tabela 30- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 5 Primeiros Bairros do Ranking (Horário "Comercial")	47
Tabela 31- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 10 Primeiros Bairros do Ranking	47
Tabela 32- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 5 Primeiros Bairros do Ranking	48
Tabela 33- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 10 Primeiros Bairros do Ranking (Horário "Comercial")	48
Tabela 34- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 5 Primeiros Bairros do Ranking (Horário "Comercial")	49

Lista de Figuras

Figura 1- Esquema Básico de Mineração de Dados.....	13
Figura 2- Quantidade de Transportes ("Corridas") por Data	14
Figura 3- ETL Desenvolvido.....	14
Figura 4- Operador CSV File Input no Pentaho.....	15
Figura 5- Operador Calculator no Pentaho	15
Figura 6- Operador Table Output no Pentaho	16
Figura 7- Estrutura da Tabela "corridas_sumário" no banco de dados PostGres.....	18
Figura 8- Total de Transportes Originados em Fortaleza ao longo de Janeiro/2020	25
Figura 9- Total de Transportes Originados em Petrolina ao longo de Janeiro/2020.....	26
Figura 10- Total de Transportes Originados em Patos de Minas ao longo de Janeiro/2020...	26
Figura 11- Overfit em rede neural de 3 camadas com 160, 80 e 1 neurônios e Otimizador Adam	31
Figura 12- Distribuição de Transportes Realizados por Faixa Horária.....	32
Figura 13- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Patos de Minas	33
Figura 14- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Patos de Minas Horário "Comercial"	34
Figura 15- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Patos de Minas	35
Figura 16- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Patos de Minas Horário "Comercial"	36
Figura 17- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Fortaleza	37
Figura 18- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Fortaleza Horário "Comercial"	38
Figura 19- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Fortaleza	39
Figura 20- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Fortaleza Horário "Comercial"	40
Figura 21- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Petrolina	41
Figura 22- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Petrolina Horário "Comercial"	42

1. INTRODUÇÃO

1.1. MOTIVAÇÃO

Os volumes na economia movimentados pelos aplicativos de transporte no Brasil e no Mundo evidenciam um mercado em expansão. Segundo pesquisa da *Mobile Time*, 71% dos usuários de *smartphone* no Brasil já haviam solicitado um transporte por aplicativo em 2019. Isso representa um aumento de 21 pontos percentuais em apenas dois anos (em 2017 o total era de 50%) (Oliveira, 2020). Já outra pesquisa realizada pela Toluna mostrou que os brasileiros gastam, em média, R\$300 por mês com aplicativos de transporte, conforme matéria divulgada em Fev/2020. Isso corresponde a um gasto médio de 10% do salário em aplicativos de transporte. (Oliveira, 2020).

Quanto ao número de motoristas que trabalham por conta própria por aplicativos, a Pesquisa Nacional de Empregados e Desempregados (Pnad) Contínua Trimestral mostra um aumento de 137,60% no período de 2012 e 2019 (Cardim, 2020). Estima-se que existiam em 2019, no Brasil, cerca de 1,1 milhão de motoristas de aplicativos. A expectativa do governo brasileiro é que mais da metade deles tornem-se microempreendedores. (Cruz, 2019)

Dentre as empresas de aplicativos que atuam no setor, a Uber segue como a preferida de 80% dos usuários de transporte por aplicativo. Em 2018, o Brasil já era o segundo maior mercado da empresa no mundo, correspondendo a um faturamento de mais de 1 bilhão de reais. Outras empresas tiveram também, grande crescimento nesse mercado. A espanhola Cabify chegou ao país em 2016 como a principal concorrente da Uber e em 2017 viu seu faturamento ser multiplicado em 20 vezes comparado ao ano anterior, passando a contar com mais de 200 mil motoristas cadastrados no país e presença em seis capitais (Oliveira, 2020). Fruto de uma *startup* brasileira fundada em 2012, o aplicativo de transporte 99 tornou-se oficialmente o primeiro “unicórnio” brasileiro, ao ser adquirida pela chinesa Didi Chuxing, em 2018, por mais de US\$ 1 bilhão (Manzoni, 2020).

O modelo de negócio dos transportes por aplicativos é, portanto, relativamente recente e viabilizado essencialmente pelo uso da tecnologia. A partir desse entendimento, surgiu a motivação para pesquisar trabalhos realizados nessa área e para desenvolver um Trabalho de Conclusão de Curso que buscasse atender à necessidade prática de empresas do setor.

Na busca por um tema para o Trabalho de Conclusão de Curso, tive a oportunidade de conhecer líderes da Gaudium¹, empresa que desenvolveu e comercializa o aplicativo *Machine*². A *Machine* oferece soluções para organização de centrais de Taxi (*Taxi Machine*), organização de centrais de Moto Taxi (*Moto Machine*), gerenciamento de frota de entregas (*apps* tipo Loggi) e o *Driver Machine*, aplicativo com funcionalidades semelhantes ao Uber que oferece uma alternativa para os modelos dominantes e foco em cidades de médio e pequeno porte. Em reunião com um de seus diretores foi reportado o interesse em dispor de um modelo preditivo que indicasse as áreas onde havia maior probabilidade de serem solicitados transportes ao longo do dia. Esse se tornou o objetivo desse trabalho.

1.2. OBJETIVOS DO TRABALHO

Para realização desse trabalho foi disponibilizada, pela *Machine*, uma base completa de transportes realizados em janeiro de 2020. Essa base dispõe para cada transporte, de informações como a data da “corrida” (transporte), horário, cidade de partida, cidade de destino, bairro de partida e bairro de destino. De posse dessa base os objetivos do trabalho foram:

- Gerar base de informações, a partir da base disponível de transportes realizados, com a probabilidade de cada bairro, em cada cidade de partida, demandar transportes a cada dia e hora do mês de modo a servir como base de treino e base de testes do modelo preditivo a desenvolver

¹ <https://gaudium.global/>

² <https://machine.global/>

- Eleger, dentre as 1.118 cidades de partida presentes na base, quais seriam usadas nesse estudo
- Desenvolver Rede Neural para prever, por método de regressão, a probabilidade de cada bairro das cidades eleitas no estudo, demandar transportes a cada dia, de hora em hora

1.3. **DESCRIÇÃO DO TRABALHO**

O desenvolvimento dessa monografia envolveu:

- O estudo de ferramentas de *ETL*
- O estudo de técnicas de *Data Mining*
- O estudo de modelos de *Machine Learning* e redes neurais para previsão por regressão
- O desenvolvimento de *ETL* para extração e transformação da base de dados do cliente em uma base de informações das probabilidades dos bairros originarem transportes
- Desenvolvimento de Rede Neural para previsão por regressão da probabilidade de bairros originarem transportes
- Execução de simulações para parametrização da Rede Neural
- Avaliação dos Resultados

O Capítulo 2 apresenta uma breve descrição da oportunidade de estudo de uma solução para esse desafio de mercado.

No Capítulo 3, a seção 3.1 trata essencialmente da solução de *ETL* e sua aplicação. Já a seção 3.2 trata da aplicação de técnicas de data mining, bem como do desenvolvimento do modelo preditivo por rede neural.

O Capítulo 4 detalha as simulações realizadas para determinar os parâmetros ótimos a serem aplicados no modelo preditivo. Trata, também, dos resultados obtidos e da análise dos mesmos.

Por fim, o Capítulo 5 apresenta as conclusões deste trabalho.

2. DESCRIÇÃO DO PROBLEMA

O uso da tecnologia e de métodos “inteligentes” tem sido fundamental para viabilizar o negócio de transporte por aplicativos, permitindo, inclusive, que motoristas sem experiência prévia em transportes atuem na área. Aplicativos de navegação permitem que um motorista busque um passageiro e o leve até o destino sem nunca ter ido antes aos endereços de partida e de destino. Mesmo motoristas experientes no ramo de transporte e conhecedores dos mapas e ruas da cidade onde atuam, podem se beneficiar dessa inteligência usando a rota mais rápida apontada pelo aplicativo naquele momento, em vez de se basearem unicamente em sua experiência prévia, insuficiente para prever situações excepcionais como congestionamentos do tráfego, sinistros nas vias etc.

Nesse sentido, motoristas buscam circular nas áreas que, segundo a experiência adquirida, tem mais probabilidade de demandar transportes, a fim de maximizar sua receita e evitar ficarem circulando pela cidade a esmo consumindo combustível.

O modelo proposto nesse Trabalho de Conclusão de Curso se destina a estimar, para um conjunto de cidades eleitas, a probabilidade de serem demandados transportes em cada bairro dessas cidades, a cada período de 1 (uma) hora do dia.

Esse modelo viabiliza que qualquer motorista tenha conhecimento das áreas da cidade com maior probabilidade de originar transportes naquele dia e horário, independente de experiência prévia. Dessa forma, ele poderá se dirigir para uma das áreas apontadas por essa solução, que se situe mais próxima de sua localização atual.

Alguns dos benefícios dessa solução são:

- Maximizar a receita dos transportadores
- Minimizar o custo dos transportadores, que podem despriorizar a circulação em áreas com menor probabilidade de demandar transportes, reduzindo o gasto de combustível sem receita associada
- Maximizar o total de clientes satisfeitos, reduzindo o tempo de espera por um transportador em função da maior disponibilidade de transportadores nas áreas que mais geram transportes

3. ARQUITETURA DO SISTEMA PROPOSTO

A arquitetura do sistema proposto nesse trabalho segue o esquema básico de mineração de dados e está ilustrado na Figura 1- Esquema Básico de Mineração de Dados (Koshiyama, 2015). As etapas desse esquema são detalhadas nas seções a seguir.

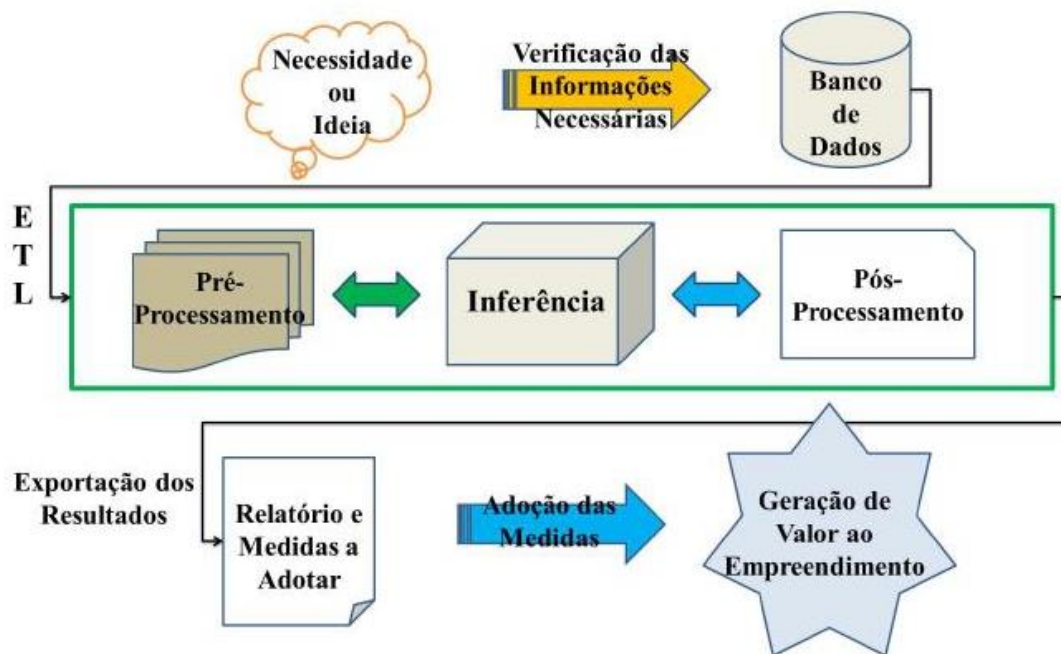


Figura 1- Esquema Básico de Mineração de Dados

3.1. Etapa Macro 1 – Pré DM

Engloba as etapas de “Necessidade ou Idéia” até a obtenção de uma Base de Dados (“Banco de Dados”).

Como já descrito no capítulo 2 desse documento, a “Necessidade ou Idéia” a ser suprida por essa solução é, dada uma base histórica de transportes demandados a partir de aplicativos de transporte, prever a probabilidade dos bairros das cidades atendidas pelo aplicativo, solicitarem transporte. A partir dessa informação seria possível estabelecer uma ordem (“*ranking*”) com os bairros que geram mais demanda de transporte a cada hora, de modo a direcionar os motoristas previamente para essas áreas.

Quanto a “Verificação das Informações Necessárias” e “Banco de Dados”, foi disponibilizada uma base de 2.634.032 registros, em arquivo de formato “csv”, referentes aos transportes registrados no aplicativo *Machine* ao longo do mês de janeiro de 2020. A base de transportes realizados apresentava os seguintes campos:

- ID – Identificador do registro (chave primária)
- Data_Hora_Criação – Data e Hora de início do transporte em formato Timestamp (yyyy-MM-dd HH:mm:ss)
- lat_partida – Latitude de onde o transporte foi iniciado
- lng_partida – Longitude de onde o transporte foi iniciado
- nome_cidade_partida – Nome da cidade onde o transporte foi iniciado
- bairro_partida – Bairro onde o transporte foi iniciado
- lat_destino – Latitude de destino do transporte
- lng_destino – Longitude de destino do transporte
- nome_cidade_destino – Nome da cidade de destino do transporte

- bairro_destino – Bairro de destino do transporte

Quanto ao desenvolvimento de uma solução de *ETL* (*Extract, Transform and Load*), os principais motivadores foram:

1. É contraproducente trabalhar com esse volume de registros em Excel, justificando a realização da extração (*Extract*) das informações do arquivo de origem e carga (*Load*) em banco de dados.
2. Como queremos que o modelo preveja a probabilidade de um bairro solicitar transporte de hora em hora, precisamos dispor de um campo adicional calculado a partir do campo de *Data_Hora_Criação* (*transform*) com a hora de início do transporte. Por exemplo, para um transporte iniciado às 17:33:30, precisamos dispor de um campo com a hora preenchida com 17.
3. A informação de dia do mês também foi incluída em campo separado extraído a partir do campo *Data_Hora_Criação*, para ser utilizada como uma das variáveis preditoras do modelo
4. Adicionalmente, a análise dos dados da base mostrou, posteriormente, uma sazonalidade em função do dia da semana, como podemos ver na Figura 2, a qual apresenta o total de transportes (“corridas”) da base por data do transporte. Exemplificando, os domingos tendem a apresentar um volume de transportes menor que o restante dos dias da semana. Já as 6as-feiras e sábados apresentam os maiores volumes de transportes. Assim sendo, foi entendido que seria interessante dispor, também, de um campo calculado indicando o dia da semana.

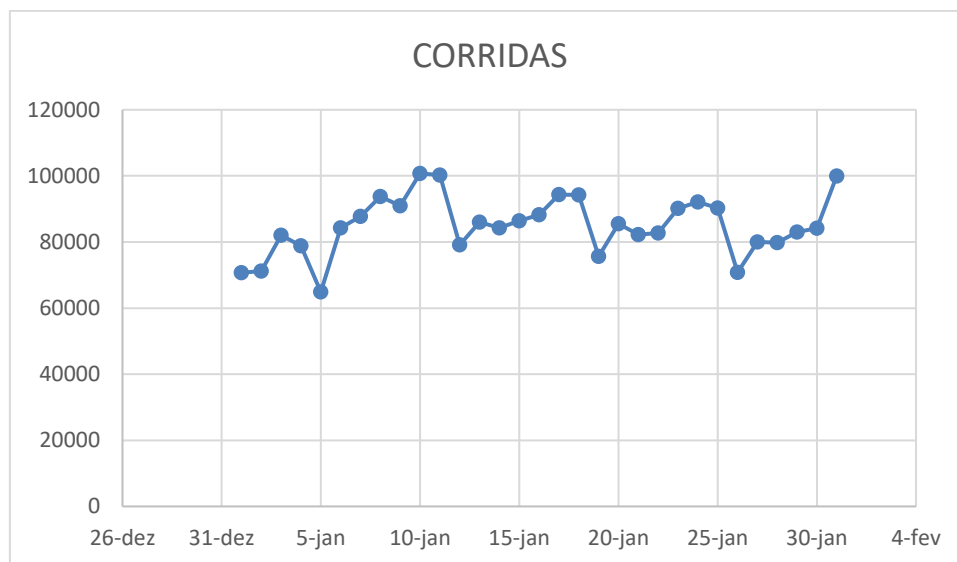


Figura 2- Quantidade de Transportes ("Corridas") por Data

Foi desenvolvido, portanto, o *ETL* ilustrado na Figura 3 utilizando o *software Pentaho*:

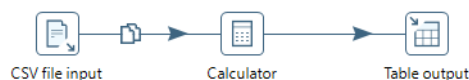


Figura 3- ETL Desenvolvido

O operador “*CSV file input*” lê o arquivo em formato “*CSV*” disponibilizado com os transportes realizados em janeiro de 2020, extraindo as informações de cada campo de cada registro do arquivo de acordo com a formatação especificada (vide Figura 4).

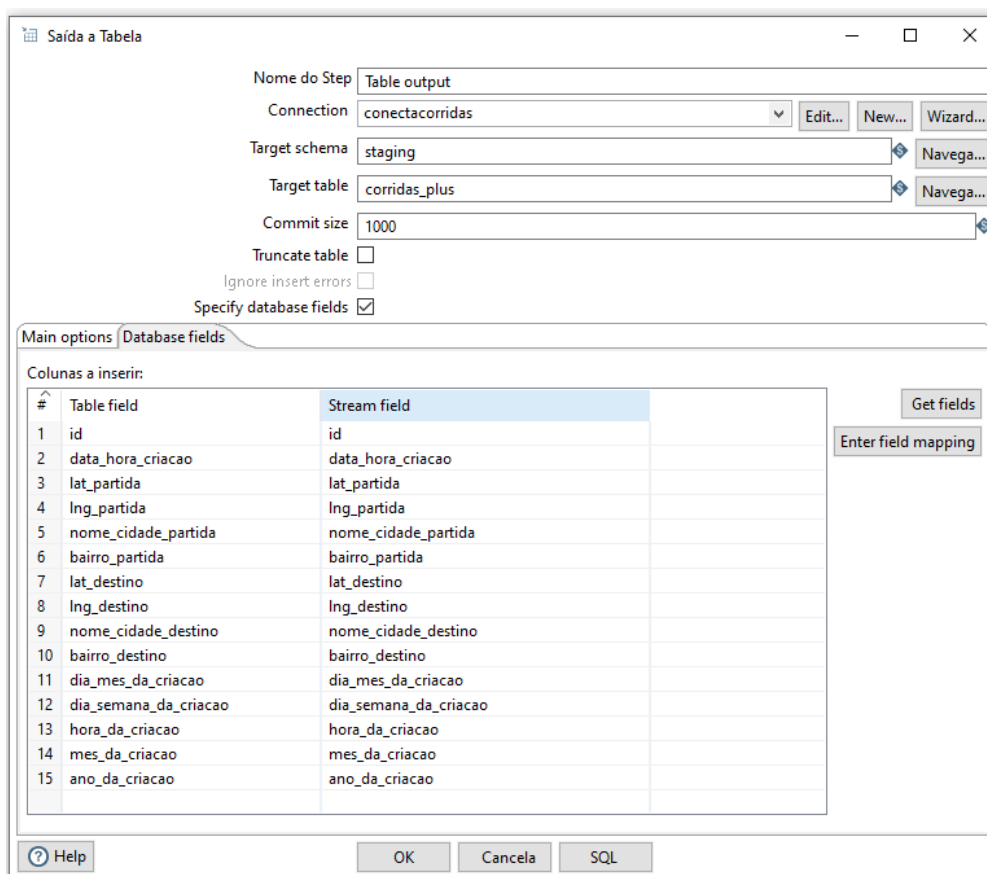


Figura 6- Operador Table Output no Pentaho

3.2. Pré-processamento

3.2.1. Seleção das Variáveis

Consistiu na seleção dos campos e informações da base de dados que serviram como variáveis preditoras da resposta que se deseja que o modelo preveja. A Tabela 1 apresenta a relação de variáveis carregadas no Banco de Dados via processo de ETL, a indicação de quais foram eleitas como preditoras e a justificativa:

Tabela 1- Relação de Campos da Base de Transportes Demandados em Janeiro/2020 e Variáveis Preditoras Eleitas

Campo	Preditora	Justificativa
id	Não	Funciona como identificador (chave) do registro somente
data_hora_criacao	Não	Base dispõe de campos de dia, mês, hora, ano, dia da semana da criação calculados a partir desse campo
lat_partida	Não	Base dispõe da informação de Bairro de Partida calculado a partir desse campo. Latitude tem uma granularidade muito baixa dado que o modelo se dispõe a prever probabilidade a partir do bairro
lng_partida	Não	Base dispõe da informação de Bairro de Partida calculado a partir desse campo. Longitude tem uma granularidade muito baixa dado que o modelo se dispõe a prever probabilidade a partir do bairro

Campo	Preditora	Justificativa
nome_cidade_partida	Sim	A previsão a ser feita se refere a probabilidade de um bairro de uma determinada cidade demandar transportes. Por isso, foi considerada como preditora, embora tenham sido desenvolvidos modelos diferentes para cada cidade de partida.
bairro_partida	Sim	A previsão a ser feita se refere a probabilidade de um bairro de uma determinada cidade demandar transporte. Cabe, portanto, incluir como preditora.
lat_destino	Não	Base dispõe da informação de Bairro de Destino calculado a partir desse campo. Latitude tem uma granularidade muito baixa
lng_destino	Não	Base dispõe da informação de Bairro de Destino calculado a partir desse campo. Longitude tem uma granularidade muito baixa
nome_cidade_destino	Não	O modelo proposto se dispõe a prever a probabilidade de transportes serem originados em um bairro, independente do destino dos mesmos
bairro_destino	Não	O modelo proposto se dispõe a prever a probabilidade de transportes serem originados em um bairro, independente do destino dos mesmos
dia_mes_da_criacao	Sim	Dia do mês em que o transporte iniciou. Mantida como preditora pois os transportes iniciados em um dia podem contribuir para prever outros transportes que poderão vir a ser demandados em outros horários naquele mesmo dia. Entendo que essa informação seria mais relevante se a base de dados dispusesse de informações de vários meses.
dia_semana_da_criacao	Sim	Conforme verificado, o volume de transportes apresenta sazonalidade em função do dia da semana , por isso foi considerada como variável preditora.
hora_da_criacao	Sim	O modelo proposto se dispõe a prever a probabilidade de transportes serem originadas em um bairro a cada hora , por isso foi considerada como preditora.
mes_da_criacao	Não	Dado que a base de dados somente apresenta um mês de informações, ou seja, todos os registros da base se referem ao mês 1, não foi considerada como preditora. Trabalhos futuros com uma base dispondo de informações de vários meses ou vários anos deveriam levar em consideração essa variável como preditora.
ano_da_criacao	Não	Dado que a base de dados só apresenta um mês de informações, ou seja, todos os registros da base se referem ao mês 1 e ano 2020, não foi considerada como preditora.

