PREVENDO DEMANDA DE TRANSPORTES PRIVADOS COM LSTM (RNN)

Redes Neurais para Previsão de Requisições de Táxis utilizando NYC Open Data

Projeto de Inovação e Aplicação de Inteligência Artificial feito por:

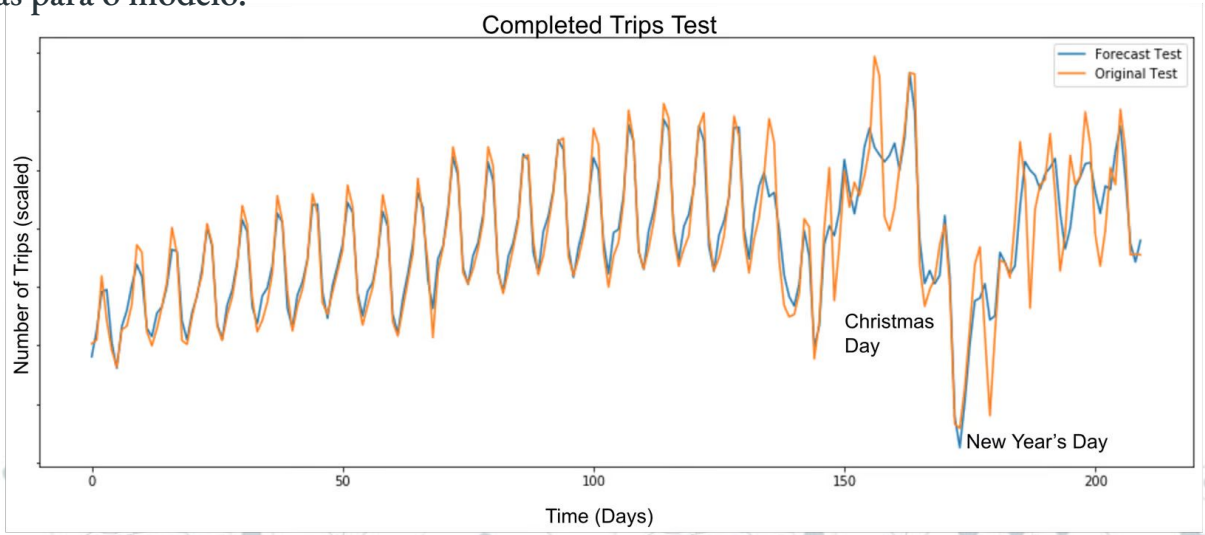
Samuel França da Costa Pedrosa

linkedin.com/in/samuelfrancapedrosa

github.com/sampedrosa

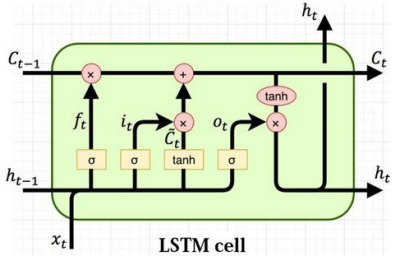
OBJETIVO

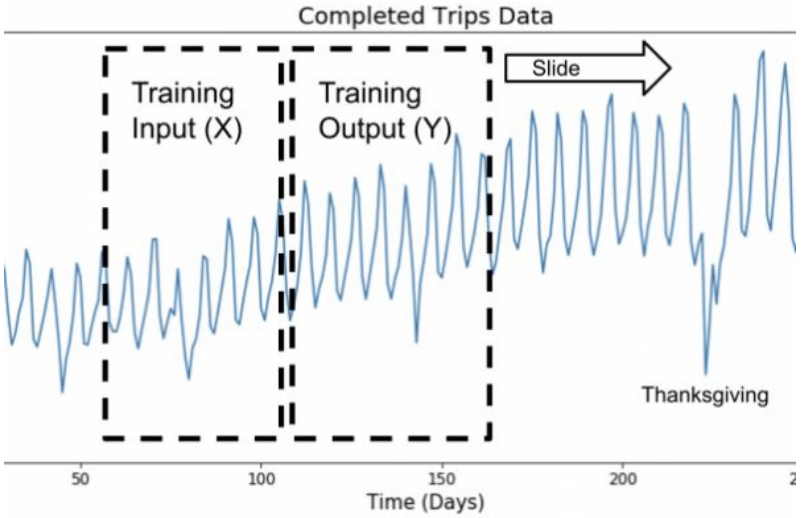
- Esse projeto tem como função utilizar ferramentas e métodos da área de inteligência artificial, mais especificamente a rede neural **LSTM**, para previsão de demandas em certas localizações de transporte privado.
- Como demonstração da aplicabilidade, é utilizado a base de dados de viagens de táxi coletada e disponibilizado pela prefeitura de Nova York para a **NYC Open Data**.
- Ou seja, os resultados esperados são de obter o dataset, conseguir transformá-lo em uma **Série-Temporal** baseada na demanda em função do tempo, aplicar o LSTM e obter as previsões para datas futuras ou não informadas para o modelo.