3.2.2. Formação da Variável de Resposta

A probabilidade de um bairro originar transportes corresponde a variável de resposta do modelo, ou seja, a variável que o modelo desenvolvido nesse trabalho se dispõe a prever. Como pode ser observado na Tabela 1, essa informação não está presente na base de dados disponibilizada. Essa

variável foi calculada resumando informações da base carregada no banco de dados Postgres e aplicando a seguinte fórmula de cálculo:

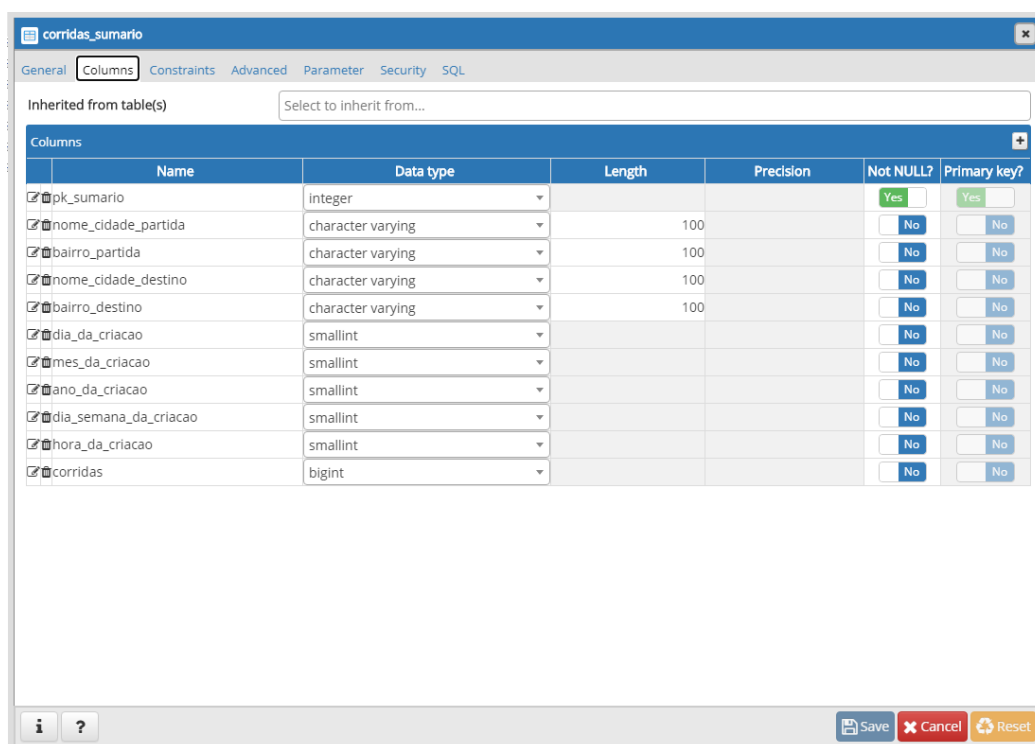
$$Probabilidade\ do\ Bairro\ (d, h) = \frac{Total\ de\ Transportes\ originados\ no\ Bairro\ (d, h)}{Total\ de\ Transportes\ originados\ na\ Cidade\ (d, h)}$$

onde :

$d \rightarrow$ corresponde a um dia específico

$h \rightarrow$ corresponde ao período de 1 hora “cheia” nesse dia “d”

Inicialmente foi criada uma nova tabela “corridas_sumario” no banco de dados adicionando um campo totalizador “Corridas” o qual contava a quantidade de transportes que apresentavam as mesmas características das demais colunas da tabela. A estrutura da tabela “corridas_sumário” é apresentado na Figura 7.



The screenshot shows the 'Columns' tab of the 'corridas_sumario' table in a database management tool. The table has the following columns:

Name	Data type	Length	Precision	Not NULL?	Primary key?
pk_sumario	integer			Yes	Yes
nome_cidade_partida	character varying	100		No	No
bairro_partida	character varying	100		No	No
nome_cidade_destino	character varying	100		No	No
bairro_destino	character varying	100		No	No
dia_da_criacao	smallint			No	No
mes_da_criacao	smallint			No	No
ano_da_criacao	smallint			No	No
dia_semana_da_criacao	smallint			No	No
hora_da_criacao	smallint			No	No
corridas	bigint			No	No

Figura 7- Estrutura da Tabela "corridas_sumário" no banco de dados PostGres

Essa tabela foi populada a partir das informações da tabela “corridas_plus” carregada conforme descrito no item 3.1 desse relatório. Utilizei o seguinte script SQL para popular a tabela “corridas_sumario”:

```
INSERT INTO staging.corridas_sumario(
    nome_cidade_partida, bairro_partida, nome_cidade_destino, bairro_destino,
    dia_da_criacao, mes_da_criacao, ano_da_criacao, dia_semana_da_criacao,
    hora_da_criacao, corridas)
select nome_cidade_partida, bairro_partida, nome_cidade_destino,
bairro_destino,
dia_mes_da_criacao, mes_da_criacao, ano_da_criacao, dia_semana_da_criacao,
hora_da_criacao,
count(*) as corridas from staging.corridas_plus
group by 1,2,3,4,5,6,7,8,9
```

Para gerar as informações do Numerador da fórmula de probabilidade, ou seja, “Total de Transportes originados no Bairro (d, h)” apliquei o seguinte script SQL:

```
select nome_cidade_partida, dia_da_criacao, dia_semana_da_criacao,
hora_da_criacao, bairro_partida, sum(corridas)
from staging.corridas_sumario
group by 1,2,3,4,5
```

Para gerar as informações do Denominador da fórmula de probabilidade, ou seja, “Total de Transportes originados na Cidade (d, h)” apliquei o seguinte script SQL:

```
select nome_cidade_partida, dia_da_criacao, dia_semana_da_criacao,
hora_da_criacao, sum(corridas)
from staging.corridas_sumario
group by 1,2,3, 4
```

Os resultados desses scripts foram introduzidos em Planilha Excell, reproduzida parcialmente na Tabela 2.

Tabela 2- Amostra da Base de Informações com Probabilidade Calculada para Input no Modelo de Inferência

id	cidade	dia	Dia semana	hora	bairro_partida	Probabilidade	Corridas	TotalPorCidade
1	Fortaleza	31	6	23	Praia de Iracema	0,015873016	1	63
2	Fortaleza	31	6	23	Jose Bonifacio	0,015873016	1	63
3	Fortaleza	31	6	23	Mondubim	0,047619048	3	63
4	Fortaleza	31	6	23	Passare	0,047619048	3	63
5	Fortaleza	31	6	23	Alvaro Weyne	0,015873016	1	63
6	Fortaleza	31	6	23	Varjota	0,015873016	1	63
7	Fortaleza	31	6	23	Coco	0,015873016	1	63
8	Fortaleza	31	6	23	Lagoa Redonda	0,031746032	2	63
9	Fortaleza	31	6	23	Edson Queiroz	0,095238095	6	63
10	Fortaleza	31	6	23	Lagoa Sapiranga (Coite)	0,031746032	2	63
11	Fortaleza	31	6	23	Dionisio Torres	0,015873016	1	63
12	Fortaleza	31	6	23	Meireles	0,111111111	7	63
13	Fortaleza	31	6	23	Presidente Kennedy	0,047619048	3	63
14	Fortaleza	31	6	23	Planalto Ayrton Senna	0,031746032	2	63
15	Fortaleza	31	6	23	Messejana	0,015873016	1	63
16	Fortaleza	31	6	23	Engenheiro Luciano Cavalcante	0,015873016	1	63
17	Fortaleza	31	6	23	Centro	0,031746032	2	63
18	Fortaleza	31	6	23	Parque Manibura	0,031746032	2	63
19	Fortaleza	31	6	23	Varjota	0,015873016	1	63
20	Fortaleza	31	6	23	Granja Lisboa	0,015873016	1	63
21	Fortaleza	31	6	23	Bom Jardim	0,015873016	1	63

id	cidade	dia	Dia semana	hora	bairro_partida	Probabilidade	Corridas	TotalPorCidade
22	Fortaleza	31	6	23	Cidade dos Funcionarios	0,015873016	1	63
23	Fortaleza	31	6	23	Carlito Pamplona	0,015873016	1	63
24	Fortaleza	31	6	23	Vila Velha	0,015873016	1	63
25	Fortaleza	31	6	23	Vila Uniao	0,015873016	1	63
26	Fortaleza	31	6	23	Aldeota	0,126984127	8	63
27	Fortaleza	31	6	23	Conjunto Prefeito Jose Walter	0,126984127	8	63
28	Fortaleza	31	6	22	Conjunto Prefeito Jose Walter	0,097222222	7	72
29	Fortaleza	31	6	22	Papicu	0,027777778	2	72
30	Fortaleza	31	6	22	Farias Brito	0,013888889	1	72
31	Fortaleza	31	6	22	Parque Manibura	0,013888889	1	72
32	Fortaleza	31	6	22	Bairro de Fatima	0,027777778	2	72
33	Fortaleza	31	6	22	Centro	0,041666667	3	72
34	Fortaleza	31	6	22	Curio	0,013888889	1	72
35	Fortaleza	31	6	22	Parquelandia	0,041666667	3	72
36	Fortaleza	31	6	22	Antonio Bezerra	0,013888889	1	72
37	Fortaleza	31	6	22	Varjota	0,013888889	1	72
38	Fortaleza	31	6	22	Bonsucesso	0,013888889	1	72

Segue, abaixo, o significado de cada coluna da base de informações apresentada na Tabela 2:

Id	→ Identificador de cada linha da tabela
Cidade	→ Cidade de partida do transporte
Dia	→ Dia do mês de início do transporte
Dia Semana	→ Dia da semana (Domingo, 2ª, 3ª, 4ª, 5ª, 6ª, sábado) de início do transporte
Hora	→ Hora cheia do início do transporte (transporte iniciado às 17:42hs tem o campo preenchido com 17)
Bairro Partida	→ Bairro de partida do transporte
Probabilidade	→ Campo resultante da divisão do valor do campo “Corridas” pelo valor do campo “TotalPorCidade”. Corresponde a probabilidade do bairro indicado no campo “Bairro Partida” originar transportes no dia indicado no campo “Dia” e na Hora indicada no campo “Hora”
Corridas	→ Corresponde ao total de transportes originados no bairro indicado no campo “Bairro Partida”, no dia indicado no campo “Dia” e na Hora indicada no campo “Hora”. (numerador “ <i>Total de Transportes originados no Bairro (d, h)</i> ”)
TotalPorCidade	→ Corresponde ao total de transportes originados na cidade indicado no campo “Cidade”, no dia indicado no campo “Dia” e na Hora indicada no campo “Hora”. (denominador “ <i>Total de Transportes originados na Cidade (d, h)</i> ”)

3.2.3. Padronização de Informações

O nome de algumas cidades e bairros vieram acentuados conforme grafia portuguesa na base original. Quando carregados em banco de dados via processo de *ETL*, alguns desses caracteres acentuados foram convertidos em caracteres especiais.

Isso gerou problemas de padronização e erros no processamento do programa *Python* desenvolvido para inferência do modelo. Assim, optei por padronizar a grafia dos nomes de cidades e bairros sem acentuação, conforme exemplos na Tabela 3. Nesta tabela, a coluna “Original” representa a grafia original na base de dados disponibilizada pela Machine, a coluna “Carregado no Banco” mostra como as informações ficaram grafadas após carga no banco de dados PostGres e a coluna “Padronizado” apresenta a grafia após a padronização efetuada.

Tabela 3- Amostra de Padronização da Acentuação no Campo Bairro de Partida

Original	Carregado no Banco	Padronizado
Açude	AÃfÂŞude	Acude
Água Fria	Ãfiç½gua Fria	Agua Fria
Dionisio Torres	DionÃfÂ-sio Torres	Dionisio Torres
Fátima	FÃfÂtima	Fatima
Joaquim Távora	Joaquim TÃfÂjvora	Joaquim Tavora

Adicionalmente, foi observado que vários bairros foram grafados de mais de uma maneira na base. Segue abaixo, alguns bairros de Fortaleza como exemplo e as diferentes maneiras como foram grafados na base original:

Bairro Dionísio Torres:

Formas como foi grafado na base original:

- D Torres
- Dionósio Torres
- Dionísio Torres
- DIONISIO TORRES
- DIONISIO TORRES
- Dionizio Torres

Bairro Cidade dos Funcionários:

Formas como foi grafado na base original:

- C Funcionarios
- C. dos Funcionarios
- CID DOS Funcionarios
- Cid. dos Funcionario
- Cid. Funcionarios
- Cidade dos Func.
- Cidade dos Funcionário
- Cidade dos Funcionários
- Cidade Funcionario

Bairro de Fátima:

Formas como foi grafado na base original:

- BAIRRO DE FATIMA
- Campus de Fátima
- DE FATIMA
- Fátima
- FÁTIMA
- Fatima
- FATIMA

Se a grafia não fosse padronizada, seriam considerados como bairros distintos com probabilidades independentes a serem previstas no modelo, comprometendo o sucesso do mesmo.

Não foi computado o total de ajustes efetuados na base para fins de padronização, mas para exemplificar o esforço despendido nessa padronização, verifiquei que a base original apresentava 391 nomes de bairros de partida diferentes para a cidade de Fortaleza. Após os ajustes de padronização restaram apenas 251 bairros. Ou seja, cerca de 35% dos nomes correspondiam a “duplicidades” decorrentes da falta de padronização da base original.

3.2.4. Base de Informações de Entrada do Modelo de Inferência

As colunas “Id”, “Cidade”, “Dia”, “Dia Semana”, “Hora”, “Bairro Partida” e “Probabilidade” foram selecionados para compor a base de entrada do modelo de inferência, servindo de base de treino e teste do modelo.

Cabe lembrar que os campos “Cidade”, “Dia”, “Dia Semana”, “Hora” e “Bairro Partida” correspondem as variáveis preditoras selecionadas para treinamento do modelo, conforme já detalhado no item 3.2.1 desse documento. Já o campo “Probabilidade” corresponde a variável de resposta do modelo, conforme já mencionado na seção 3.2.2.

A Tabela 4 ilustra uma amostra da base de entrada do modelo de inferência.

Tabela 4- Amostra da Base de Informações com Probabilidade Calculada para Input no Modelo de Inferência

id	nome_cidade_partida	dia_da_criacao	dia_semana_da_criacao	hora_da_criacao	bairro_partida	PROBABILIDADE
1	Fortaleza	1	4	0	Alagadico Novo	0,018018018
2	Fortaleza	1	4	0	Aldeota	0,072072072
3	Fortaleza	1	4	0	Alvaro Weyne	0,009009009
4	Fortaleza	1	4	0	Bairro de Fatima	0,036036036
5	Fortaleza	1	4	0	Barroso	0,009009009
6	Fortaleza	1	4	0	Benfica	0,027027027
7	Fortaleza	1	4	0	Bom Futuro	0,009009009
8	Fortaleza	1	4	0	Cambeba	0,027027027
9	Fortaleza	1	4	0	Centro	0,063063063
10	Fortaleza	1	4	0	Cidade dos Funcionarios	0,018018018
11	Fortaleza	1	4	0	Coco	0,018018018
12	Fortaleza	1	4	0	Conjunto Prefeito Jose Walter	0,054054054
13	Fortaleza	1	4	0	Dionisio Torres	0,036036036
14	Fortaleza	1	4	0	Edson Queiroz	0,018018018
15	Fortaleza	1	4	0	Engenheiro Luciano Cavalcante	0,018018018
16	Fortaleza	1	4	0	Farias Brito	0,009009009
17	Fortaleza	1	4	0	Fortaleza	0,009009009
18	Fortaleza	1	4	0	Henrique Jorge	0,009009009
19	Fortaleza	1	4	0	Jardim das Oliveiras	0,036036036
20	Fortaleza	1	4	0	Joaquim Tavora	0,018018018
21	Fortaleza	1	4	0	Joquei Clube	0,009009009
22	Fortaleza	1	4	0	Jose Bonifacio	0,018018018
23	Fortaleza	1	4	0	Lagoa Sapiroanga (Coite)	0,063063063
24	Fortaleza	1	4	0	Mata Galinha	0,009009009
25	Fortaleza	1	4	0	Meireles	0,054054054
26	Fortaleza	1	4	0	Messejana	0,018018018
27	Fortaleza	1	4	0	Mondubim	0,027027027
28	Fortaleza	1	4	0	Mucuripe	0,018018018
29	Fortaleza	1	4	0	Padre Andrade	0,009009009
30	Fortaleza	1	4	0	Pan Americano	0,009009009
31	Fortaleza	1	4	0	Parangaba	0,027027027
32	Fortaleza	1	4	0	Parque Dois Irmaos	0,027027027
33	Fortaleza	1	4	0	Parquelandia	0,009009009
34	Fortaleza	1	4	0	Passare	0,081081081
35	Fortaleza	1	4	0	Patriolino Ribeiro	0,009009009
36	Fortaleza	1	4	0	Planalto Ayrton Senna	0,045045045
37	Fortaleza	1	4	0	Praia de Iracema	0,009009009

3.2.5. Seleção de Instâncias

A proposta foi desenvolver modelos preditivos específicos para cada uma das cidades da base de informações que originam transportes. Analisando as cidades de partida da base de informações, foi identificada a existência de 1.118 cidades diferentes. Não seria viável, no escopo desse TCC, o desenvolvimento de 1.118 modelos. Assim, coube eleger uma amostra de cidades para desenvolvimento dos modelos preditivos.

Classificando as cidades em ordem decrescente de quantidade de transportes, identificou-se que as 42 cidades com maior volume de transportes correspondem a 70% dos transportes demandados da base. Isso encontra-se ilustrado na Tabela 5.

Tabela 5- Instâncias (Cidades) selecionadas para Modelagem e Indicadores Utilizados na Seleção

#	CIDADE_PARTIDA	BAIROS	CORRIDAS	%	% ACUM.	CORRIDAS / BAIRRO
1	Porto Velho	718	333.900	12,7%	12,7%	465,0417827
2	Santarã@m	80	116.353	4,4%	17,1%	1454,4125

#	CIDADE_PARTIDA	BAIROS	CORRIDAS	%	% ACUM.	CORRIDAS / BAIRRO
3	Fortaleza	391	109.894	4,2%	21,3%	281,0588235
4	Minas Gerais	1.899	100.844	3,8%	25,1%	53,10373881
5	Rondônia	443	85.170	3,2%	28,3%	192,2573363
6	Patos de Minas	118	73.675	2,8%	31,1%	624,3644068
7	Sinop	259	72.778	2,8%	33,9%	280,996139
8	Petrolina	108	68.059	2,6%	36,5%	630,1759259
9	São Paulo	2.338	53.515	2,0%	38,5%	22,88922156
10	Palmas	134	50.158	1,9%	40,4%	374,3134328
11	Caruaru	72	45.569	1,7%	42,1%	632,9027778
12	Mossoró ³	80	42.309	1,6%	43,7%	528,8625
13	Ji-Paraná	111	42.184	1,6%	45,3%	380,036036
14	Mato Grosso	1.190	42.041	1,6%	46,9%	35,32857143
15	Corumbá	50	38.495	1,5%	48,4%	769,9
16	Rio de Janeiro	657	34.612	1,3%	49,7%	52,68188737
17	Ariquemes	109	34.576	1,3%	51,0%	317,2110092
18	Pará	285	32.858	1,2%	52,3%	115,2912281
19	Campos dos Goytacazes	290	32.850	1,2%	53,5%	113,2758621
20	Vilhena	136	30.758	1,2%	54,7%	226,1617647
21	Lins	124	26.432	1,0%	55,7%	213,1612903
22	Sorriso	104	23.951	0,9%	56,6%	230,2980769
23	Cacoal	104	22.886	0,9%	57,5%	220,0576923
24	Governador Valadares	155	22.137	0,8%	58,3%	142,8193548
25	Goiás	745	21.812	0,8%	59,1%	29,27785235
26	Caçeros	75	20.708	0,8%	59,9%	276,1066667
27	São João del-Rei	102	20.568	0,8%	60,7%	201,6470588
28	Bahia	471	19.273	0,7%	61,4%	40,91932059
29	Londrina	585	19.181	0,7%	62,2%	32,78803419
30	Juazeiro do Norte	67	19.160	0,7%	62,9%	285,9701493
31	Catanduva	167	18.903	0,7%	63,6%	113,1916168
32	Tangará da Serra	129	18.164	0,7%	64,3%	140,8062016
33	Itumbiara	100	17.998	0,7%	65,0%	179,98
34	Passos	86	16.388	0,6%	65,6%	190,5581395
35	Brasília	452	16.172	0,6%	66,2%	35,77876106
36	Rio Grande	83	16.104	0,6%	66,8%	194,0240964
37	Altamira	68	15.897	0,6%	67,4%	233,7794118
38	Jequiá	40	14.959	0,6%	68,0%	373,975
39	Barra do Garças	113	14.298	0,5%	68,5%	126,5309735
40	Vitória da Conquista	363	14.015	0,5%	69,1%	38,60881543
41	Amparo	121	13.752	0,5%	69,6%	113,6528926
42	Mineiros	93	13.638	0,5%	70,1%	146,6451613

Optou-se por eleger, portanto, as cidades de Fortaleza, Petrolina e Patos de Minas como instâncias dos modelos. Os motivadores foram:

- **Representatividade:** Como indicado, essas cidades estão entre as 8 com mais transportes registrados na base. Entende-se, portanto, que são relevantes para o negócio de aplicativos de transporte em questão. Além disso, a maior quantidade de informações históricas tende a proporcionar melhores resultados no treino do modelo preditivo.
- **Mix de Perfil:** Ao eleger uma capital e duas cidades do interior de portes diferentes, foi proporcionada a avaliação da aplicabilidade do modelo a diferentes tipos de cidades
- **Relação Corridas (Transportes) / Bairros:** O modelo a desenvolver pretende prever a probabilidade de transportes serem originados em cada bairro da cidade. Um volume elevado de corridas por bairro também contribui para melhores resultados no treino do modelo preditivo.

Distribuição dos Transportes (Corridas) ao longo do mês: Importante, para o modelo, que haja transportes ao longo do mês inteiro, sem “buracos”, com o mínimo de *missing values*. Importante, também, que não haja mudanças de patamar que denotem uma cidade em transição de perfil de comportamento. Esses são fatores que poderiam impactar negativamente o resultado do modelo. Como pode ser verificado nos gráficos da A Tabela 8 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador Nadam, na cidade de Patos de Minas, considerando transportes iniciados em qualquer hora do dia. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 8, A Tabela 9 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador Nadam, na cidade de Patos de Minas, considerando transportes iniciados entre 8 e 19 horas. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 9 e A Tabela 10 Tabela 8 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador SGD, na cidade de Patos de Minas, considerando transportes iniciados em qualquer hora do dia. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

- Tabela 10, as 3 cidades possuem demanda de transporte distribuído ao longo do mês. Fortaleza e Petrolina, em especial, apresentam sazonalidades claras em função dos dias da semana.

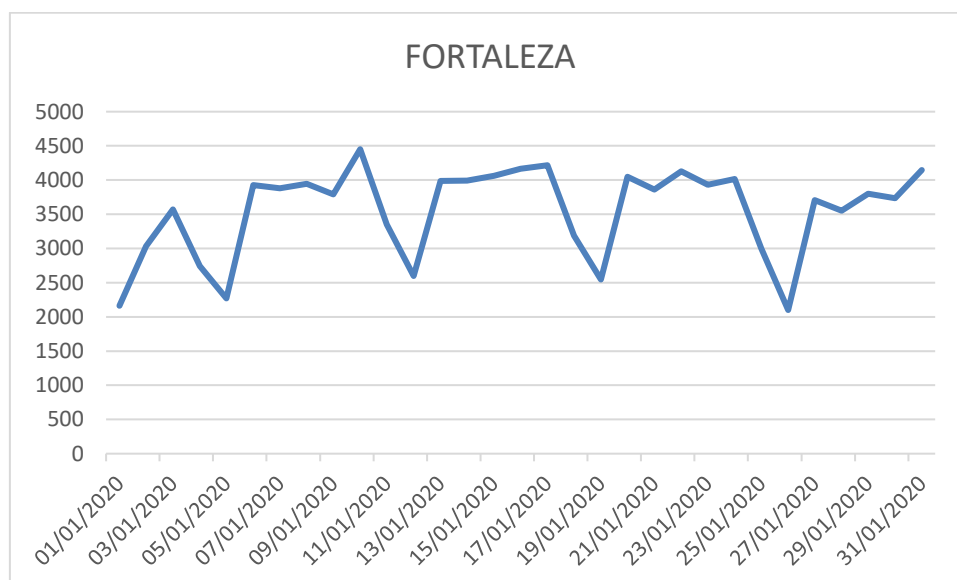


Figura 8- Total de Transportes Originados em Fortaleza ao longo de Janeiro/2020

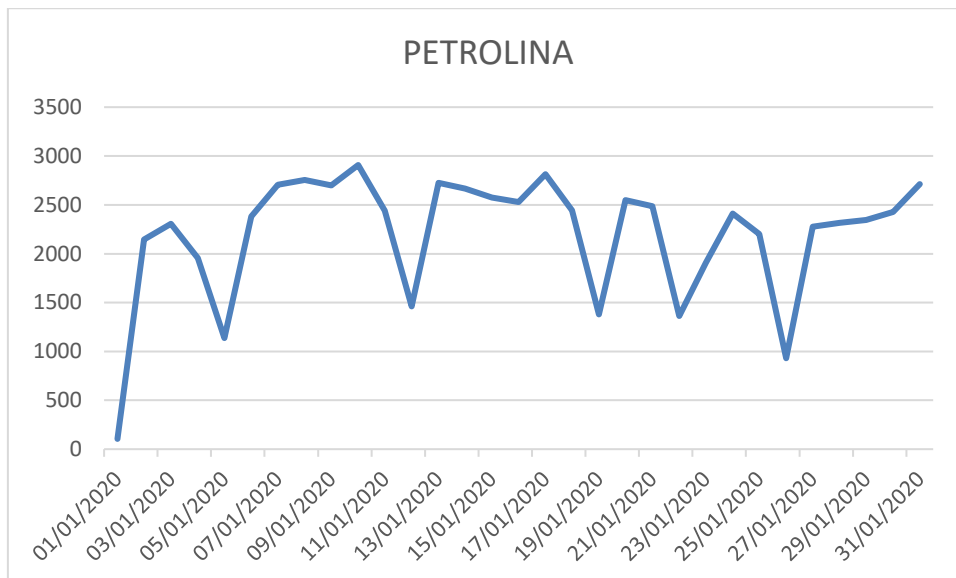


Figura 9- Total de Transportes Originados em Petrolina ao longo de Janeiro/2020



Figura 10- Total de Transportes Originados em Patos de Minas ao longo de Janeiro/2020

3.3. Inferência

O modelo de inferência foi desenvolvido em um programa utilizando a linguagem *Python*. O código completo do programa encontra-se no **Apêndice A – Código fonte do programa de inferência**.

O programa foi desenvolvido utilizando o **Google Colaboratory**. As diversas simulações executadas utilizaram a mesma plataforma.

As seções a seguir detalham as etapas executadas pelo programa de inferência e as simulações realizadas para parametrização do melhor modelo.

3.3.1. Carga da Base de Informações e Seleção da Cidade de Partida

Conforme descrito na seção 3.2.4, a base histórica com as informações de um mês de transportes foi transformada, a partir da etapa de processamento, em um arquivo em formato CSV (*Comma Separated Values*). A etapa inicial do programa lê esse arquivo usando o método `readcsv` da biblioteca Pandas do *python*.

Como o campo “ID” do arquivo é um índice dos registros da base, ele foi identificado através do método `set_index` do Pandas.

Por último, dado que a base gerada dispõe dos transportes de diversas cidades, são selecionados os registros da cidade que será usada na simulação de modo a gerar um modelo específico para aquela cidade.

3.3.2. Conversão para Dummy Variables

Como indicado na seção 3.2.2 deste documento, a base de entrada do modelo de inferência foi composta pelos campos “Id”, “Cidade”, “Dia”, “Dia Semana”, “Hora”, “Bairro Partida” e “Probabilidade”. O campo “Probabilidade” contém o valor a ser previsto e “Id” é apenas um rótulo dos registros, não devendo ser usado como parâmetro para treino do modelo.

Os campos de “Cidade” e “Bairro Partida” são “Categóricos”. Como são preenchidos com “texto” precisam ser convertidos para números de modo que a informação possa ser introduzida e usada em um modelo matemático.

Os demais campos, “Dia”, “Dia Semana”, “Hora”, embora sejam numéricos, também são categóricos. É adequado que sejam “ajustados” de modo que o modelo não interprete que os dias e horários de maior valor numérico tenham maior relevância que os demais para o resultado do modelo.

Assim sendo, todas as variáveis selecionadas para entrada no modelo foram convertidas para *Dummy Variables*. Por exemplo, a variável “Dia Semana”, representada na base pelo campo `dia_semana_da_criação`, o qual é preenchido com valores inteiros na faixa de 1 a 7, correspondendo a cada dia da semana, foi convertido nos campos `dia_semana_da_criação_1`, `dia_semana_da_criação_2`, `dia_semana_da_criação_3`, `dia_semana_da_criação_4`, `dia_semana_da_criação_5`, `dia_semana_da_criação_6` e `dia_semana_da_criação_7`.

O campo `dia_semana_da_criação_1` é preenchido com “1” para os registros em que originalmente o campo `dia_semana_da_criação` estava preenchido com “1”. Para os demais registros o campo `dia_semana_da_criação_1` é preenchido com “0”. Da mesma forma, O campo `dia_semana_da_criação_2` é preenchido com “1” para os registros em que originalmente o campo `dia_semana_da_criação` estava preenchido com “2”. Para os demais registros o campo `dia_semana_da_criação_2` é preenchido com “0”. O mesmo raciocínio é usado para as demais *Dummy Variables*.

3.3.3. Separação em Base de Treino e Base de Testes

Dado que a proposta é que o modelo preveja a probabilidade de transportes serem originados em cada bairro de uma cidade, a cada dia, de hora em hora e que, conforme apresentado no item 3.1, foi identificado que a base apresenta sazonalidade quanto aos dias da semana, foi considerado pertinente que a base de testes dispusesse de registros de todas as horas e todos os dias da semana com transportes disponíveis.

Assim, considerei para base de testes, os registros dos transportes que ocorreram no período de 12 a 18 de janeiro de 2020. Os transportes constantes da base de informações ocorridos nos demais dias do mês de janeiro de 2020 foram consideradas como base de treino.

3.3.4. Criação da Rede Neural

Para criação da rede neural foi utilizada a biblioteca *Keras* do *Python*.

Foi criada uma rede neural de 3 camadas com as seguintes dimensões:

- Camada de entrada com total de neurônios igual ao total de variáveis de entrada do modelo
- Camada “escondida” seguindo a heurística abaixo, quanto a quantidade de neurônios:

$$N_{escondida} = \frac{(N_{entrada} + N_{saida})}{2}$$

- Camada de saída com 1 (um) neurônio gerando o resultado da probabilidade de determinado bairro originar transportes em cada dia e hora.

Como função de ativação dos neurônios da rede foi utilizada a função “Relu”.

Foram realizadas simulações com vários otimizadores. Foram eles:

- SGD → Otimizador de Gradiente descendente com Momentum ³
- RMSProp → Otimizador que implementa o algoritmo RMSProp ⁴

A essência do RMSprop é:

- Mantém uma média móvel (com desconto) do quadrado dos gradientes
- Divide o gradiente pela raiz desta média

Esta implementação de RMSprop usa momentum simples, não momentum de Nesterov.

A versão centrada adicionalmente mantém uma média móvel dos gradientes e usa essa média para estimar a variância.

- Adam → Otimizador que implementa o algoritmo Adam ⁵

O otimizador Adam é um método de gradiente descendente estocástico que se baseia na estimativa adaptativa de momentos de primeira e segunda ordem.

De acordo com Kingma et al., 2014, o método é "computacionalmente eficiente, tem pouca necessidade de memória, invariante para o reescalonamento diagonal de gradientes e é adequado para problemas que são grandes em termos de dados / parâmetros".

- NAdam → Otimizador que implementa o algoritmo Nadam ⁶

Da mesma forma que o algoritmo Adam é essencialmente o algoritmo RMSProp conjugado com momentum, o algoritmo Nadam é o algoritmo Adam conjugado com o momentum Nesterov.

- Adamax → Otimizador que implementa o algoritmo Adamax ⁷

É uma variante do algoritmo Adam baseado na norma infinita.

- Adagrad → Otimizador que implementa o algoritmo Adagrad ⁸

O Adagrad é um otimizador com parâmetros de taxas de aprendizagem específicos, os quais são adaptados em função da frequência com que um parâmetro é atualizado durante o treinamento. Quanto mais atualizações um parâmetro recebe, menores são as atualizações.

- Adadelta → Otimizador que implementa o algoritmo Adadelta ⁹

A algoritmo Adadelta é um método de gradiente descendente estocástico que se baseia na taxa de aprendizagem adaptativa por dimensão para resolver duas desvantagens:

³ <https://keras.io/api/optimizers/sgd/>

⁴ <https://keras.io/api/optimizers/rmsprop/>

⁵ <https://keras.io/api/optimizers/adam/>

⁶ <https://keras.io/api/optimizers/Nadam/>

⁷ <https://keras.io/api/optimizers/adamax/>

⁸ <https://keras.io/api/optimizers/adagrad/>

⁹ <https://keras.io/api/optimizers/adadelta/>

- A queda contínua das taxas de aprendizagem ao longo do treinamento
- A necessidade de uma taxa de aprendizagem global selecionada manualmente

Como pode ser observado no item 4.1, os melhores resultados foram obtidos com os algoritmos Nadam e SGD.

As métricas utilizadas nos otimizadores para calcular a distância entre os valores previstos e os reais de modo a aproximar os resultados nas iterações sucessivas dos métodos foram as seguintes:

- *Mean Absolute Error (MAE)* ou Média do valor Absoluto dos Erros.

É calculado através da fórmula:

$$\frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

onde y_i são as previsões do modelo e x_i são os valores reais.

- *Mean Squared Error (MSE)* ou Média dos Quadrados dos Erros.

É calculado através da fórmula:

$$\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}$$

onde y_i são as previsões do modelo e x_i são os valores reais.

- *Root Mean Squared Error (RMSE)* ou Raiz da Média dos Quadrados dos Erros.

É calculado através da fórmula:

$$\sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}}$$

onde y_i são as previsões do modelo e x_i são os valores reais.

- *Mean Absolute Percentage Error (MAPE)* ou Média do Erro Percentual Absoluto.

É calculado através da fórmula:

$$\frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - x_i}{x_i} \right|$$

onde y_i são as previsões do modelo e x_i são os valores reais.

4. RESULTADOS

4.1. Simulações

Foram realizadas, ao todo, 127 simulações desse modelo com diferentes parâmetros. Nas simulações realizadas as seguintes variáveis e configurações foram alteradas:

- Número de Camadas da Rede Neural

Do total de 127 simulações, 116 foram realizadas com redes neurais de 3 camadas. Após atingir o melhor modelo com 3 camadas, algumas simulações foram realizadas com 4 ou 5 camadas (11 simulações no total). Os resultados não apresentaram ganhos significativos, portanto as melhores soluções mantiveram somente 3 camadas.

- Quantidade de Neurônios das Camadas Escondidas

As simulações, em geral, seguiram a heurística mencionada na seção 3.3.4 quanto a quantidade de neurônios das camadas escondidas.

- Otimizador

Foram realizadas simulações utilizando todos os otimizadores mencionados na seção 3.3.4. Os melhores resultados, contudo, foram obtidos com os otimizadores Nadam e SGD.

- Indicador de Perda

Foram realizadas simulações utilizando os indicadores MAE, MSE e MAPE (descritos na seção 3.3.4), associados ao algoritmo otimizador. Observou-se, contudo, que a maior parte dos valores a serem previstos eram muito pequenos. Considerando as bases de informações das cidades eleitas, 88% dos registros (bairros por horário) apresentavam probabilidade inferior a 5% (0,05). Isso está demonstrado na Tabela 6. Nesta tabela, a coluna “Faixas de Probabilidades” apresenta o limite superior e inferior da faixa de probabilidades dos registros consolidados nessa linha, a coluna “Linhas da Base de Informações” indica a quantidade de registros da base de informações cuja probabilidade está contida naquela faixa, onde cada registro representa a probabilidade de um bairro demandar corridas em uma data e hora. Por último, a coluna “%” indica a representatividade percentual do total de registros naquela faixa (coluna “Faixa de Probabilidades”) em relação ao total de registros da base.

Tabela 6- Distribuição do Total de Linhas da Base de Informações por Faixa de Probabilidades nas Cidades Eleitas

Faixas de Probabilidades	Linhas da Base de Informações	%
1-Maior que 20%	609	1%
2-Entre 10% e 20%	2.285	3%
3-Entre 5% e 10%	6.342	8%
4-Menor ou igual a 5%	70.147	88%
Total Geral	79.383	100%

Assim, os erros medidos pelos indicadores MAE e MSE eram muito pequenos, levando a resultados inferiores quando esses foram utilizados no modelo preditivo.

Já o indicador MAPE, por representar a distância percentual entre o valor previsto e o real, gerou valores mais significativos. Por exemplo, para um valor esperado de 0,3 e um valor estimado de 0,39, o indicador MAE gera como resultado somente 0,09 enquanto que o MAPE apresenta como resultado 30%. Por esse motivo, entende-se que os melhores resultados das simulações foram obtidos quando os otimizadores foram associados com o uso do indicador MAPE.

- Épocas de Treinamento

Foram realizadas simulações com até 20.000 épocas de treinamento. No entanto, o ajuste desse parâmetro evidenciou que bastavam em torno de 250 épocas para obter os melhores resultados.

A realização de simulações com até 20.000 épocas proporcionou a observação do fenômeno de *Overfitting* (Super Treinamento), conforme podemos observar no gráfico da Figura 11.

Tal gráfico representa a variação do indicador MAE em função do total de Épocas utilizadas na simulação. Nas simulações ilustradas nesse gráfico foi utilizada rede neural de 3 camadas com 160, 80 e 1 neurônio respectivamente. O otimizador utilizado foi o Adam. A cidade em questão foi Petrolina.

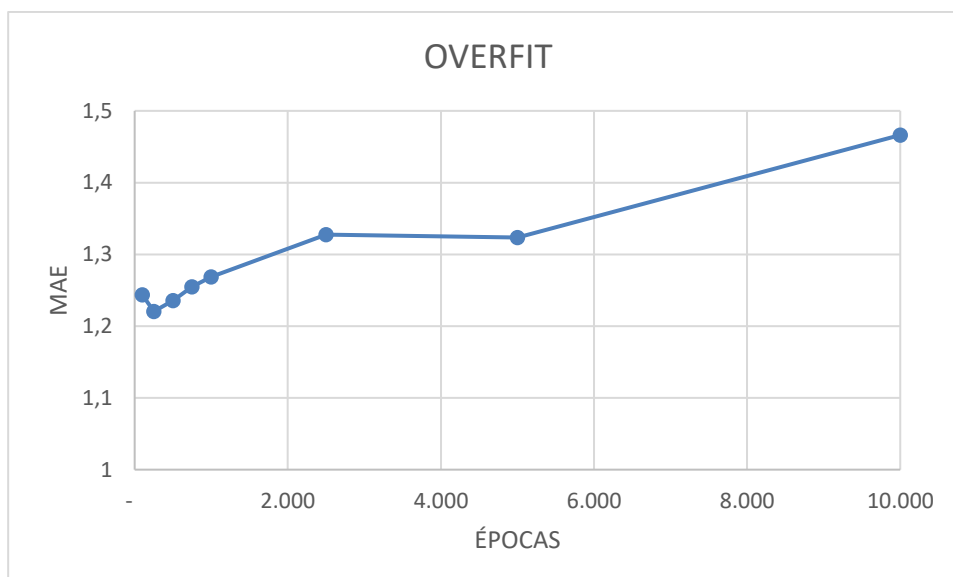


Figura 11- Overfit em rede neural de 3 camadas com 160, 80 e 1 neurônios e Otimizador Adam

- **Faixa Horária**

Na base de informações disponibilizada de transportes realizados, para as cidades eleitas nesse trabalho, 73% dos transportes ocorriam no período de 8 as 19 horas. Por conta disso, foram executadas não só simulações visando desenvolver modelos para prever a probabilidade de transportes a qualquer hora do dia, mas também modelos visando prever somente os transportes no período de 8 às 19 horas, faixa próxima ao “Horário Comercial”.