LSTM (LONG SHORT-TERM MEMORY)

- É uma Rede Neural Recorrente que guarda informações do processamento neural durante várias execuções.
- Idealmente aplicável em features e targets seriais, geralmente vinculados ao tempo.
- Possui uma **Entrada (X)** de um período de tempo anterior e **Saída (Y)** de um período geralmente menor do tempo posterior.





BASE DE DADOS

NYC Open Data

- O NYC Open Data é um conjunto de datasets abertos gerados por agências e parcerias da prefeitura de Nova York com intuito de manter transparência e acessibilidade das informações.

- Utiliza os serviços da **Socrata** que é uma empresa B2G de plataformas de base de dados. Possui uma API própria capaz de obter acesso aos dados utilizando chamadas otimizadas de CRUD e formatos específicos.

```
from sodapy import Socrata

client = Socrata("data.cityofnewyork.us", None)
client.timeout = 100
results = pd.DataFrame(client.get("biws-g3hs", select="Coluna", where="Coluna >= Valor", order="Coluna ASC"))
```

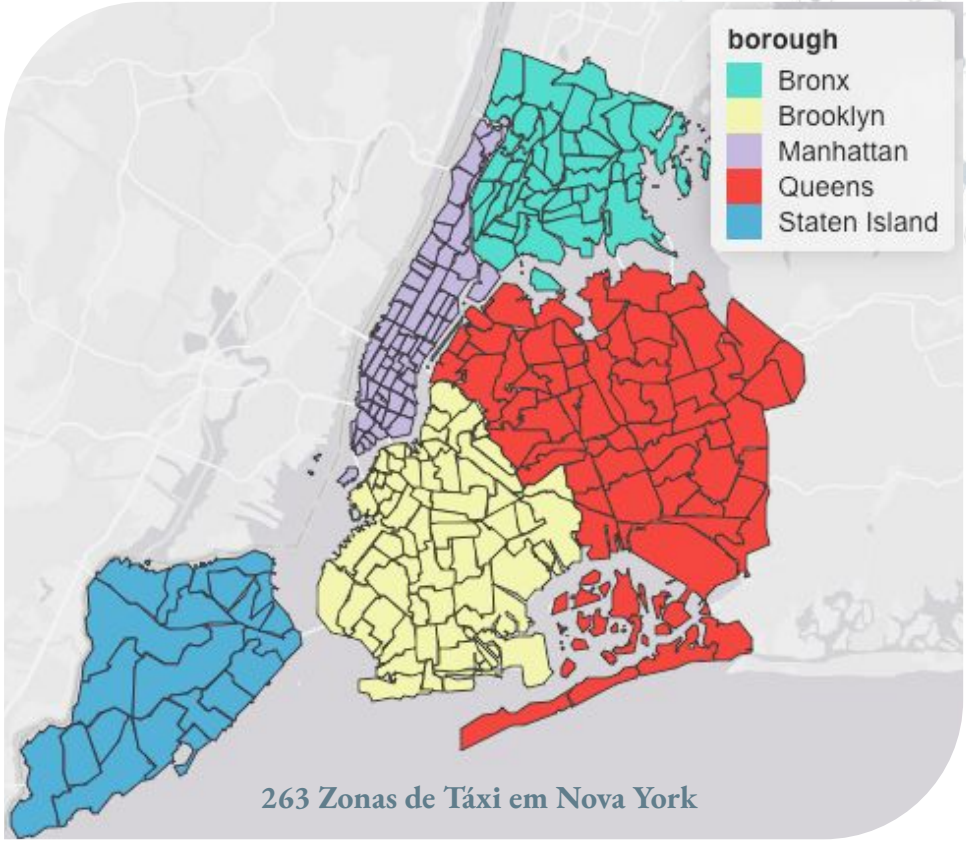
BASE DE DADOS

• Yellow Taxi Trip Data

Uma das bases de dados do NYC Open Data é o Yellow Taxi Trip Data que é basicamente os registros de todas as viagens de táxi que ocorreram na cidade de nova york desde 2009.

Contém diversas informações como a data, horário e localização do embarque e desembarque, distância percorrida, valor, gorjeta e etc.