O objetivo foi otimizar os resultados dos modelos nesse período horário (próximo ao “Horário Comercial”), sem prejuízo da relevância dos resultados.

A Tabela 7 apresenta a quantidade de transportes realizados a cada hora de partida para as cidades “eleitas” mencionadas na seção 3.2.5, bem como a representatividade percentual da quantidade de registros sobre o total. Já a Figura 12 apresenta, graficamente, a distribuição do total de transportes realizados por hora para cada uma das cidades eleitas.

Tabela 7- Transportes Realizados por Hora de Partida para as Cidades Eleitas

Hora de Partida	Transportes	%	% Grupos
0	2.849	1,13%	14,29%
1	1.883	0,75%	
2	1.364	0,54%	
3	1.150	0,46%	
4	1.200	0,48%	
5	1.953	0,78%	
6	8.764	3,48%	
7	16.799	6,68%	
8	16.474	6,55%	73,42%

Hora de Partida	Transportes	%	% Grupos
9	15.369	6,11%	
10	13.966	5,55%	
11	13.919	5,53%	
12	14.845	5,90%	
13	15.811	6,28%	
14	15.824	6,29%	
15	15.755	6,26%	
16	15.824	6,29%	
17	16.624	6,61%	
18	16.537	6,57%	
19	13.809	5,49%	
20	10.549	4,19%	12,28%
21	8.799	3,50%	
22	6.774	2,69%	
23	4.787	1,90%	
Total	251.628	100,00%	

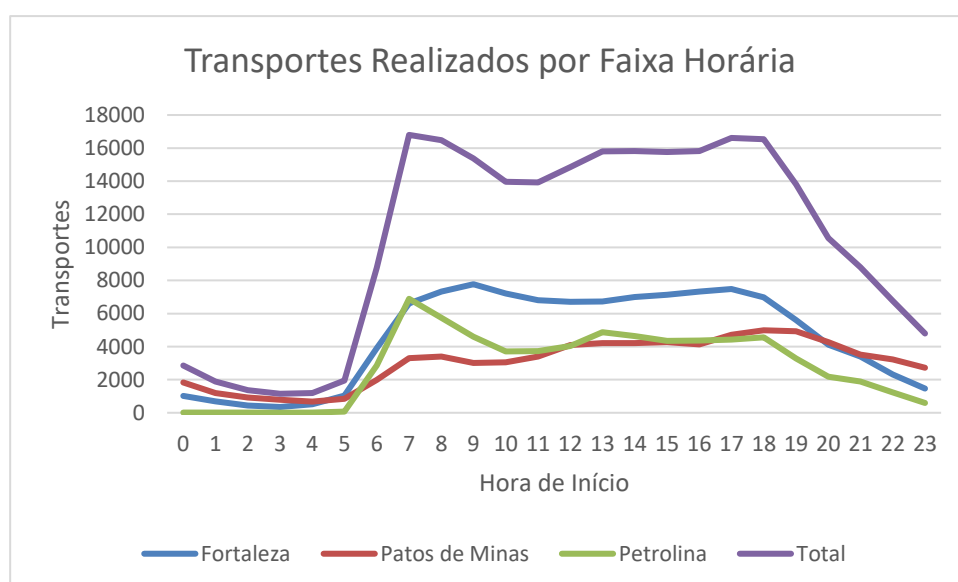


Figura 12- Distribuição de Transportes Realizados por Faixa Horária

A relação completa de simulações realizadas encontra-se na seção “


```

#cidade_partida = 'Santarem'
#cidade_partida = 'Petroлина'

if (cidade_partida == 'Fortaleza'):
    df1 = pd.read_csv('Corridas_Fortaleza-FINAL-Acentuado-Probabilidades-ComSemanaTeste.csv')
else:
    df1 = pd.read_csv('Corridas_Cidades_Eleitas-FINAL-Acentuado-Probabilidades-ComSemanaTeste.csv')

df1=df1[df1['nome_cidade_partida'] == cidade_partida]

### TESTANDO SÓ PERÍODO DO HORÁRIO COMERCIAL
# Se for rodar simulação somente para um período do dia, descomentar a linha abaixo
df1 = df1[(df1['hora_da_criacao'] >= 8) & (df1['hora_da_criacao'] <= 19)]

# colocar id como nome de linha
df1 = df1.set_index('id')
df1.head(1000)


```

id	nome_cidade_partida	dia_da_criacao	dia_semana_da_criacao	hora_da_criacao	b
241	Fortaleza	1	4	8	
242	Fortaleza	1	4	8	
243	Fortaleza	1	4	8	
244	Fortaleza	1	4	8	
245	Fortaleza	1	4	8	
...
1519	Fortaleza	2	5	15	
1520	Fortaleza	2	5	15	
1521	Fortaleza	2	5	15	
1522	Fortaleza	2	5	15	
1523	Fortaleza	2	5	15	

1000 rows × 6 columns

```

#dimensões da base
df1.shape


```

(42510, 6)

```

### MULTIPLICA POR 100 A PROBABILIDADE PARA DIMINUIR PROBLEMAS DE CÁLCULOS COM VALORES PEQ
df1['PROBABILIDADE'] = df1['PROBABILIDADE']*100
df1['PROBABILIDADE'].head()

```

```

id
241    4.761905
242    1.587302
243    1.587302
244    1.587302
245    1.587302
Name: PROBABILIDADE, dtype: float64

```

▼ Converte Variáveis para DUMMY

▼ Converte Variáveis Numéricas para String

```

# Essa conversão é necessária para permitir que no passo seguinte sejam convertidas para D
# Foram convertidas para DUMMY pois são categóricas
df1['dia_semana_da_criacao'] = df1['dia_semana_da_criacao'].astype(str)
df1['dia_da_criacao'] = df1['dia_da_criacao'].astype(str)
df1['hora_da_criacao'] = df1['hora_da_criacao'].astype(str)

```

▼ Conversão para DUMMY

```

df1 = pd.get_dummies(df1)
df1.head()

```

```

PROBABILIDADE  nome_cidade_partida_Fortaleza  dia_da_criacao_1  dia_da_criacao_
id
241          4.761905                                1          1
242          1.587302                                1          1
243          1.587302                                1          1
244          1.587302                                1          1
245          1.587302                                1          1
5 rows x 303 columns

```

▼ Separa em Base de Treino e Base de Teste

▼ Separa a Semana de Teste

```

# Criei uma semana de teste na base de entrada com todas as combinações
# necessário de treino dia da semana e horário para extrair as estimativas
https://colab.research.google.com/drive/1a7lm8bKWPgHUSa-xDs3SjH6KIBK13Zz5#scrollTo=QyKc1qZWu9pM&printMode=true

```

3/10

28/09/2020

TCC-Gaudium-SemanaTeste - Limpo.ipynb - Colaboratory

```
# POSSÍVEIS DE OBITOS, dia da semana e horário para extrair as estimativas
# do modelo para essa semana
# A semana corresponde ao mesmo período da Base de Teste, ou seja,
# 12 a 18 de Janeiro

#SEPARA A SEMANA DE TESTE
if cidade_partida == 'Petrolina':
    df_semana_teste = df1.loc[200000:299999] #PETROLINA
    df1.loc[200000:299999]
    df1 = df1.loc[:199999]
elif cidade_partida == 'Patos de Minas':
    df_semana_teste = df1.loc[100000:199999] #Patos de MINAS
    df1.loc[100000:199999]
    df1 = df1.loc[:99999]
elif cidade_partida == 'Fortaleza':
    df_semana_teste = df1.loc[300000:399999] #Fortaleza
    df1.loc[300000:399999]
    df1 = df1.loc[:299999]

# EXCLUI A COLUNA PROBABILIDADE (TODA ZERADA) DA SEMANA DE TESTE
# Não precisamos dessa informação pois essa será a variável a ser prevista
# pelo modelo
df_semana_teste.head()
df_semana_teste=df_semana_teste.drop(columns=['PROBABILIDADE'])
df_semana_teste.head()
```



```
nome_cidade_partida_Fortaleza  dia_da_criacao_1  dia_da_criacao_10  dia_da_c
```

id			
300008	1	0	0
300009	1	0	0
300010	1	0	0
300011	1	0	0
300012	1	0	0

5 rows × 302 columns

▼ Criação de Base de Teste "Aleatória"

▼ Indicar a semente inicial e para divisão da base em treino e teste

```
## NÃO usei essa solução no treino do MODELO
## Usei a opção seguinte de base de teste direcionada para o período de
## 12 a 18 de Janeiro
import random
```

<https://colab.research.google.com/drive/1a7im8bkWPgHU5a-xDs3SjH6K1BK13Zz5#scrollTo=QyKc1qZWu9pM&printMode=true>

4/10

```
np.random.seed(0) #semente inicial
nlinhas = df1.shape[0]
nlinhas
```

```
21426
```

```
from sklearn.model_selection import train_test_split
#Divide a base em treino e teste. A coluna 'PROBABILIDADE' é o Label. Test_size diz o tama
#x_train, x_test, y_train, y_test = train_test_split(df1.drop(columns=['PROBABILIDADE']),
#                                                    df1['PROBABILIDADE'], test_size=0.3)
#x_train.head()
```

▼ Criação de Base de Teste "Direcionada" (Semana de 12 a 18 / Janeiro)

```
## VERSÃO DIRECIONADA PARA TREINAR COM MÊS TODO EXCETO SEMANA DE TESTE
```

```
x_test = df1[(df1['dia_da_criacao_12'] == 1)|(df1['dia_da_criacao_13'] == 1)|(df1['dia_da_
y_test = x_test['PROBABILIDADE']
x_test = x_test.drop(columns=['PROBABILIDADE'])
```

```
x_train = df1[(df1['dia_da_criacao_12'] == 0)&(df1['dia_da_criacao_13'] == 0)&(df1['dia_da_
y_train = x_train['PROBABILIDADE']
x_train = x_train.drop(columns=['PROBABILIDADE'])
x_train
y_train
```

```
id
241    4.761905
242    1.587302
243    1.587302
244    1.587302
245    1.587302
...
31427   0.442478
31428   0.442478
31429   0.442478
31430   0.884956
31431   0.442478
Name: PROBABILIDADE, Length: 16470, dtype: float64
```

▼ Rede Neural

▼ Converte Dataframe de Pandas para Numpy para inserir na RN

```
## Conversão necessária para inserir dados nas funções da RN
X_train_normalized = x_train.to_numpy()
X_test_normalized = x_test.to_numpy()
df_semana_teste_normalized = df_semana_teste.to_numpy()
```

▼ Importa bibliotecas necessárias para uso na RN

```
import tensorflow as tf
from keras import Model, Sequential
from keras.layers import Dense
from keras.optimizers import SGD, RMSprop, Adam, Adamax, Adagrad, Adadelta, Nadam
```

```
X_train_normalized.shape[1:]
```

```
↳ (302,)
```

▼ Cria a Rede Neural

```
#Inicia a rede
RN = Sequential()
# Cria primeira camada e 'input_shape' entradas
RN.add(Dense(314, input_shape = X_train_normalized.shape[1:], activation = 'relu')) # antes
#RN.add(Dropout(0.2))

RN.add(Dense(157, activation = 'relu'))
#RN.add(Dense(40, activation = 'relu'))
#RN.add(Dense(22, activation = 'relu'))
#RN.add(Dense(50, activation = 'sigmoid'))
#RN.add(dropout(0.2))
#RN.add(Dense(4, activation = 'sigmoid'))
RN.add(Dense(1, activation = 'relu'))
# linha do programa original --> Cria segunda e última camada 2 células (número de classes)
# RN.add(Dense(NumberOfClasses, activation = 'sigmoid'))

# Diferença Sigmoidal X Softmax
# https://www.google.com/search?q=sigmoid+vs+softmax&oq=sigmoid&aqs=chrome.3.69i57j35i39j0

RN.summary()
```

```
↳
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 314)	95142
dense_1 (Dense)	(None, 157)	48455

▼ Treina a Rede Neural

```
Total params: 144,355

# treinamento

#Otimização por Gradiente Descendente (SGD)
#otimizador = SGD(lr=0.001, decay=1e-6, momentum=0.9) #decay=1e-6,
#otimizador = RMSprop()
#otimizador = SGD()
#otimizador = Adam()
otimizador = Nadam()

#Configura o modelo para treinamento
#RN.compile(optimizer = sgd, loss = 'mean_squared_error', metrics = ['accuracy', 'mean_squ
RN.compile(optimizer = otimizador, loss = 'mean_absolute_percentage_error', #tf.keras.loss
            metrics = ['mean_absolute_error', #tf.keras.metrics.RootMeanSquaredError(),
                      'mean_squared_error', 'mean_absolute_percentage_error', 'accuracy' ])

# Treina o modelo para um certo numero de epocas
trainedRN = RN.fit(X_train_normalized,y_train, epochs = 350, verbose = 0)
```

▼ Métricas de Avaliação

▼ Métricas Calculadas Automaticamente pelo Keras

```
trainedRN.model.metrics_names

['loss',
 'mean_absolute_error',
 'mean_squared_error',
 'mean_absolute_percentage_error',
 'accuracy']

import math
score = trainedRN.model.evaluate(X_test_normalized, y_test, verbose = 0)
print('Test score (loss):', score[0])
print('Test MAE:', score[1])
print('Test RMSE:', math.sqrt(score[2]))
print('Test MSE:', score[2])
```

```
print('MAPE:', score[3])
print('Test Accuracy:', score[4])
```

Test score (loss): 34.15443420410156
 Test MAE: 0.5490463972091675
 Test RMSE: 0.962030920685685
 Test MSE: 0.9255034923553467
 MAPE: 34.15443420410156
 Test Accuracy: 0.0014124293811619282

▼ Métricas Calculadas por mim para Validação do Resultado do Keras

```
# Previsão
y_test_predicted = RN.predict(X_test_normalized)

#import math
from sklearn.metrics import mean_squared_error
rmse = math.sqrt(mean_squared_error(y_test, y_test_predicted))
mse = mean_squared_error(y_test, y_test_predicted)
mape = np.mean(np.abs((y_test.to_numpy() - y_test_predicted.transpose())/y_test.to_numpy()))
mae = np.mean(np.abs((y_test.to_numpy() - y_test_predicted.transpose())))

print('MAE: ',mae)
print('RMSE: ', rmse)
print('MSE: ',mse)
print('MAPE: ',mape)
```

MAE: 0.5490463541090214
 RMSE: 0.9620308013503334
 MSE: 0.9255032627467645
 MAPE: 34.154424609962994

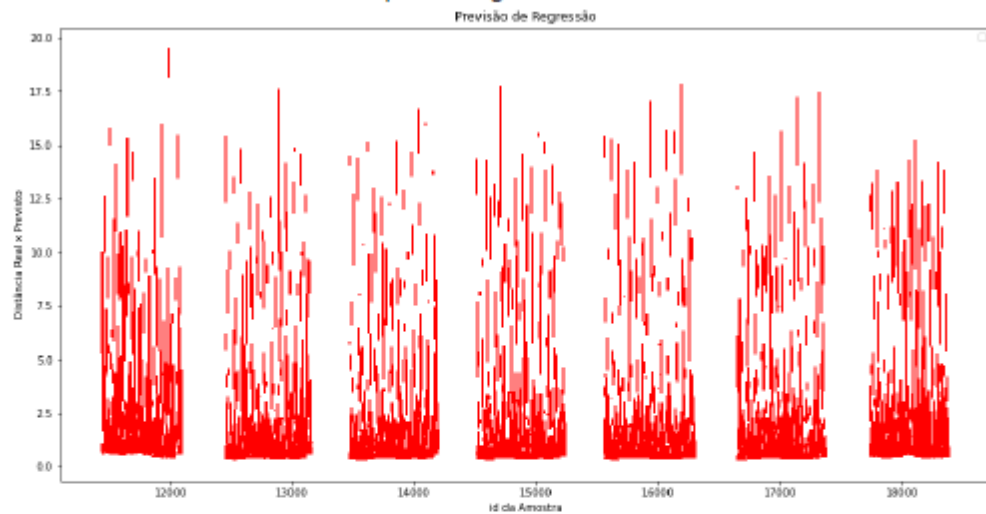
Apresentação Gráfica da Distância entre Probabilidade Real e Estimada

```
plt.figure(figsize=(16, 8))
plt.vlines(y_test.index, y_test_predicted,y_test.values, color='red', linewidth=2)

plt.title('Previsão de Regressão')
plt.xlabel('id da Amostra')
plt.ylabel('Distância Real x Previsto')
plt.legend()
plt.savefig('predictions_training_test.pdf',format = 'pdf')
plt.show()
```

↳

No handles with labels found to put in legend.



▼ Faz a previsão da Semana de Testes e Grava resultados para Análise

```
y_semana_teste = RN.predict(df_semana_teste_normalized)
df_semana_teste.insert(loc=df_semana_teste.shape[1], column='PROBABILIDADE', value=y_semana_teste)
df_semana_teste.shape[1]
df_semana_teste.head()
df_semana_teste.to_csv(path_or_buf = 'Resultados_Semana_Testes.csv', sep=';', decimal = ',',
```


4.2. Melhores Resultados

O melhor resultado obtido, para cada uma das cidades eleitas nesse trabalho, está listado nas seções a seguir.

4.2.1. Cidade: Patos de Minas

4.2.1.1. Otimizador Nadam – Todas as horas do dia

A Tabela 8 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador Nadam, na cidade de Patos de Minas, considerando transportes iniciados em qualquer hora do dia. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 8- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador Nadam

Faixa Horária:	0 a 23 horas
Algoritmo Otimizador:	Nadam
Parâmetros do Algoritmo Otimizador:	Default
Rede Neural:	3 camadas densamente conectadas com 176, 88 e 1 neurônio respectivamente
Total de Épocas:	50
Indicador de Perda:	MAPE
Resultados:	RMSE: 2,2541 MSE: 5,0809 MAE: 1,0858 MAPE: 41,79362

Já o gráfico da Figura 13 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

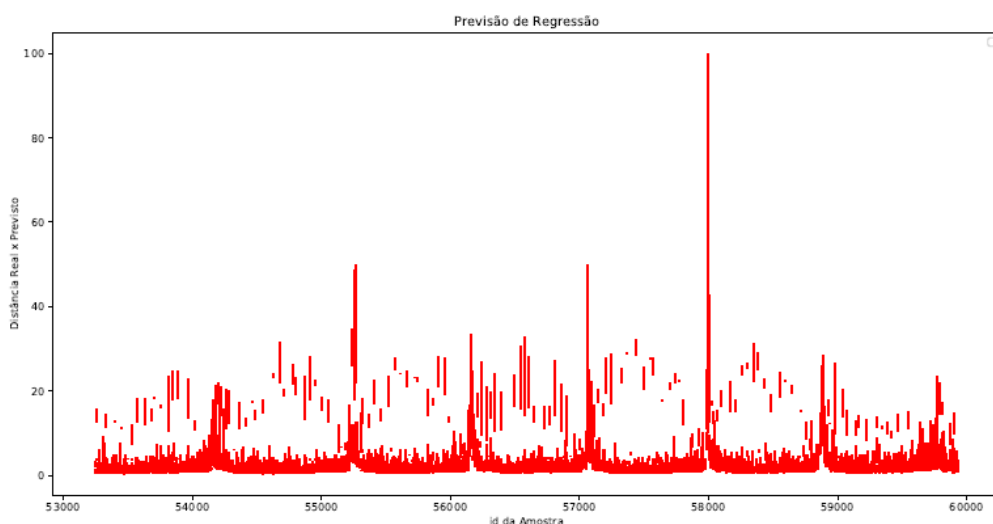


Figura 13- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Patos de Minas

4.2.1.2. Otimizador Nadam – Horário de 8 a 19 horas

A Tabela 9 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador Nadam, na cidade de Patos de Minas, considerando transportes iniciados entre 8 e 19 horas. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 9- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador Nadam (Horário "Comercial")

Faixa Horária:	8 a 19 horas
Algoritmo Otimizador:	Nadam
Parâmetros do Algoritmo Otimizador:	Default
Rede Neural:	3 camadas densamente conectadas com 176, 88 e 1 neurônio respectivamente
Total de Épocas:	50
Indicador de Perda:	MAPE
Resultados:	RMSE: 1,2777 MSE: 1,6324 MAE: 0,8226 MAPE: 43,17156982

Já o gráfico da Figura 14 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

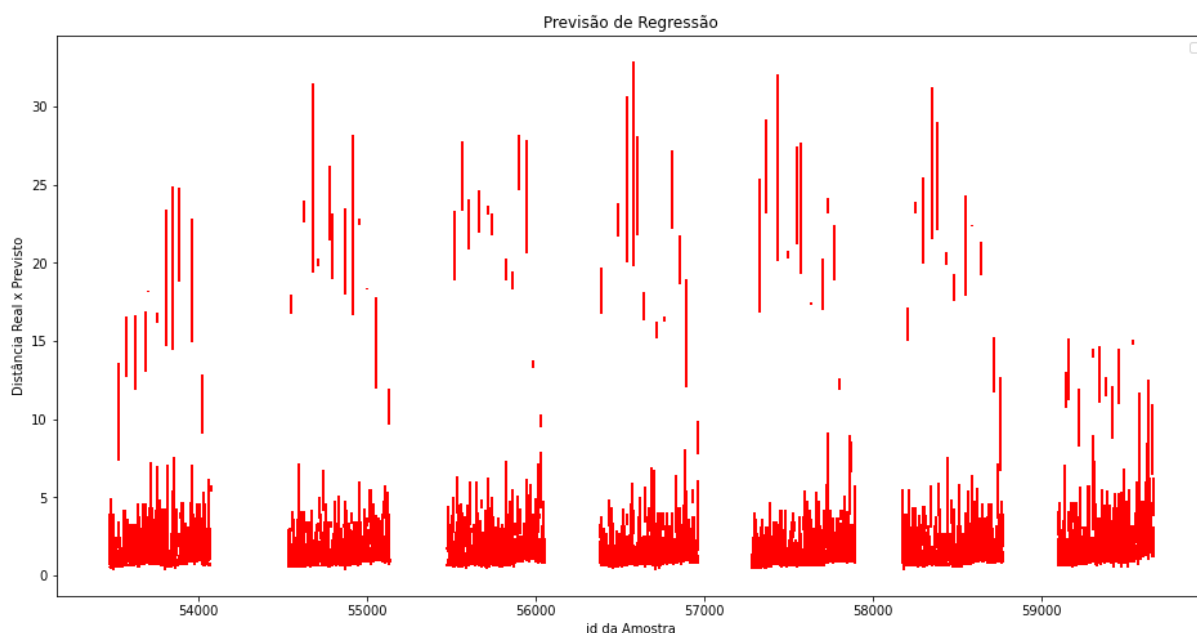


Figura 14- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Patos de Minas Horário "Comercial"

4.2.1.3. Otimizador SGD – Todas as horas do dia

A Tabela 10Tabela 8 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador SGD, na cidade de Patos de Minas, considerando transportes iniciados em qualquer hora do dia. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 10- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador SGD

Faixa Horária:	0 a 23 horas
Algoritmo Otimizador:	Nadam
Parâmetros do Algoritmo Otimizador:	Default
Rede Neural:	3 camadas densamente conectadas com 176, 88 e 1 neurônio respectivamente
Total de Épocas:	50
Indicador de Perda:	MAPE
Resultados:	RMSE: 2,1620 MSE: 4,6743 MAE: 1,0783 MAPE: 50,30212402

Já o gráfico da Figura 14 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

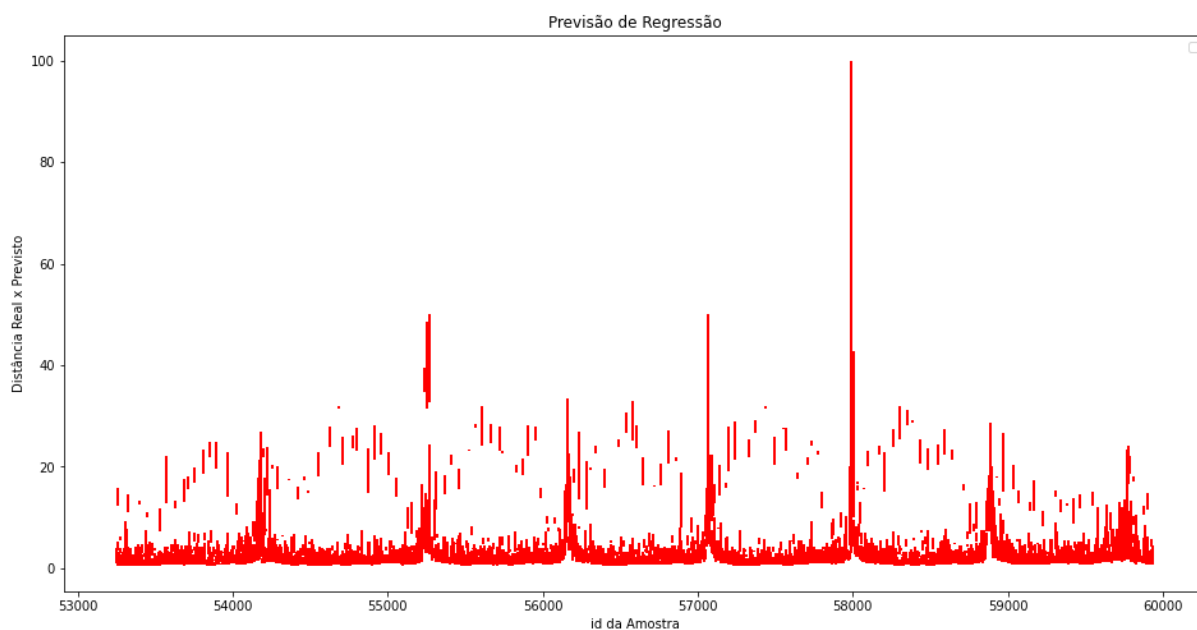


Figura 15- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Patos de Minas

4.2.1.4. Otimizador SGD – Horário de 8 a 19 horas

A Tabela 11 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador SGD, na cidade de Patos de Minas, considerando transportes iniciados entre 8 e 19 horas. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 11- Parâmetros e Resultados da Melhor Simulação de Patos de Minas com Otimizador SGD (Horário "Comercial")

Faixa Horária:	8 a 19 horas
Algoritmo Otimizador:	Nadam
Parâmetros do Algoritmo Otimizador:	Default
Rede Neural:	3 camadas densamente conectadas com 176, 88 e 1 neurônio respectivamente
Total de Épocas:	250
Indicador de Perda:	MAPE
Resultados:	RMSE: 1,3636 MSE: 1,8595 MAE: 0,8557 MAPE: 55,00800812

Já o gráfico da Figura 16 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

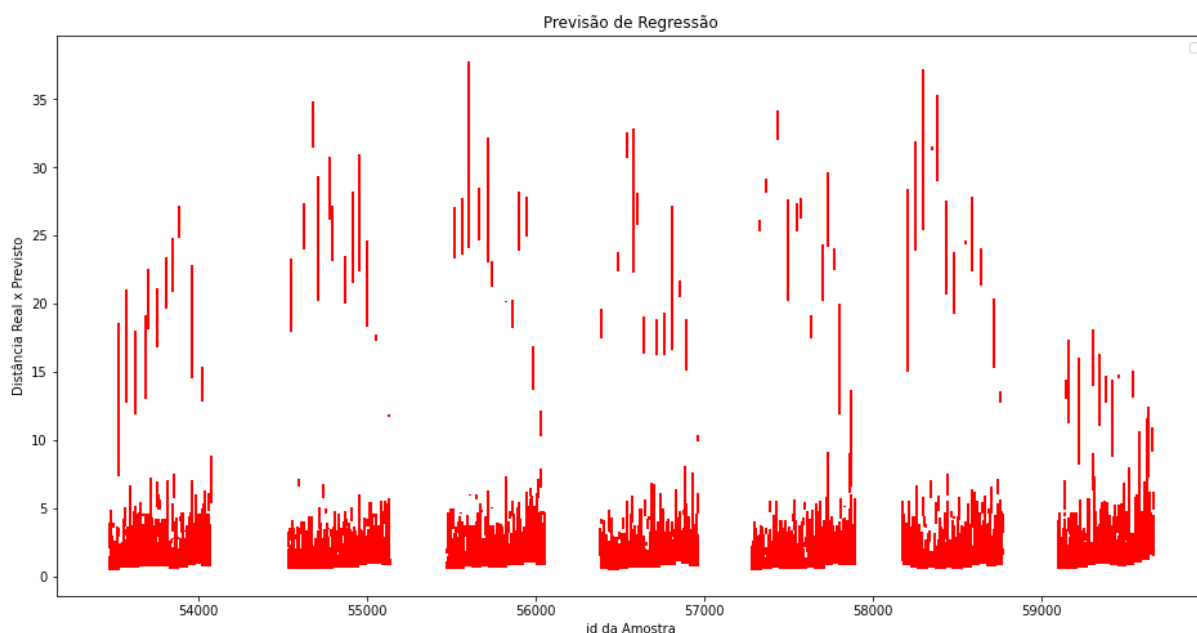


Figura 16- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Patos de Minas Horário "Comercial"

4.2.2. Cidade: Fortaleza

4.2.2.1. Otimizador Nadam – Todas as horas do dia

A Tabela 12 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador Nadam, na cidade de Fortaleza, considerando transportes iniciados em qualquer hora do dia. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 12- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador Nadam

Faixa Horária:	0 a 23 horas
Algoritmo Otimizador:	Nadam
Parâmetros do Algoritmo Otimizador:	Default
Rede Neural:	3 camadas densamente conectadas com 314, 157 e 1 neurônio respectivamente
Total de Épocas:	10
Indicador de Perda:	MAPE
Resultados:	RMSE: 1,6736 MSE: 2,8009 MAE: 0,797523916 MAPE: 31,68181038

Já o gráfico da Figura 17 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

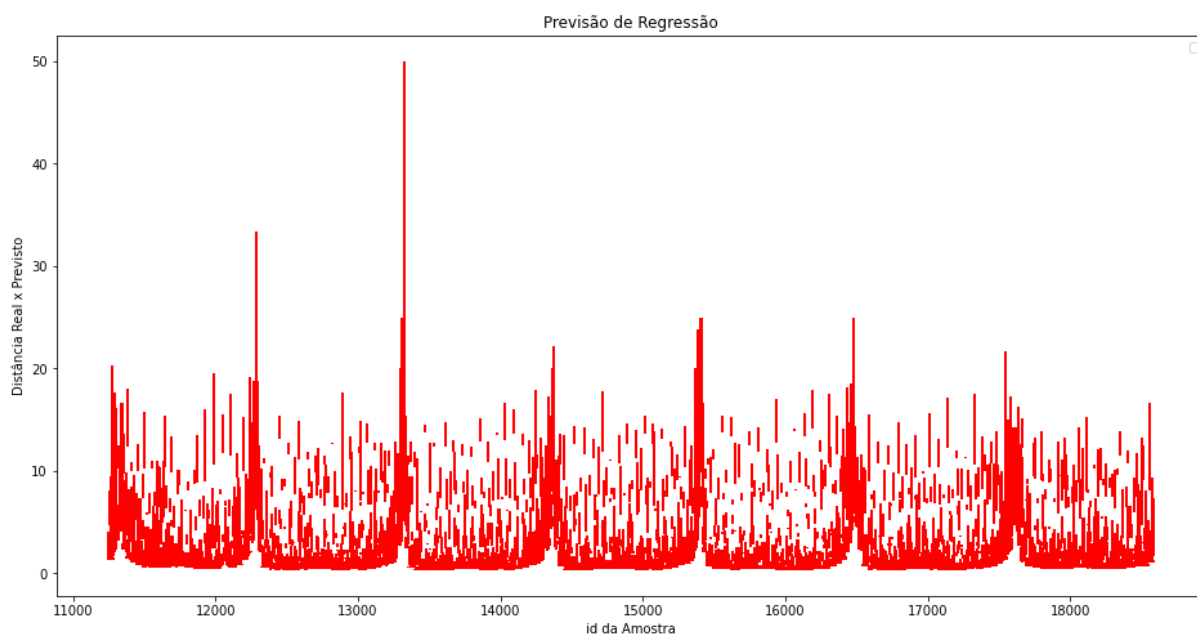


Figura 17- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Fortaleza

4.2.2.2. Otimizador Nadam – Horário de 8 a 19 horas

A Tabela 13 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador Nadam, na cidade de Fortaleza, considerando transportes iniciados entre 8 e 19 horas. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 13- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador Nadam (Horário "Comercial")

Faixa Horária:	8 a 19 horas		
Algoritmo Otimizador:	Nadam		
Parâmetros do Algoritmo Otimizador:	Default		
Rede Neural:	3 camadas densamente conectadas com 314, 157 e 1 neurônio respectivamente		
Total de Épocas:	350		
Indicador de Perda:	MAPE		
Resultados:	RMSE:	0,9447	
	MSE:	0,8924	
	MAE:	0,5397	
	MAPE:	34,20235825	

Já o gráfico da Figura 18 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

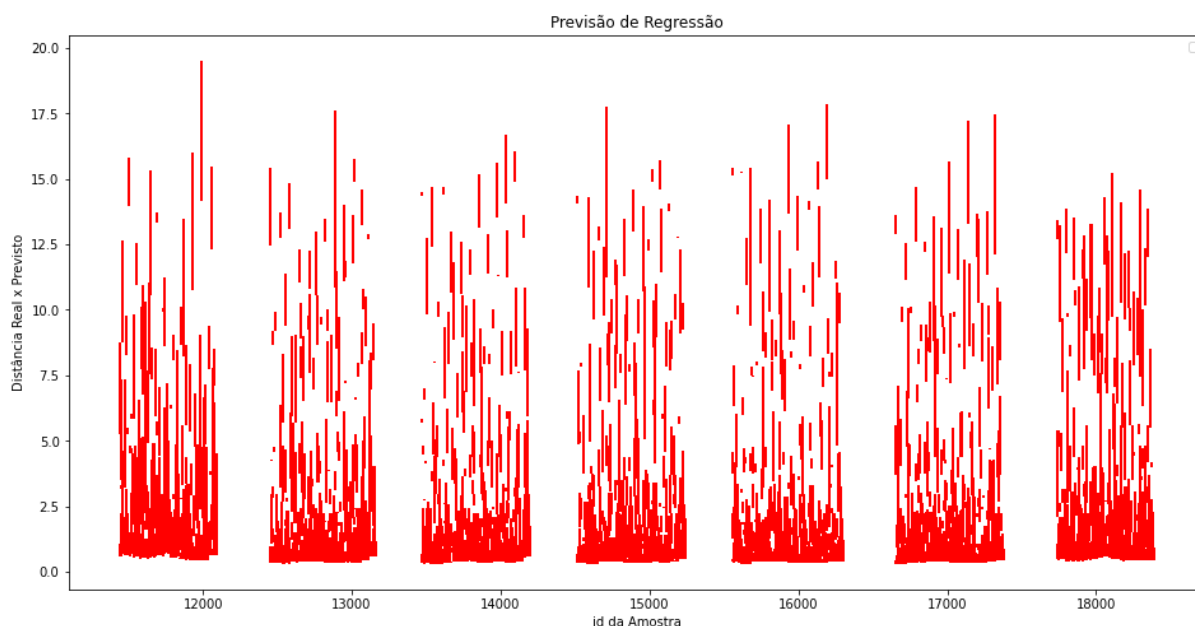


Figura 18- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Fortaleza Horário "Comercial"

4.2.2.3. Otimizador SGD – Todas as horas do dia

A Tabela 14Tabela 10Tabela 8 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador SGD, na cidade de Fortaleza, considerando transportes iniciados em qualquer hora do dia. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 14- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador SGD

Faixa Horária:	0 a 23 horas
Algoritmo Otimizador:	SGD
Parâmetros do Algoritmo Otimizador:	Learning Rate: 0,001 Decay: 1,00E-06 Momentum: 0,9 Obs: O modelo não converge quando se usa o SGD com parâmetros default para Fortaleza. O MAPE fica em 100%
Rede Neural:	3 camadas densamente conectadas com 314, 157 e 1 neurônio respectivamente
Total de Épocas:	250
Indicador de Perda:	MAPE
Resultados:	RMSE: 1,6331 MSE: 2,6672 MAE: 0,809509695 MAPE: 37,11328506

Já o gráfico da Figura 19 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

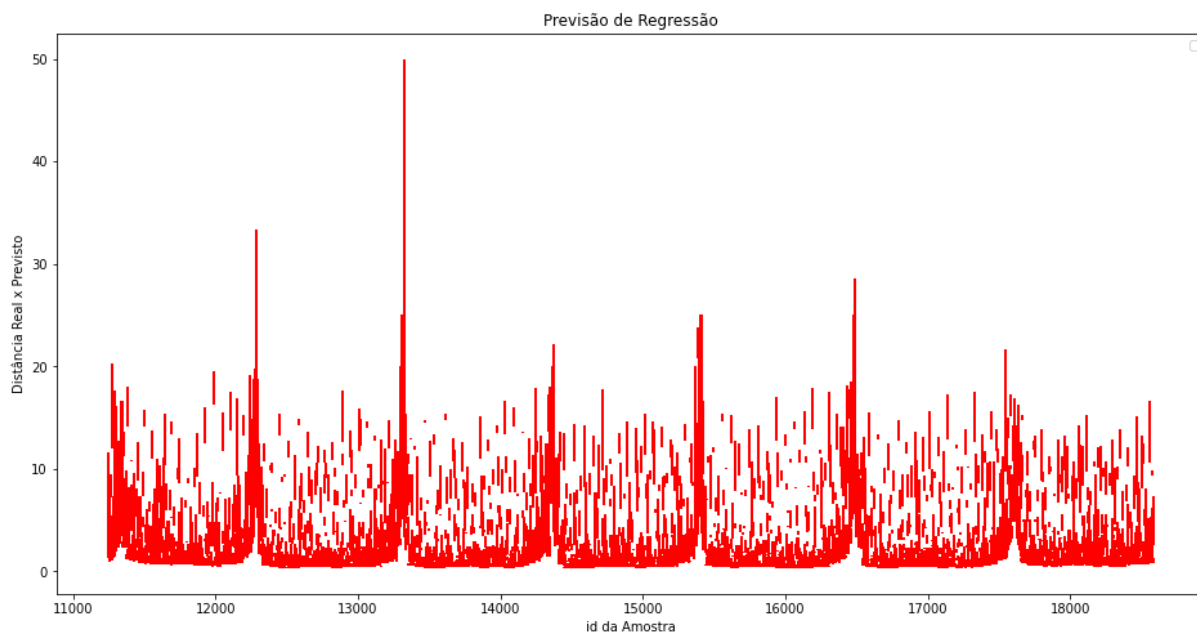


Figura 19- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Fortaleza

4.2.2.4. Otimizador SGD – Horário de 8 a 19 horas

A Tabela 15 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador SGD, na cidade de Fortaleza, considerando transportes iniciados entre 8 e 19 horas. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 15- Parâmetros e Resultados da Melhor Simulação de Fortaleza com Otimizador SGD (Horário "Comercial")

Faixa Horária:	8 a 19 horas
Algoritmo Otimizador:	SGD
Parâmetros do Algoritmo Otimizador:	Learning Rate: 0,001 Decay: 1,00E-06 Momentum: 0,9 Obs: O modelo não converge quando se usa o SGD com parâmetros default para Fortaleza. O MAPE fica em 100%
Rede Neural:	3 camadas densamente conectadas com 314, 157 e 1 neurônio respectivamente
Total de Épocas:	250
Indicador de Perda:	MAPE
Resultados:	RMSE: 0,9810 MSE: 0,9623 MAE: 0,5675 MAPE: 36,7091217

Já o gráfico da Figura 20 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

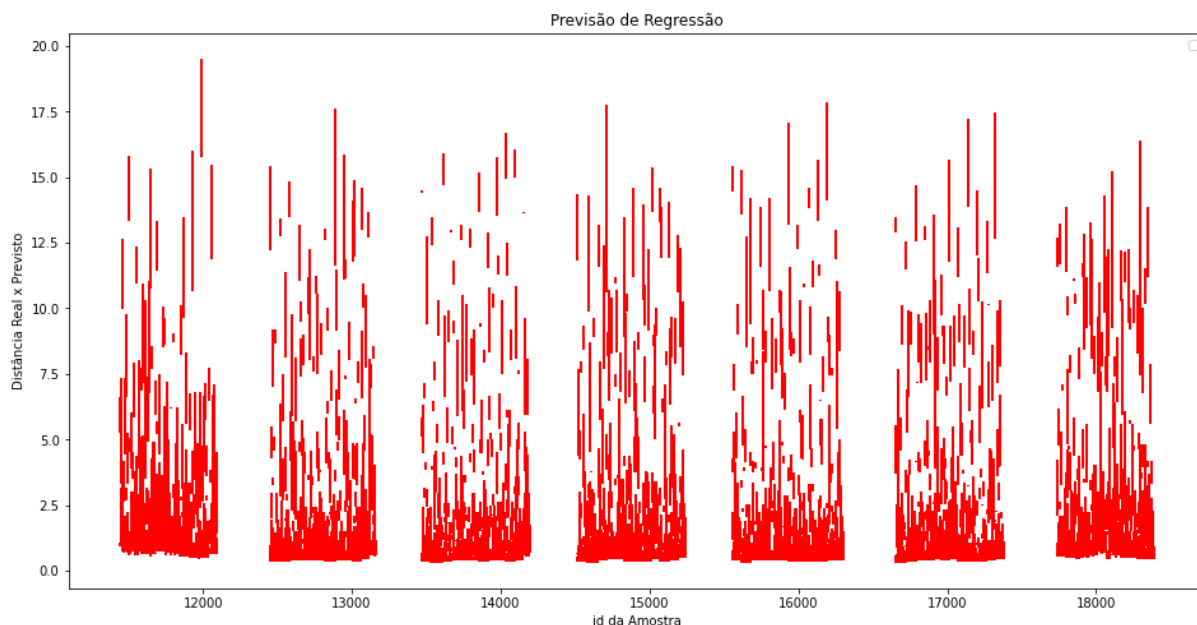


Figura 20- Distância entre Probabilidades Reais e Estimadas com Otimizador SGD em Fortaleza Horário "Comercial"

4.2.3. Cidade: Petrolina

Para Petrolina são detalhados, a seguir, somente os resultados obtidos com o algoritmo Nadam por terem sido bem superiores a resultados com outros algoritmos, inclusive o SGD.

4.2.3.1. Otimizador Nadam – Todas as horas do dia

A Tabela 16Tabela 10Tabela 8 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador NADAM, na cidade de Petrolina, considerando transportes iniciados em qualquer hora do dia. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 16- Parâmetros e Resultados da Melhor Simulação de Petrolina com Otimizador Nadam

Faixa Horária:	0 a 23 horas
Algoritmo Otimizador:	Nadam
Parâmetros do Algoritmo Otimizador:	Default
Rede Neural:	3 camadas densamente conectadas com 160, 80 e 1 neurônio respectivamente
Total de Épocas:	250
Indicador de Perda:	MAPE
Resultados:	RMSE: 2,3037 MSE: 5,3072

	MAE: 1,213514686
	MAPE: 43,98124695

Já o gráfico da Figura 21 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

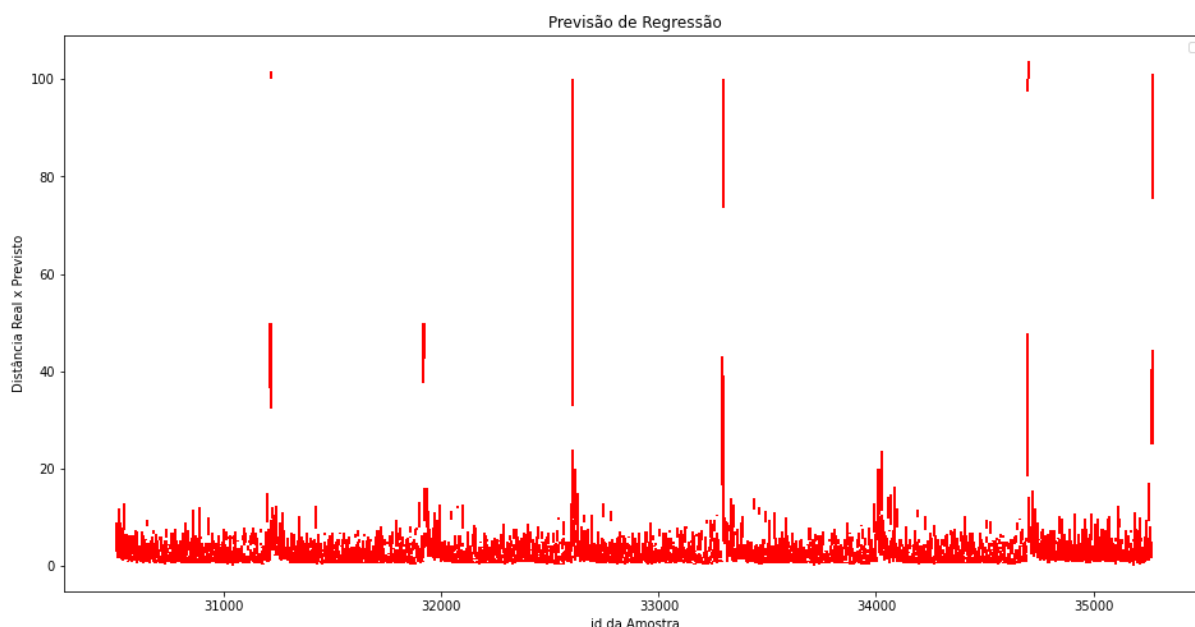


Figura 21- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Petrolina

4.2.3.2. Otimizador Nadam – Horário de 8 a 19 horas

A Tabela 17 apresenta os parâmetros da simulação na qual foram obtidos os melhores resultados utilizando o Otimizador Nadam, na cidade de Petrolina, considerando transportes iniciados entre 8 e 19 horas. Apresenta, também, os resultados dos indicadores de performance obtidos nessa simulação.

Tabela 17- Parâmetros e Resultados da Melhor Simulação de Petrolina com Otimizador Nadam (Horário "Comercial")

Faixa Horária:	8 a 19 horas
Algoritmo Otimizador:	Nadam
Parâmetros do Algoritmo Otimizador:	Default
Rede Neural:	3 camadas densamente conectadas com 160, 80 e 1 neurônio respectivamente
Total de Épocas:	500
Indicador de Perda:	MAPE
Resultados:	RMSE: 1,4478 MSE: 2,0961 MAE: 0,9989

MAPE: 44,50652313

Já o gráfico da Figura 22 apresenta, para cada elemento da base de teste, a distância entre a probabilidade real e a estimada pelo modelo nessa simulação.

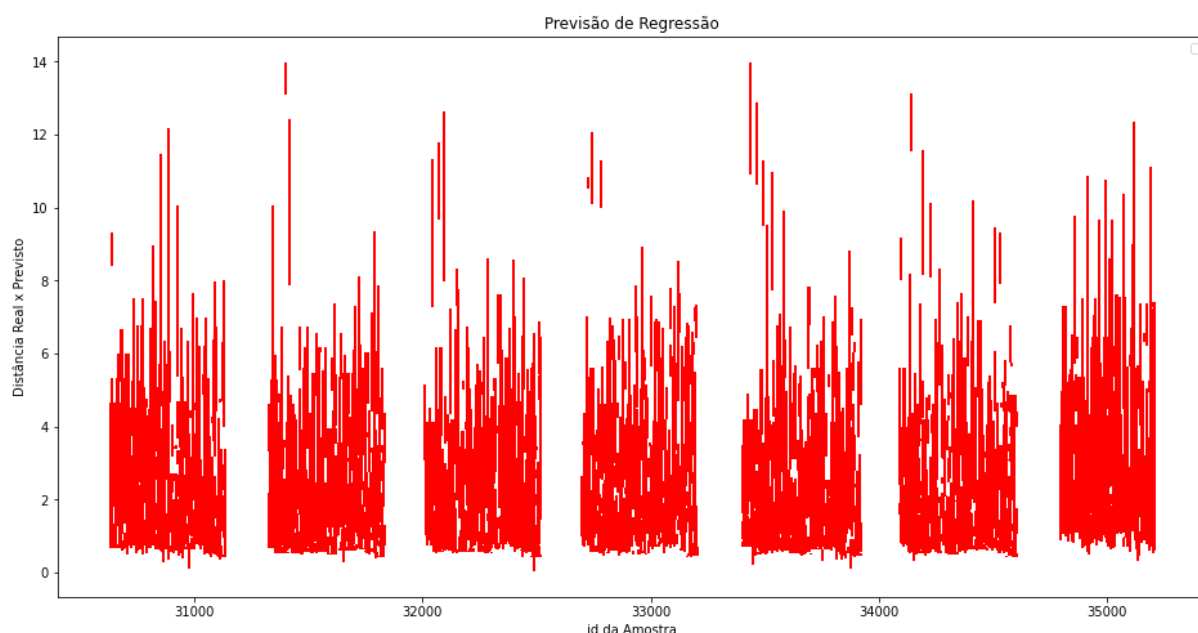


Figura 22- Distância entre Probabilidades Reais e Estimadas com Otimizador Nadam em Petrolina Horário "Comercial"

4.3. Análise da Ordem ("Ranking") de Bairros com Maior Probabilidade de Demandar Transporte

Os melhores modelos simulados apresentaram MAPE de 31,68%. Ou seja, as probabilidades estimadas distam, em média, da ordem de 31,68% das probabilidades reais. Independentemente dessa distância, a questão que cabe ser respondida para indicar a utilidade dos modelos é: as probabilidades previstas são capazes de indicar os bairros com maior probabilidade de originar transportes, em linha com o "ranking real" de bairros com maior probabilidade de originar transportes?

Para responder essa pergunta os bairros foram classificados em ordem decrescente de probabilidade real e prevista, na base de teste, para cada dia e hora cheia. Foi estabelecido o "ranking" para cada dia e hora cheia, de modo que o bairro com maior probabilidade recebeu o número 1 (primeiro), o bairro com segunda maior probabilidade recebeu o número 2 (segundo) e assim por diante.

De modo a ilustrar o realizado, a Tabela 18 apresenta, como exemplo, o "ranking" para Fortaleza no dia 15 para transportes iniciados na faixa de 7 horas da manhã para as probabilidades reais da base de testes.

Tabela 18- Ordem ("Ranking") dos Bairros com maior Probabilidade (real) de Solicitarem Transporte em Fortaleza, em 15/Janeiro às 7:00hs

nome_cidade_partida	dia_da_criacao	dia_semana_da_criacao	hora_da_criacao	bairro_partida	Probabilidade REAL	RANKING
Fortaleza	15	4	7	Aldeota	13,442623	1
Fortaleza	15	4	7	Conjunto Prefeito Jose Walter	8,5245902	2
Fortaleza	15	4	7	Meireles	7,5409836	3
Fortaleza	15	4	7	Bairro de Fatima	6,557377	4
Fortaleza	15	4	7	Centro	5,2459016	5
Fortaleza	15	4	7	Mondubim	4,2622951	6
Fortaleza	15	4	7	Dionisio Torres	3,2786885	7
Fortaleza	15	4	7	Joaquim Tavora	3,2786885	8
Fortaleza	15	4	7	Papicu	2,6229508	9
Fortaleza	15	4	7	Coco	2,295082	10

Já a Tabela 19 apresenta, como exemplo, o “*ranking*” para Fortaleza no dia 15 para transportes iniciados na faixa de 7 horas da manhã para as probabilidades estimadas pelo modelo para a base de testes.

Tabela 19- Ordem (“Ranking”) dos Bairros com maior Probabilidade (estimada) de Solicitarem Transporte em Fortaleza, em 15/Janeiro às 7:00hs

nome_cidade_partida	dia_da_criacao	dia_semana_da_criacao	hora_da_criacao	bairro_partida	Probabilidade PREVISTA	RANKING
Fortaleza	15	4	7	Aldeota	10,51550198	1
Fortaleza	15	4	7	Conjunto Prefeito Jose Walter	8,172930717	2
Fortaleza	15	4	7	Meireles	7,201272488	3
Fortaleza	15	4	7	Centro	4,353507996	4
Fortaleza	15	4	7	Bairro de Fatima	4,114582539	5
Fortaleza	15	4	7	Joaquim Tavora	2,678854227	6
Fortaleza	15	4	7	Mondubim	2,312119007	7
Fortaleza	15	4	7	Coco	2,051606655	8
Fortaleza	15	4	7	Papicu	1,704099059	9
Fortaleza	15	4	7	Dionisio Torres	1,627932549	10

Em seguida, foi realizada a correspondência dos “*rankings*” associados a probabilidade real e a probabilidade estimada para cada bairro, dia e horário, criando-se o indicador de “Distância” correspondendo a fórmula abaixo:

$$Distância = |Ranking_{real} - Ranking_{estimada}|$$

O resultado da criação do indicador de “Distância” está ilustrado na Tabela 20, a qual apresenta o resultado desse indicador para Fortaleza no dia 15 para transportes iniciados na faixa de 7 horas da manhã.

Tabela 20- Distância entre Ranking Real e Estimado dos Bairros com maior Probabilidade de Solicitarem Transporte em Fortaleza, em 15/Janeiro às 7:00hs

nome_cidade_partida	dia_da_criacao	dia_semana_da_criacao	hora_da_criacao	bairro_partida	Ranking REAL	Ranking PREVISTO	DISTÂNCIA
Fortaleza	15	4	7	Aldeota	1	1	0
Fortaleza	15	4	7	Conjunto Prefeito Jose Walter	2	2	0
Fortaleza	15	4	7	Meireles	3	3	0
Fortaleza	15	4	7	Bairro de Fatima	4	5	1
Fortaleza	15	4	7	Centro	5	4	1
Fortaleza	15	4	7	Mondubim	6	7	1
Fortaleza	15	4	7	Dionisio Torres	7	10	3
Fortaleza	15	4	7	Joaquim Tavora	8	6	2
Fortaleza	15	4	7	Papicu	9	9	0
Fortaleza	15	4	7	Coco	10	8	2

Como podemos observar na Tabela 20, para o bairro **Dionisio Torres**, a Distância calculada foi 3 pois esse bairro tem a 7ª maior probabilidade de originar transportes na base real (*ranking* 7) enquanto que, na probabilidade estimada, foi a 10ª maior (*ranking* 10). Já para os bairros **Aldeota**, **Conjunto Prefeito José Walter**, **Meireles** e **Papicu**, a Distância foi 0 pois esses bairros tiveram o mesmo *ranking* para probabilidades reais e probabilidades estimadas de originarem transportes.

A análise dessas distâncias e o resultado consolidado encontra-se detalhado nas seções a seguir.

4.3.1. Cidade: Fortaleza

Para a simulação com o otimizador Nadam, todas as horas do dia, descrito na seção 4.2.2.1, a distribuição por faixas de distância referente aos 10 Bairros com maior probabilidade real de originar transportes em relação aos *ranking* estimado é o apresentado na Tabela 21.

Tabela 21- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 10 Primeiros Bairros do Ranking

Faixa de Distâncias	% Faixa	% Acumulado
0	21%	21%
1 a 3	45%	66%
4 a 5	8%	74%
mais de 5	26%	100%

Ou seja:

- 21% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 66% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 74% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Se considerarmos somente os 5 bairros com maior probabilidade real de originar transportes, a distribuição por faixas de distância é a apresentada na Tabela 22.

Tabela 22- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 5 Primeiros Bairros do Ranking

Faixa de Distâncias	% Faixa	% Acumulado
0	31%	31%
1 a 3	49%	80%
4 a 5	5%	85%
mais de 5	15%	100%

Ou seja:

- 31% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 80% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 85% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Analisando um *ranking* maior, ou seja, os 20 bairros com maior probabilidade real de originar transportes, observamos também bons resultados quanto a distância em relação ao *ranking* estimado, como demonstrado na Tabela 23.

Tabela 23- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 20 Primeiros Bairros do Ranking

Faixa de Distâncias	% Faixa	% Acumulado
0	13%	13%
1 a 3	34%	47%
4 a 5	9%	56%
mais de 5	44%	100%

Ou seja:

- 56% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Já para a simulação com o otimizador Nadam, somente no período de 8hs às 19hs, descrito na seção 4.2.2.2, a distribuição por faixas de distância referente aos 10 Bairros com maior probabilidade real de originar transportes em relação aos *ranking* estimado é o apresentado na Tabela 24.

Tabela 24- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 10 Primeiros Bairros do Ranking (Horário "Comercial")

Faixa de Distâncias	% Faixa	% Acumulado
0	26%	26%
1 a 3	44%	70%
4 a 5	7%	77%
mais de 5	23%	100%

Ou seja:

- 26% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 70% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 77% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Se considerarmos somente os 5 bairros com maior probabilidade real de originar transportes, a distribuição por faixas de distância é a apresentada na Tabela 25.

Tabela 25- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 5 Primeiros Bairros do Ranking (Horário "Comercial")

Faixa de Distâncias	% Faixa	% Acumulado
0	39%	39%
1 a 3	45%	84%
4 a 5	2%	86%
mais de 5	14%	100%

Ou seja:

- 39% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 84% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 86% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Analisando um *ranking* maior, ou seja, os 20 bairros com maior probabilidade real de originar transportes, observamos também bons resultados quanto a distância em relação ao *ranking* estimado, como demonstrado na Tabela 26.

Tabela 26- Distribuição das Distâncias de Ranking por Faixa para Fortaleza, Otimizador Nadam, 20 Primeiros Bairros do Ranking (Horário "Comercial")

Faixa de Distâncias	% Faixa	% Acumulado
0	15%	15%
1 a 3	33%	47%
4 a 5	10%	58%
mais de 5	42%	100%

Ou seja:

- 58% dos bairros tem distância de até 5 posições em relação ao *ranking* real

4.3.2. Cidade: Patos de Minas

Para a simulação com o otimizador Nadam, todas as horas do dia, descrito na seção 4.2.1.1, a distribuição por faixas de distância referente aos 10 Bairros com maior probabilidade real de originar transportes em relação aos *ranking* estimado é o apresentado na Tabela 27.

Tabela 27- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 10 Primeiros Bairros do Ranking

Faixa de Distâncias	% Faixa	% Acumulado
0	13%	13%
1 a 3	23%	36%
4 a 5	9%	45%
mais de 5	55%	100%

Ou seja:

- 13% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 36% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 45% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Se considerarmos somente os 5 bairros com maior probabilidade real de originar transportes, a distribuição por faixas de distância é a apresentada na Tabela 28.

Tabela 28- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 5 Primeiros Bairros do Ranking

Faixa de Distâncias	% Faixa	% Acumulado
0	23%	23%
1 a 3	25%	48%
4 a 5	7%	55%
mais de 5	45%	100%

Ou seja:

- 23% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 48% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 55% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Já para a simulação com o otimizador Nadam, somente no período de 8hs às 19hs, descrito na seção 4.2.1.2, a distribuição por faixas de distância referente aos 10 Bairros com maior probabilidade real de originar transportes em relação aos *ranking* estimado é o apresentado na Tabela 29.

Tabela 29- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 10 Primeiros Bairros do Ranking (Horário "Comercial")

Faixa de Distâncias	% Faixa	% Acumulado
0	16%	16%
1 a 3	26%	42%
4 a 5	10%	52%
mais de 5	48%	100%

Ou seja:

- 16% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 42% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 52% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Se considerarmos somente os 5 bairros com maior probabilidade real de originar transportes, a distribuição por faixas de distância é a apresentada na Tabela 30.

Tabela 30- Distribuição das Distâncias de Ranking por Faixa para Patos de Minas, Otimizador Nadam, 5 Primeiros Bairros do Ranking (Horário "Comercial")

Faixa de Distâncias	% Faixa	% Acumulado
0	28%	28%
1 a 3	31%	59%
4 a 5	8%	67%
mais de 5	33%	100%

Ou seja:

- 28% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 59% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 67% dos bairros tem distância de até 5 posições em relação ao *ranking* real

4.3.3. Cidade: Petrolina

Para a simulação com o otimizador Nadam, todas as horas do dia, descrito na seção 4.2.3.1, a distribuição por faixas de distância referente aos 10 Bairros com maior probabilidade real de originar transportes em relação aos *ranking* estimado é o apresentado na Tabela 31.

Tabela 31- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 10 Primeiros Bairros do Ranking

Faixa de Distâncias	% Faixa	% Acumulado
0	10%	10%
1 a 3	35%	45%
4 a 5	14%	59%
mais de 5	41%	100%

Ou seja:

- 10% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 45% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 59% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Se considerarmos somente os 5 bairros com maior probabilidade real de originar transportes, a distribuição por faixas de distância é a apresentada na Tabela 32.

Tabela 32- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 5 Primeiros Bairros do Ranking

Faixa de Distâncias	% Faixa	% Acumulado
0	16%	16%
1 a 3	39%	55%
4 a 5	10%	65%
mais de 5	35%	100%

Ou seja:

- 16% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 55% dos bairros tem distância de até 3 posições em relação ao *ranking* real

- 65% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Já para a simulação com o otimizador Nadam, somente no período de 8hs às 19hs, descrito na seção 4.2.3.2, a distribuição por faixas de distância referente aos 10 Bairros com maior probabilidade real de originar transportes em relação aos *ranking* estimado é o apresentado na Tabela 33.

Tabela 33- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 10 Primeiros Bairros do Ranking (Horário "Comercial")

Faixa de Distâncias	% Faixa	% Acumulado
0	10%	10%
1 a 3	35%	46%
4 a 5	18%	63%
mais de 5	37%	100%

Ou seja:

- 10% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 46% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 63% dos bairros tem distância de até 5 posições em relação ao *ranking* real

Se considerarmos somente os 5 bairros com maior probabilidade real de originar transportes, a distribuição por faixas de distância é a apresentada na Tabela 34.

Tabela 34- Distribuição das Distâncias de Ranking por Faixa para Petrolina, Otimizador Nadam, 5 Primeiros Bairros do Ranking (Horário "Comercial")

Faixa de Distâncias	% Faixa	% Acumulado
0	16%	16%
1 a 3	39%	55%
4 a 5	13%	68%
mais de 5	32%	100%

Ou seja:

- 16% dos bairros foram previstos na posição correta do *ranking* (Distância 0)
- 55% dos bairros tem distância de até 3 posições em relação ao *ranking* real
- 68% dos bairros tem distância de até 5 posições em relação ao *ranking* real

5. CONCLUSÕES E TRABALHOS FUTUROS

O objetivo deste trabalho foi criar modelo preditivo capaz de indicar os bairros com maior probabilidade de demandarem transportes, a cada hora, de modo a orientar o posicionamento dos transportadores por aplicativo.

Conforme demonstrado na seção 4.2 os resultados foram satisfatórios. Já a seção 4.3 demonstra a proximidade do “*ranking*” de principais bairros demandadores de transporte e o “*ranking*” estimado pelo modelo. Entendo, portanto, que os objetivos do trabalho foram cumpridos.

Em trabalhos futuros as seguintes oportunidades de melhoria poderiam ser exploradas visando aprimoramento dos modelos e de seus resultados:

1. Período da Base de Dados de Transportes

A base de transportes dispõe de informações do mês de janeiro/2020 apenas. Os modelos preditivos foram treinados a partir de simulações com essas bases. Assim, entendo que os modelos em questão são capazes de efetuar previsões referentes a janeiro/2020. Não é possível afirmar que esse modelo é adequado para efetuar previsões de outros meses, ou mesmo que será adequado para efetuar previsões em relação a meses de janeiro de outros anos, como por exemplo, janeiro/2021.

Serie interessante dispor de base de pelo menos 2 anos de transportes realizados de modo a viabilizar que o modelo possa inferir:

- O comportamento de outros meses do ano. Cada mês pode ter sazonalidades específicas. Sem dispor de informações de outros meses para treinamento do modelo, não podemos afirmar que o mesmo estaria capaz de efetuar previsões de todos os meses do ano
- O padrão de comportamento de um mês ao longo dos anos. Importante que o modelo disponha de informações de mais de um mês de janeiro. Somente assim poderíamos avaliar se o modelo estaria habilitado a prever os meses de janeiro independente do ano em questão. O mesmo vale para os demais meses do ano
- Sazonalidades específicas tais como o comportamento do modelo em feriados, festividades e eventos locais, dia das mães, dia dos pais, dia dos namorados etc.

2. Padronização de Nomes de Bairros

A oportunidade de padronização informada na seção 3.2.3 foi identificada quando as simulações para a cidade de Fortaleza foram executadas. Há oportunidade semelhante para as demais cidades selecionadas para treinamento. Cabe avaliar se resultados melhores poderiam ser obtidos com a aplicação do mesmo tipo de padronização nas demais cidades.

3. Agrupamento de Bairros por Regiões

A quantidade de bairros nas cidades eleitas, bem como, nas cidades em geral da base de transportes é muito grande, superando em pouco a quantidade de variáveis preditoras do modelo.

Fortaleza possui 251 bairros contra 314 variáveis preditoras, após criação das *Dummy Variables*.

Patos de Minas possui 118 bairros na base contra 176 variáveis preditoras, após criação das *Dummy Variables*.

Petrolina possui 108 bairros na base contra 160 variáveis preditoras, após criação das *Dummy Variables*.

A baixa granularidade dos bairros torna o modelo menos assertivo. Ao mesmo tempo, tal nível de detalhe pode ser desnecessário para os objetivos do negócio.

Em trabalhos futuros seria recomendável o estudo das zonas e áreas das cidades de modo a agrupar os bairros por zonas (Norte, Sul, Leste, Oeste, Centro) ou áreas (área

Portuária, Universitária, Industrial, Comercial, Residencial etc). A base de treino passaria a dispor dessa informação e o modelo seria treinado para prever a probabilidade por Região da cidade e não bairro a bairro

4. Correlação com outros fatores externos (Condições Climáticas)

Fatores externos, tais como, as condições climáticas, talvez afetem a probabilidade de bairros ou regiões originarem transportes. É esperado, por exemplo, que em dias de chuva, ou de baixa temperatura, haja menos demandas de transportes envolvendo regiões praianas.

Seria interessante, portanto, em trabalhos futuros, incluir informações das condições climáticas como variáveis preditoras na base de treino e teste, bem como outras informações pertinentes.

5. COVID-19

A base de dados utilizada para treino dos modelos é de janeiro/2020, antes do início do isolamento social em geral no Brasil. Após 6 meses do início do isolamento, várias medidas de flexibilização já foram adotadas, no entanto paradigmas relacionados a ineficiência e ineficácia do home-office e ensino a distância foram quebrados. Costumes foram alterados com as pessoas fazendo mais uso de serviços de “delivery”, compras pela internet e outros. Áreas das cidades antes ocupadas principalmente por escritórios comerciais permanecem com pouca circulação de pessoas mesmo com a flexibilização do isolamento. Por tudo isso é possível que o perfil de probabilidades de regiões das cidades originarem transportes seja diferente agora do que era antes do início do isolamento.

Trabalhos futuros devem levar em consideração também, dados recentes, visando o aprendizado pelos modelos das probabilidades após o evento do isolamento, do COVID-19 e dos novos costumes da população.

Referências Bibliográficas

OLIVEIRA, Evandro. **O mercado de aplicativos de transporte no Brasil**. *Voz do Pará*, Pará, 01 de fevereiro de 2020.

Disponível em :<<http://vozdopara.com.br/o-mercado-de-aplicativos-de-transporte-no-brasil-%EF%BB%BF/#:~:text=O%20mercado%20de%20aplicativos%20de%20transporte%20no%20Brasil%20tem%20se.solicitaram%20um%20transporte%20por%20aplicativo.&text=Em%202018%2C%20esse%20nC3%BAmero%20era%20de%2064%25%20dos%20usu%C3%A1rios%20brasileiros.>>

Acessado em: 12 de agosto de 2020

CARDIM, Maria Eduarda. **Número de motoristas por aplicativo cresceu 136% de 2012 a 2019**.

Correio Braziliense, Brasília, 23 de fevereiro de 2020.

Disponível em :

<https://www.correiobraziliense.com.br/app/noticia/economia/2020/02/23/internas_economia,829826/numero-de-motoristas-por-aplicativo-cresceu-136-de-2012-a-2019.shtml>

Acessado em: 12 de agosto de 2020

CRUZ, Elaine Patrícia. **Mais de 1,5 mil motoristas de aplicativos já são microempreendedores**.

Agência Brasil, São Paulo, 30 de agosto de 2019.

Disponível em :< <https://agenciabrasil.ebc.com.br/economia/noticia/2019-08/mais-de-15-mil-motoristas-de-aplicativos-ja-se-registraram-como-meio-viv#:~:text=Existem%20hoje%20no%20Brasil%20cerca,metade%20deles%20tornem%20se%20microempreendedores>>

Acessado em: 12 de agosto de 2020

MANZONI Jr., Ralphe. **Aplicativo 99 torna-se, oficialmente, primeiro unicórnio brasileiro**. *Isto É*, São Paulo, 2 de janeiro de 2018.

Disponível em :< <https://www.istoedinheiro.com.br/aplicativo-99-torna-se-oficialmente-primeiro-unicornio-brasileiro/>>

Acessado em: 12 de agosto de 2020

Kingma et al., 2015. **Adam: A Method for Stochastic Optimization**. Conference paper na 3rd International Conference for Learning Representations, San Diego, 2015

Apêndice A – Código fonte do programa de inferência

28/09/2020

TCC-Gaudium-SemanaTeste - Limp0.ipynb - Colaboratory

▼ Inicialização