Nesse projeto, o foco é analisar apenas a data-horário e localização das viagens, então é coletado apenas essas duas colunas. Além disso é obtido os registros das **263 Zonas de Táxi** e seu **GeoJSON** para plotagem do mapa, disponibilizado também pela NYC Open Data.



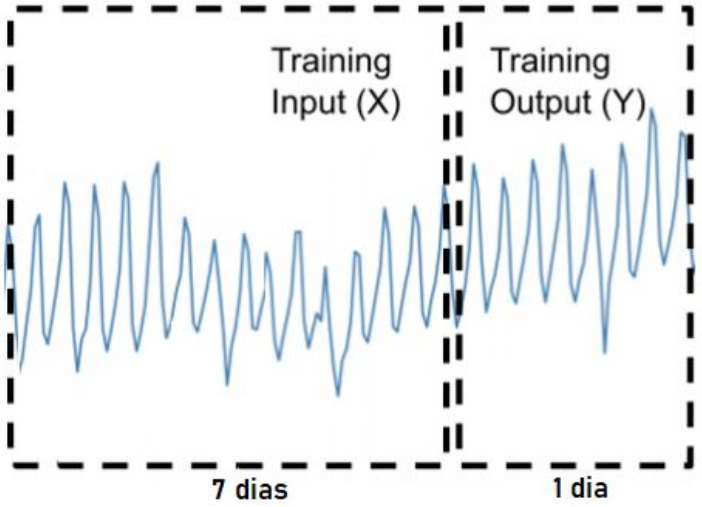
263 Zonas de Táxi em Nova York

FORMATAÇÃO DOS DADOS

Como o nível de processamento é limitado e o objetivo do projeto é apenas demonstrativo, será utilizado apenas o dataset de **2019** com **7 dias de Entrada (INPUT) e 1 dia de Saída (OUTPUT).**

A validação do teste será feita com os as entradas e saídas de teste que serão os últimos períodos da base de dados.

Ou seja, o objetivo é:
Prever um Dia baseado na Semana anterior.



TRATAMENTO DE DADOS

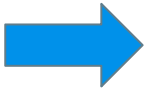
Para a aplicação do dataset, é necessário um tratamento de dados:

1. Padronização das Datas.
2. Excluir datas erradas ou nulas.
3. Transformar as datas em um Tempo Percorrido.
4. Definir uma constante de **Período** de Tempo.
5. Definir as **Requisições** com agrupamento pelo Período.

```
PERIOD = 15 # Group of PERIOD Minutes
DAY = int(24*60/PERIOD)
WEEK = int(7*DAY)
YEAR = int(365*DAY)
```

Período (PERIOD) é a unidade mínima em minutos do tempo para agrupamento, é inversamente proporcional ao tempo de processamento do LSTM.

	Date	Time
32	2019-01-01 00:00:00	0
33	2019-01-01 00:00:11	0
34	2019-01-01 00:00:37	0
35	2019-01-01 00:01:19	0
36	2019-01-01 00:02:38	0
...	...	...
3641655	2019-12-31 23:56:17	35040



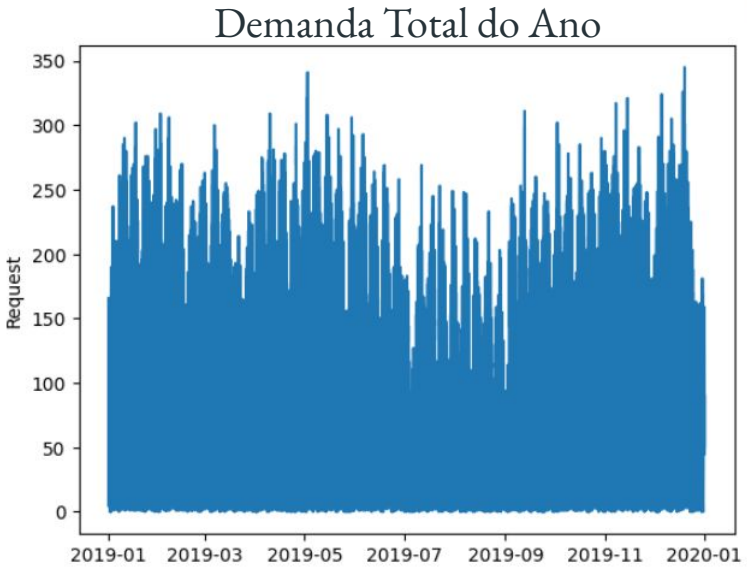
	Request
0	19
1	117
2	152
3	143
4	166
...	...
35036	94

ANÁLISE DOS DADOS

Os dados podem ser analisados de várias formas, o mais intuitivo é observar o comportamento **diário, semanal e mensal**, uma vez que os dados são temporais.