```
#para conectar ao Drive
from google.colab import drive # Carrega biblioteca para montar e carregar drive
drive.mount('/content/drive') # Esse código pedirá autenticação
```

Mounted at /content/drive

```
import os
os.chdir("/content/drive/My Drive/TCC")
```

```
ls
#https://colab.research.google.com/drive/1MPBU7kXUIbZcyR8jMJgobImvmQ5TGWAA
```

BreastCancer.csv
Caruaru-SOFTMAX.csv
Corridas_Cidades_Eleitas-FINAL-Acentuado-Probabilidades-ComSemanaTeste.csv
Corridas_Cidades_Eleitas-FINAL-Acentuado-Probabilidades.csv
Corridas_Cidades_Eleitas-fINAL.csv
Corridas_Cidades_Fortaleza-fINAL-Acentuado.csv
Corridas_Fortaleza-FINAL-Acentuado-Probabilidades-ComSemanaTeste.csv
Corridas_TESTE_SOFTMAX.csv
Corumba-SOFTMAX.csv
kddCupNovo.txt
kddCup.txt
PatosDeMinas-SOFTMAX.csv
Petrolina-SOFTMAX.csv
Petrolina-SOFTMAX-Regioes.csv
'Petrolina-SOFTMAX - Relevantes.csv'
predictions_training_test.pdf
predictions_training_test.svg
Resultados_Semana_Testes.csv
Resultados_Semana_Testes.gsheet
Santarem-SOFTMAX.csv

▼ Importacao de bibliotecas

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

▼ Carga da Base e Seleção da Cidade de Partida

```
#Retirar do Comentário a Cidade de Partida
#cidade_partida = 'Patos de Minas'
cidade_partida = 'Fortaleza'
```

<https://colab.research.google.com/drive/1a7Im8bkWPgHU5a-xDs3SJH6KtBKl3Zz5#scrollTo=QyKc1qZWu9pM&printMode=true>

1/10

```

#cidade_partida = 'Santarem'
#cidade_partida = 'Petroлина'

if (cidade_partida == 'Fortaleza'):
    df1 = pd.read_csv('Corridas_Fortaleza-FINAL-Acentuado-Probabilidades-ComSemanaTeste.csv')
else:
    df1 = pd.read_csv('Corridas_Cidades_Eleitas-FINAL-Acentuado-Probabilidades-ComSemanaTeste.csv')

df1=df1[df1['nome_cidade_partida'] == cidade_partida]

### TESTANDO SÓ PERÍODO DO HORÁRIO COMERCIAL
# Se for rodar simulação somente para um período do dia, descomentar a linha abaixo
df1 = df1[(df1['hora_da_criacao'] >= 8) & (df1['hora_da_criacao'] <= 19)]

# colocar id como nome de linha
df1 = df1.set_index('id')
df1.head(1000)


```

id	nome_cidade_partida	dia_da_criacao	dia_semana_da_criacao	hora_da_criacao	b
241	Fortaleza	1	4	8	
242	Fortaleza	1	4	8	
243	Fortaleza	1	4	8	
244	Fortaleza	1	4	8	
245	Fortaleza	1	4	8	
...
1519	Fortaleza	2	5	15	
1520	Fortaleza	2	5	15	
1521	Fortaleza	2	5	15	
1522	Fortaleza	2	5	15	
1523	Fortaleza	2	5	15	

1000 rows × 6 columns

```

#dimensões da base
df1.shape

(42510, 6)

### MULTIPLICA POR 100 A PROBABILIDADE PARA DIMINUIR PROBLEMAS DE CÁLCULOS COM VALORES PEQ
df1['PROBABILIDADE'] = df1['PROBABILIDADE']*100
df1['PROBABILIDADE'].head()

```

```

id
241    4.761905
242    1.587302
243    1.587302
244    1.587302
245    1.587302
Name: PROBABILIDADE, dtype: float64

```

▼ Converte Variáveis para DUMMY

▼ Converte Variáveis Numéricas para String

```

# Essa conversão é necessária para permitir que no passo seguinte sejam convertidas para D
# Foram convertidas para DUMMY pois são categóricas
df1['dia_semana_da_criacao'] = df1['dia_semana_da_criacao'].astype(str)
df1['dia_da_criacao'] = df1['dia_da_criacao'].astype(str)
df1['hora_da_criacao'] = df1['hora_da_criacao'].astype(str)

```

▼ Conversão para DUMMY

```

df1 = pd.get_dummies(df1)
df1.head()

```

```

PROBABILIDADE  nome_cidade_partida_Fortaleza  dia_da_criacao_1  dia_da_criacao_
id
241          4.761905                                1          1
242          1.587302                                1          1
243          1.587302                                1          1
244          1.587302                                1          1
245          1.587302                                1          1
5 rows x 303 columns

```

▼ Separa em Base de Treino e Base de Teste

▼ Separa a Semana de Teste

```

# Criei uma semana de teste na base de entrada com todas as combinações
# necessário de treino, dia da semana e horário para extrair as estimativas
https://colab.research.google.com/drive/1a7lm8bKWPgHUSa-xDs3SjH6K1BK13Zz5#scrollTo=QyKc1qZWu9pM&printMode=true

```

3/10

28/09/2020

TCC-Gaudium-SemanaTeste - Limpo.ipynb - Colaboratory

```
# POSSÍVEIS DE OBITOS, dia da semana e horário para extrair as estimativas
# do modelo para essa semana
# A semana corresponde ao mesmo período da Base de Teste, ou seja,
# 12 a 18 de Janeiro
```

```
#SEPARA A SEMANA DE TESTE
```

```
if cidade_partida == 'Petrolina':
    df_semana_teste = df1.loc[200000:299999] #PETROLINA
    df1.loc[200000:299999]
    df1 = df1.loc[:199999]
elif cidade_partida == 'Patos de Minas':
    df_semana_teste = df1.loc[100000:199999] #Patos de MINAS
    df1.loc[100000:199999]
    df1 = df1.loc[:99999]
elif cidade_partida == 'Fortaleza':
    df_semana_teste = df1.loc[300000:399999] #Fortaleza
    df1.loc[300000:399999]
    df1 = df1.loc[:299999]
```

```
# EXCLUI A COLUNA PROBABILIDADE (TODA ZERADA) DA SEMANA DE TESTE
# Não precisamos dessa informação pois essa será a variável a ser prevista
# pelo modelo
df_semana_teste.head()
df_semana_teste=df_semana_teste.drop(columns=['PROBABILIDADE'])
df_semana_teste.head()
```



```
nome_cidade_partida_Fortaleza dia_da_criacao_1 dia_da_criacao_10 dia_da_c
```

id			
300008	1	0	0
300009	1	0	0
300010	1	0	0
300011	1	0	0
300012	1	0	0

5 rows × 302 columns

▼ Criação de Base de Teste "Aleatória"

▼ Indicar a semente inicial e para divisão da base em treino e teste

```
## NÃO usei essa solução no treino do MODELO
## Usei a opção seguinte de base de teste direcionada para o período de
## 12 a 18 de Janeiro
import random
```

<https://colab.research.google.com/drive/1a7im8bkWPgHU5a-xDs3SjH6K1BK13Zz5#scrollTo=QyKc1qZWu9pM&printMode=true>

4/10


```
np.random.seed(0) #semente inicial
nlinhas = df1.shape[0]
nlinhas
```

21426

```
from sklearn.model_selection import train_test_split
#Divide a base em treino e teste. A coluna 'PROBABILIDADE' é o Label. Test_size diz o tama
#x_train, x_test, y_train, y_test = train_test_split(df1.drop(columns=['PROBABILIDADE']),
#                                                    df1['PROBABILIDADE'], test_size=0.3)
#x_train.head()
```

▼ Criação de Base de Teste "Direcionada" (Semana de 12 a 18 / Janeiro)

```
## VERSÃO DIRECIONADA PARA TREINAR COM MÊS TODO EXCETO SEMANA DE TESTE
```

```
x_test = df1[(df1['dia_da_criacao_12'] == 1)|(df1['dia_da_criacao_13'] == 1)|(df1['dia_da_
y_test = x_test['PROBABILIDADE']
x_test = x_test.drop(columns=['PROBABILIDADE'])
```

```
x_train = df1[(df1['dia_da_criacao_12'] == 0)&(df1['dia_da_criacao_13'] == 0)&(df1['dia_da_
y_train = x_train['PROBABILIDADE']
x_train = x_train.drop(columns=['PROBABILIDADE'])
x_train
y_train
```

```
id
241    4.761905
242    1.587302
243    1.587302
244    1.587302
245    1.587302
...
31427   0.442478
31428   0.442478
31429   0.442478
31430   0.884956
31431   0.442478
Name: PROBABILIDADE, Length: 16470, dtype: float64
```

▼ Rede Neural

▼ Converte Dataframe de Pandas para Numpy para inserir na RN

```
## Conversão necessária para inserir dados nas funções da RN
X_train_normalized = x_train.to_numpy()
X_test_normalized = x_test.to_numpy()
df_semana_teste_normalized = df_semana_teste.to_numpy()
```

▼ Importa bibliotecas necessárias para uso na RN

```
import tensorflow as tf
from keras import Model, Sequential
from keras.layers import Dense
from keras.optimizers import SGD, RMSprop, Adam, Adamax, Adagrad, Adadelta, Nadam
```

```
X_train_normalized.shape[1:]
```

```
↳ (302,)
```

▼ Cria a Rede Neural

```
#Inicia a rede
RN = Sequential()
# Cria primeira camada e 'input_shape' entradas
RN.add(Dense(314, input_shape = X_train_normalized.shape[1:], activation = 'relu')) # antes
#RN.add(Dropout(0.2))

RN.add(Dense(157, activation = 'relu'))
#RN.add(Dense(40, activation = 'relu'))
#RN.add(Dense(22, activation = 'relu'))
#RN.add(Dense(50, activation = 'sigmoid'))
#RN.add(dropout(0.2))
#RN.add(Dense(4, activation = 'sigmoid'))
RN.add(Dense(1, activation = 'relu'))
# linha do programa original --> Cria segunda e última camada 2 células (número de classes)
# RN.add(Dense(NumberOfClasses, activation = 'sigmoid'))

# Diferença Sigmoidal X Softmax
# https://www.google.com/search?q=sigmoid+vs+softmax&oq=sigmoid&aqs=chrome.3.69i57j35i39j0

RN.summary()
```

```
↳
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 314)	95142
dense_1 (Dense)	(None, 157)	48455

▼ Treina a Rede Neural

```
Total params: 144,355

# treinamento

#Otimização por Gradiente Descendente (SGD)
#otimizador = SGD(lr=0.001, decay=1e-6, momentum=0.9) #decay=1e-6,
#otimizador = RMSprop()
#otimizador = SGD()
#otimizador = Adam()
otimizador = Nadam()

#Configura o modelo para treinamento
#RN.compile(optimizer = sgd, loss = 'mean_squared_error', metrics = ['accuracy', 'mean_squ
RN.compile(optimizer = otimizador, loss = 'mean_absolute_percentage_error', #tf.keras.loss
            metrics = ['mean_absolute_error', #tf.keras.metrics.RootMeanSquaredError(),
                      'mean_squared_error', 'mean_absolute_percentage_error', 'accuracy' ])

# Treina o modelo para um certo numero de epocas
trainedRN = RN.fit(X_train_normalized,y_train, epochs = 350, verbose = 0)
```

▼ Métricas de Avaliação


▼ Métricas Calculadas Automaticamente pelo Keras

```
trainedRN.model.metrics_names

['loss',
 'mean_absolute_error',
 'mean_squared_error',
 'mean_absolute_percentage_error',
 'accuracy']

import math
score = trainedRN.model.evaluate(X_test_normalized, y_test, verbose = 0)
print('Test score (loss):', score[0])
print('Test MAE:', score[1])
print('Test RMSE:', math.sqrt(score[2]))
print('Test MSE:', score[2])
```

```
print('MAPE:', score[3])
print('Test Accuracy:', score[4])
```


 Test score (loss): 34.15443420410156
 Test MAE: 0.5490463972091675
 Test RMSE: 0.962030920685685
 Test MSE: 0.9255034923553467
 MAPE: 34.15443420410156
 Test Accuracy: 0.0014124293811619282

▼ Métricas Calculadas por mim para Validação do Resultado do Keras

```
# Previsão
y_test_predicted = RN.predict(X_test_normalized)

#import math
from sklearn.metrics import mean_squared_error
rmse = math.sqrt(mean_squared_error(y_test, y_test_predicted))
mse = mean_squared_error(y_test, y_test_predicted)
mape = np.mean(np.abs((y_test.to_numpy() - y_test_predicted.transpose())/y_test.to_numpy()))
mae = np.mean(np.abs((y_test.to_numpy() - y_test_predicted.transpose())))

print('MAE: ',mae)
print('RMSE: ', rmse)
print('MSE: ',mse)
print('MAPE: ',mape)
```

 MAE: 0.5490463541090214
 RMSE: 0.9620308013503334
 MSE: 0.9255032627467645
 MAPE: 34.154424609962994

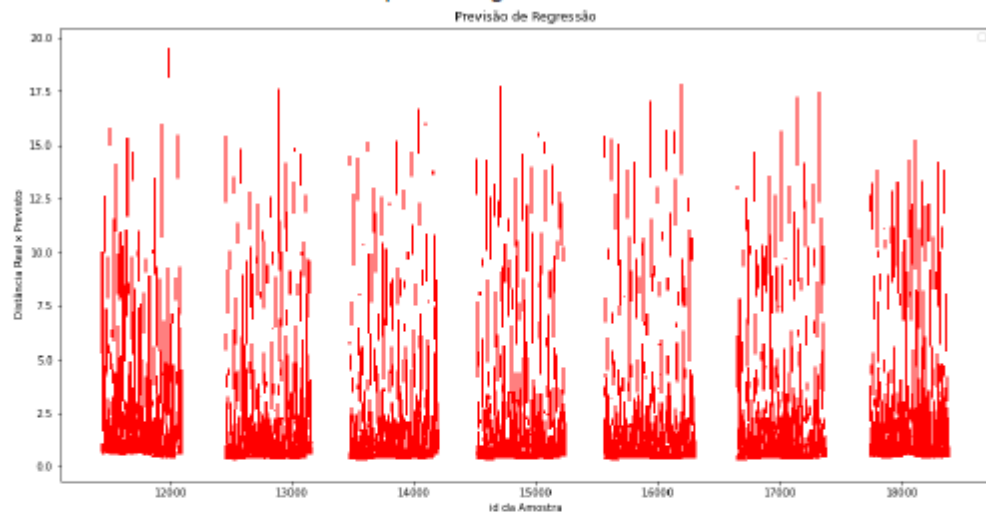
Apresentação Gráfica da Distância entre Probabilidade Real e Estimada

```
plt.figure(figsize=(16, 8))
plt.vlines(y_test.index, y_test_predicted,y_test.values, color='red', linewidth=2)

plt.title('Previsão de Regressão')
plt.xlabel('id da Amostra')
plt.ylabel('Distância Real x Previsto')
plt.legend()
plt.savefig('predictions_training_test.pdf',format = 'pdf')
plt.show()
```



No handles with labels found to put in legend.



▼ Faz a previsão da Semana de Testes e Grava resultados para Análise

```
y_semana_teste = RN.predict(df_semana_teste_normalized)
df_semana_teste.insert(loc=df_semana_teste.shape[1], column='PROBABILIDADE', value=y_semana_teste)
df_semana_teste.shape[1]
df_semana_teste.head()
df_semana_teste.to_csv(path_or_buf = 'Resultados_Semana_Testes.csv', sep=';', decimal = ',',
```

Apêndice B – Relação completa das simulações realizadas

Camadas																			
1	2	3	4	5	Cidade	Épocas	Otimizador	Parâmetros Otimizador				Indicadores							
Neurônios	Neurônios	Neurônios	Neurônios	Neurônios				Loss	Learning Rate	Decay	Momentum	RMSE	MSE	MAE	MAPE				
160	80	1			Petrolina	500	SGD	MAE	0,1	1,00E-06	0,9	3,5012	12,26	1,48	62,2				
160	80	1			Petrolina	5.000	SGD	MAE	0,1	1,00E-06	0,9	5,1329	26,35	1,58	65,29				
160	80	1			Petrolina	2.500	SGD	MAE	0,1	1,00E-06	0,9	3,925	15,41	1,575	67,47				
160	40	1			Petrolina	5.000	SGD	MAE	0,1	1,00E-06	0,9	4,9359	24,36	1,635	63,07				
160	120	1			Petrolina	5.000	SGD	MAE	0,1	1,00E-06	0,9	3,8152	14,56	1,605	70,51				
160	80	1			Petrolina	10.000	SGD	MAE	0,1	1,00E-06	0,9	4,7929	22,97	1,661	70,66				
160	80	1			Petrolina	10.000	SGD	MAPE	0,1	1,00E-06	0,9	6,3179	39,92	3,048	100				
160	80	1			Petrolina	10.000	RMSProp	MSE	0,001	0,00E+00	0	2,966	8,797	1,484	72,92				
160	80	1			Petrolina	5.000	RMSProp	MSE	0,001	0,00E+00	0	2,9671	8,804	1,489	73,68				
160	80	1			Petrolina	10.000	Adam	MSE	0,001	0,00E+00	0	3,0342	9,207	1,485	70,72				
160	80	1			Petrolina	5.000	Adam	MSE	0,001	0,00E+00	0	2,9709	8,826	1,48	69,77				
160	80	1			Petrolina	30.000	Adam	MSE	0,001	0,00E+00	0	2,9545	8,729	1,484	71,06				
160	80	1			Petrolina	5.000	Adam	MAPE	0,001	0,00E+00	0	2,6987	7,283	1,374	58,53				
160	80	1			Petrolina	10.000	Adam	MAPE	0,001	0,00E+00	0	2,7834	7,747	1,401	60,21				
160	80	1			Petrolina	2.500	Adam	MAPE	0,001	0,00E+00	0	2,7856	7,76	1,4	59,61				
160	80	1			Petrolina	5.000	RMSProp	MAPE	0,001	0,00E+00	0	2,874	8,26	1,48	63,71				
160	80	1			Petrolina	10.000	RMSProp	MAPE	0,001	0,00E+00	0	2,9007	8,414	1,521	64,28				
160	80	1			Petrolina	7.500	RMSProp	MAPE	0,001	0,00E+00	0	5,5674	31	1,709	63,9				
160	80	1			Petrolina	7.500	RMSProp	MAPE	0,001	0,00E+00	0	5,6395	31,8	1,731	53,91				
160	80	20	1		Petrolina	7.500	Adam	MAPE	0,001	0,00E+00	0	2,4657	6,08	1,336	57,24				
160	80	40	1		Petrolina	7.500	Adam	MAPE	0,001	0,00E+00	0	2,8707	8,241	1,342	55,35				
160	80	40	1		Petrolina	15.000	Adam	MAPE	0,001	0,00E+00	0	2,8794	8,291	1,387	58,19				
160	100	20	1		Petrolina	7.500	Adam	MAPE	0,001	0,00E+00	0	2,8897	8,351	1,374	57,52				
160	80	20	1		Petrolina	7.500	Adam	MSE	0,001	0,00E+00	0	2,9373	8,628	1,445	65,98				
160	80	1			Petrolina	7.500	Adam	MAE	0,001	0,00E+00	0	2,9159	8,503	1,401	62,81				
112	60	1			Corumba	7.500	Adam	MAE	0,001	0,00E+00	0	6,1013	37,23	3,263	55,1				
132	70	1			Caruaru	7.500	Adam	MAE	0,001	0,00E+00	0	6,4269	41,3	3,445	62,02				
160	80	1			Petrolina	10.000	Adam	MAE	0,001	0,00E+00	0	2,6177	6,852	1,408	63,97				
160	80	1			Petrolina	5.000	Adam	MAE	0,001	0,00E+00	0	2,6787	7,176	1,417	64,86				
160	80	1			Petrolina	10.000	Adam	MAE	0,001	0,00E+00	0	2,6062	6,792	1,4	63,84				
160	80	1			Petrolina	20.000	Adam	MAE	0,001	0,00E+00	0	2,6062	6,792	1,4	63,84				
160	80	1			Petrolina	20.000	Adam	MAE	0,001	0,00E+00	0	3,5765	12,79	1,532	62,28				
160	80	1			Petrolina	10.000	Adam	MAPE	0,001	0,00E+00	0	2,9866	8,92	1,466	45,81				
160	80	1			Petrolina	5.000	Adam	MAPE	0,001	0,00E+00	0	2,6221	6,875	1,324	46,24				
160	80	1			Petrolina	2.500	Adam	MAPE	0,001	0,00E+00	0	2,7149	7,371	1,327	46,84				
160	80	1			Petrolina	1.000	Adam	MAPE	0,001	0,00E+00	0	2,4734	6,118	1,269	46,01				
160	80	1			Petrolina	500	Adam	MAPE	0,001	0,00E+00	0	2,516	6,33	1,236	46,12				
160	80	1			Petrolina	750	Adam	MAPE	0,001	0,00E+00	0	2,3626	5,582	1,255	46,69				
160	80	1			Petrolina	250	Adam	MAPE	0,001	0,00E+00	0	2,4377	5,942	1,221	45,56				
160	80	1			Petrolina	100	Adam	MAPE	0,001	0,00E+00	0	2,5056	6,278	1,244	43,5				
160	80	1			Petrolina	500	RMSProp	MAPE	0,001	0,00E+00	0	4,592	2,109	2,62	93,06				
160	80	1			Petrolina	1.000	RMSProp	MAPE	0,001	0,00E+00	0	4,357	18,98	2,575	89				
160	80	1			Petrolina	750	SGD	MAPE	0,001	0,00E+00	0	4,3006	18,49	1,419	52,49				
160	80	1			Petrolina	2.500	SGD	MAPE	0,001	0,00E+00	0	3,7744	14,25	1,447	64,68				
160	60	20	1		Petrolina	750	Adam	MAPE	0,001	0,00E+00	0	2,4	5,76	1,244	45,27				
160	80	1			Petrolina	750	Adam	MSE	0,001	0,00E+00	0	3,6915	13,63	1,729	83,42				
160	80	1			Petrolina	750	Adam	MAE	0,001	0,00E+00	0	3,2494	10,56	1,404	62,37				
160	80	1			Petrolina	750	SGD	MAE	0,1	1,00E-06	0,9	2,7313	7,46	1,353	61,36				
160	80	1			Petrolina	250	SGD	MAE	0,1	1,00E-06	0,9	2,8136	7,916	1,305	60,18				
160	80	1			Petrolina	500	SGD	MAE	0,1	1,00E-06	0,9	3,03	9,181	1,299	52,1				
160	80	1			Petrolina	500	SGD	MAPE	0,001	1,00E-06	0,9	4,278	18,3	1,345	48,78				
160	80	1			Petrolina	500	SGD	MAPE	0,001	0,00E+00	0	4,3654	19,06	1,532	76,34				
160	80	1			Petrolina	500	SGD	MAPE	0,001	1,00E-06	0,5	4,2605	18,15	1,403	56,73				
160	80	1			Petrolina	500	SGD	MAPE	0,001	Não espec	0,9	4,3086	1,856	1,392	51,18				
160	80	1			Petrolina	500	SGD	MAPE	0,001	1,00E-03	0,9	4,3894	19,27	1,553	60,93				
160	80	1			Petrolina	500	SGD	MAPE	0,001	1,00E-09	0,9	4,3348	18,79	1,38	50,02				
160	80	1			Petrolina	500	Adamax	MAPE	0,001	0,00E+00	0	3,5311	12,47	2,016	58,09				
160	80	1			Petrolina	500	Adagrad	MAPE	0,001	0,00E+00	0	2,8141	7,919	1,552	65,63				
160	80	1			Petrolina	500	Adadelta	MAPE	0,001	0,00E+00	0	5,1835	26,87	2,43	67,15				
160	80	1			Petrolina	500	Nadam	MAPE	0,001	0,00E+00	0	2,5869	6,692	1,242	44,05				
160	80	1			Petrolina	250	Nadam	MAPE	0,001	0,00E+00	0	2,3037	5,307	1,214	43,98				
160	80	1			Petrolina	100	Nadam	MAPE	0,001	0,00E+00	0	2,4898	6,199	1,236	43,57				

Camadas																			
1	2	3	4	5															
Neurônios	Neurônios	Neurônios	Neurônios	Neurônios	Cidade	Épocas	Otimizador	Parâmetros Otimizador				Indicadores							
								LOSS	Learning Rate	Decay	Momentum	RMSE	MSE	MAE	MAPE				
160	80	1			Petrolina	750	Nadam	MAPE	0,001	0,00E+00	0	2,4743	6,122	1,264	44,93				
160	80	1			Petrolina	750	Nadam	MAPE	0,001	0,00E+00	0	2,4797	6,149	1,229	45,06				
160	80	1			Petrolina	250	Nadam	MAPE	0,001	0,00E+00	0	2,2918	5,252	1,213	45,1				
160	60	20	1		Petrolina	250	Nadam	MAPE	0,001	0,00E+00	0	2,3319	5,438	1,206	45,32				
160	80	40	20	1	Petrolina	250	Nadam	MAPE	0,001	0,00E+00	0	2,4627	6,065	1,358	43,9				
160	80	1			Petrolina	250	Adam	MAPE	0,001	0,00E+00	0	2,3272	5,416	1,201	46,1				
160	60	20	1		Petrolina	250	Adam	MAPE	0,001	0,00E+00	0	2,4871	6,186	1,249	45,37				
160	80	1			Petrolina	250	Adam	MAPE	0,001	0,00E+00	0	2,3861	5,694	1,3	54,74				
176	80	1			Patos de Minas	250	Adam	MAPE	0,001	0,00E+00	0	2,3323	5,44	1,114	44,09				
176	80	1			Patos de Minas	250	Nadam	MAPE	0,001	0,00E+00	0	2,2029	4,853	1,104	44,22				
176	88	1			Patos de Minas	1.000	Nadam	MAPE	0,001	0,00E+00	0	2,3079	5,327	1,153	44,44				
176	88	1			Patos de Minas	500	Nadam	MAPE	0,001	0,00E+00	0	2,2397	5,016	1,131	45,14				
176	88	1			Patos de Minas	100	Nadam	MAPE	0,001	0,00E+00	0	2,2675	5,142	1,106	43,85				
176	88	1			Patos de Minas	50	Nadam	MAPE	0,001	0,00E+00	0	2,2541	5,081	1,086	41,79				
176	88	1			Patos de Minas	25	Nadam	MAPE	0,001	0,00E+00	0	2,3142	5,356	1,091	40,27				
176	88	1			Patos de Minas	10	Nadam	MAPE	0,001	0,00E+00	0	2,3573	5,557	1,119	39,23				
176	88	1			Patos de Minas	250	SGD	MAPE	0,001	0,00E+00	0	2,162	4,674	1,078	50,3				
176	88	1			Patos de Minas	500	SGD	MAPE	0,001	0,00E+00	0	2,2656	5,133	1,219	54,34				
176	88	1			Patos de Minas	100	SGD	MAPE	0,001	0,00E+00	0	2,4419	5,963	1,429	64,11				
176	88	1			Patos de Minas	250	RMSProp	MAPE	0,001	0,00E+00	0	4,3153	18,62	1,992	59,94				
176	88	1			Patos de Minas	100	RMSProp	MAPE	0,001	0,00E+00	0	4,343	1,886	2,009	59,26				
176	88	1			Patos de Minas	250	SGD	MAPE	0,001	0,00E+00	0	2,3218	5,391	1,126	49,58				
176	66	22	1		Patos de Minas	250	SGD	MAPE	0,001	0,00E+00	0	2,0895	4,366	1,1	55,9				
176	66	22	1		Patos de Minas	250	Nadam	MAPE	0,001	0,00E+00	0	2,1981	4,832	1,103	44,84				
176	88	1			Patos de Minas	200	SGD	MAPE	0,001	0,00E+00	0	2,1508	4,626	1,1	59,77				
176	88	1			Patos de Minas	300	SGD	MAPE	0,001	0,00E+00	0	2,8005	7,843	1,371	55,97				
176	88	1			Patos de Minas	275	SGD	MAPE	0,001	0,00E+00	0	2,1927	4,808	1,123	48,87				
314	157	1			Fortaleza	500	Nadam	MAPE	0,001	0,00E+00	0	1,7181	2,952	0,829	35				
314	157	1			Fortaleza	250	Nadam	MAPE	0,001	0,00E+00	0	1,7262	2,98	0,836	34,99				
314	157	1			Fortaleza	100	Nadam	MAPE	0,001	0,00E+00	0	1,6613	2,76	0,803	34,22				
314	157	1			Fortaleza	50	Nadam	MAPE	0,001	0,00E+00	0	1,6598	2,755	0,801	33,54				
314	157	1			Fortaleza	25	Nadam	MAPE	0,001	0,00E+00	0	1,6622	2,763	0,8	32,46				
314	157	1			Fortaleza	10	Nadam	MAPE	0,001	0,00E+00	0	1,6736	2,801	0,798	31,68				
314	157	1			Fortaleza	5	Nadam	MAPE	0,001	0,00E+00	0	1,7481	3,056	0,832	32,04				
314	157	1			Fortaleza	250	SGD	MAPE	0,001	0,00E+00	0				100				
314	157	1			Fortaleza	250	SGD	MAPE	0,001	1,00E-06	0,9	1,6331	2,667	0,81	37,11				
314	157	1			Fortaleza	100	SGD	MAPE	0,001	1,00E-06	0,9	1,672	2,795	0,812	34,79				
314	157	1			Fortaleza	500	SGD	MAPE	0,001	1,00E-06	0,9	1,6796	2,821	0,824	37,34				
314	157	1			Fortaleza	250	RMSProp	MAPE	0,001	0,00E+00	0	3,8394	14,74	1,905	53,54				
314	157	1			Fortaleza	100	RMSProp	MAPE	0,001	0,00E+00	0	3,7717	14,23	1,855	52,35				
314	157	1			Fortaleza	250	Adam	MAPE	0,001	0,00E+00	0	1,6869	2,846	0,819	34,26				
314	157	1			Fortaleza	500	Adam	MAPE	0,001	0,00E+00	0	1,7124	2,932	0,826	34,72				
314	157	1			Fortaleza	100	Adam	MAPE	0,001	0,00E+00	0	1,7229	2,968	0,83	34,13				
314	157	1			Fortaleza	250	Nadam	MAPE	0,001	0,00E+00	0	1,7011	2,894	0,817	34,13				
314	157	1			Fortaleza	250	Nadam	MAPE	0,001	0,00E+00	0	1,1395	1,298	0,636	35,35				

Simulações Realizadas considerando somente a faixa de “Horário Comercial” (8 às 19horas):

Camadas																			
1	2	3	4	5															
Neurônios	Neurônios	Neurônios	Neurônios	Neurônios	Cidade	Épocas	Otimizador	Parâmetros Otimizador				Indicadores							
								LOSS	Learning Rate	Decay	Momentum	RMSE	MSE	MAE	MAPE				
314	157	1			Fortaleza	250	Nadam	MAPE	0,001	0,00E+00	0	0,9873	0,975	0,566	34,28				
314	157	1			Fortaleza	500	Nadam	MAPE	0,001	0,00E+00	0	0,9628	0,927	0,552	34,56				
314	157	1			Fortaleza	350	Nadam	MAPE	0,001	0,00E+00	0	0,9447	0,892	0,54	34,2				
302	150	1			Fortaleza	350	Nadam	MAPE	0,001	0,00E+00	0	0,9682	0,937	0,556	34,96				
314	157	1			Fortaleza	300	Nadam	MAPE	0,001	0,00E+00	0	0,9498	0,902	0,54	33,73				
314	157	1			Fortaleza	325	Nadam	MAPE	0,001	0,00E+00	0	0,9463	0,895	0,546	34,79				
314	157	1			Fortaleza	350	Nadam	MAE	0,001	0,00E+00	0	0,9214	0,849	0,539	37,86				
176	88	1			Patos de Minas	250	SGD	MAPE	0,001	0,00E+00	0	1,3636	1,86	0,856	55,01				
176	88	1			Patos de Minas	50	Nadam	MAPE	0,001	0,00E+00	0	1,2777	1,632	0,823	43,17				
314	157	1			Fortaleza	250	SGD	MAPE	0,001	1,00E-06	0,9	0,981	0,962	0,567	36,71				
302	150	1			Fortaleza	250	SGD	MAPE	0,001	1,00E-06	0,9	0,981	0,975	0,574	36,68				
400	200	1			Fortaleza	250	SGD	MAPE	0,001	1,00E-06	0,9	0,8996	0,809	0,563	49,35				
163	80	1			Patos de Minas	250	SGD	MAPE	0,001	0,00E+00	0	1,3558	1,838	0,993	85,94				
163	80	1			Patos de Minas	50	Nadam	MAPE	0,001	0,00E+00	0	1,3367	1,787	0,839	41,99				
160	80	1			Petrolina	250	Nadam	MAPE	0,001	0,00E+00	0	1,4665	2,151	1,004	43,45				
160	80	1			Petrolina	100	Nadam	MAPE	0,001	0,00E+00	0	1,4675	2,154	1,013	44,38				
160	80	1			Petrolina	500	Nadam	MAPE	0,001	0,00E+00	0	1,4478	2,096	0,999	44,51				
160	80	1			Petrolina	50	Nadam	MAPE	0,001	0,00E+00	0	1,4918	2,226	1,026	42,67				
160	80	1			Petrolina	1.000	Nadam	MAPE	0,001	0,00E+00	0	1,4851	2,206	1,022	45,25				
160	80	1			Petrolina	750	Nadam	MAPE	0,001	0,00E+00	0	1,4659	2,1488	1,008	44,79				