Os dados visualizados separadamente por **zonas vão adotar comportamentos diferentes**, como por exemplo uma zona de muita movimentação em Manhattan comparada com uma zona mais calma de Queens ou mesmo uma zona afastada de Staten Island com poucas corridas de táxi.

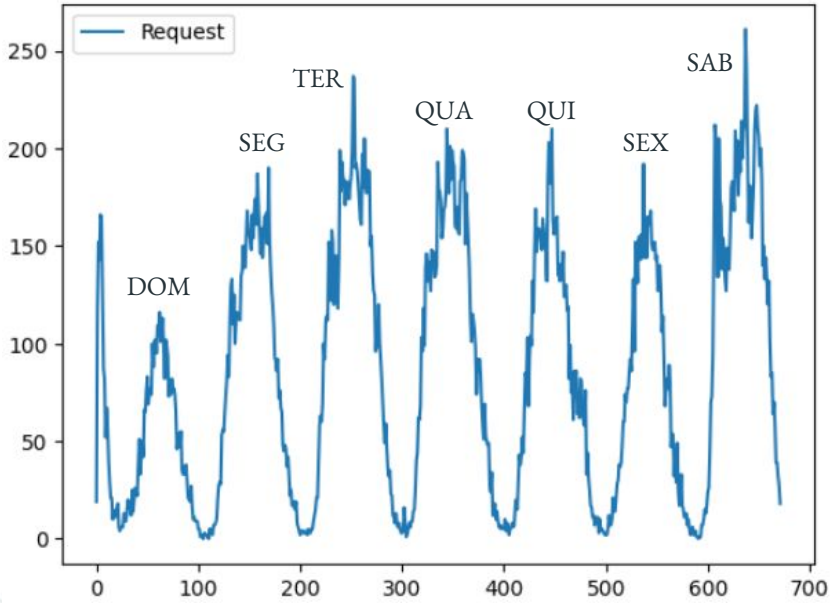
Os **picos ou comportamentos especiais** são importantes de se manter, pois podem ser características de **feriados, datas comemorativas ou até acidentes.**



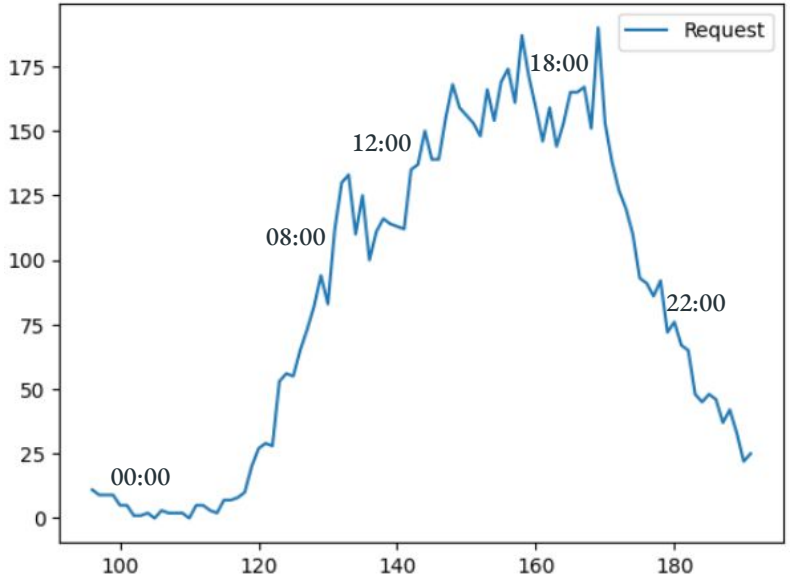
ANÁLISE DOS DADOS

Demanda em uma Zona de Manhattan

Semana



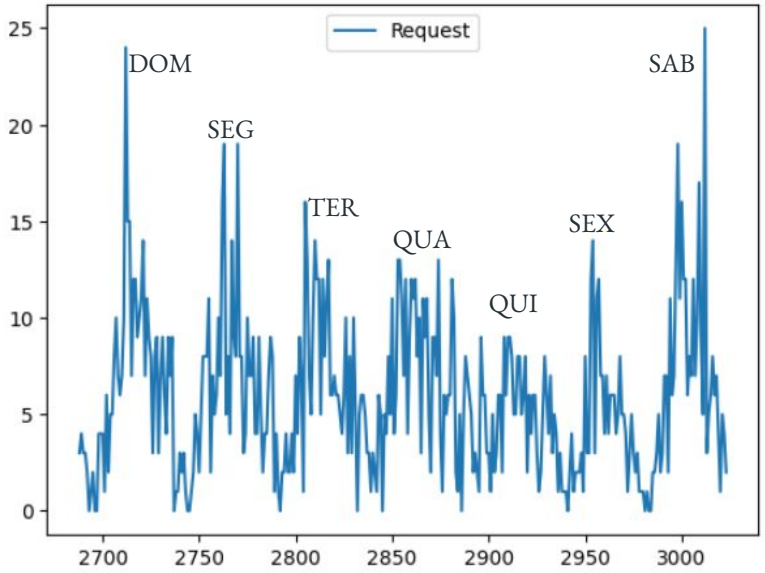
Dia



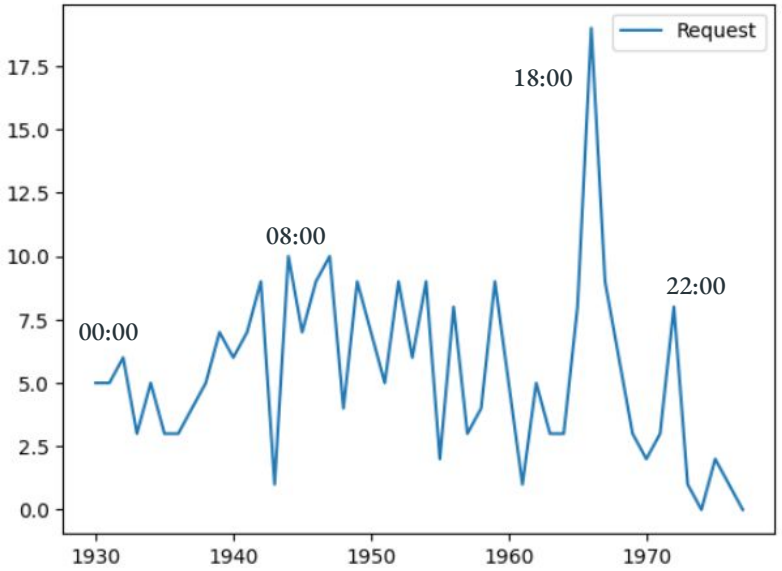
ANÁLISE DOS DADOS

Demanda em uma Zona de Queens

Semana



Dia



FORMATAÇÃO DO MODELO

O método LSTM utilizado foi o do **TensorFlow Keras** juntamente com todo seu kit de redes neurais. Basicamente a arquitetura da RNN consiste de uma camada de entrada, o processamento de longa memória de curto-prazo e duas camadas de saída.

O modelo possui os seguintes **Hiperparâmetros:**

```
TRAIN_PTG = 0.83  # Percentage of the Data for Training
NEURONS = 32  # Units of Neurons
EPOCHS = 4  # Number os Epochs
BATCH = 1  # Batchs per Execution
LEARN = 0.0001  # Learning Rate
```

```
model = Sequential()
model.add(InputLayer((WEEK,1)))
model.add(LSTM(NEURONS))
model.add(Dense(int(DAY*8), 'relu'))
model.add(Dense(DAY, 'linear'))
model.compile(loss=MeanSquaredError(), optimizer=Adam(learning_rate=LEARN), metrics=[RootMeanSquaredError()])
model.fit(X_train, y_train, epochs=EPOCHS, batch_size=BATCH, verbose=1)
```

TREINAMENTO DO MODELO

Após a criação do modelo, basta treiná-lo aplicando as métricas para validação e depois conferir os resultados das previsões.

O grande **desafio é o tempo de processamento**, uma vez que dependendo dos hiperparâmetros passados, a compilação de todas as zonas pode demorar dias.

Os resultados do treinamento do modelo de todas as zonas é **mais eficiente se compilado em um serviço de nuvem** como AWS e Azure. Entretanto é possível obter resultados interessantes em uma mesma máquina caso haja paciência.

- Cálculo da quantidade de valores que serão recebidos pela rede neural como entrada:

$$\text{VALORES DE ENTRADA} = \frac{\text{INPUT} \cdot 200 \cdot 10^9}{(\text{PERIOD})^2}$$

INP/PER	01 (min)	15 (min)	30 (min)
03 (dias)	600 bi	2,6 bi	0,6 bi
07 (dias)	1,4 tri	6,2 bi	1.5 bi
15 (dias)	3 tri	13,3 bi	3.3 bi

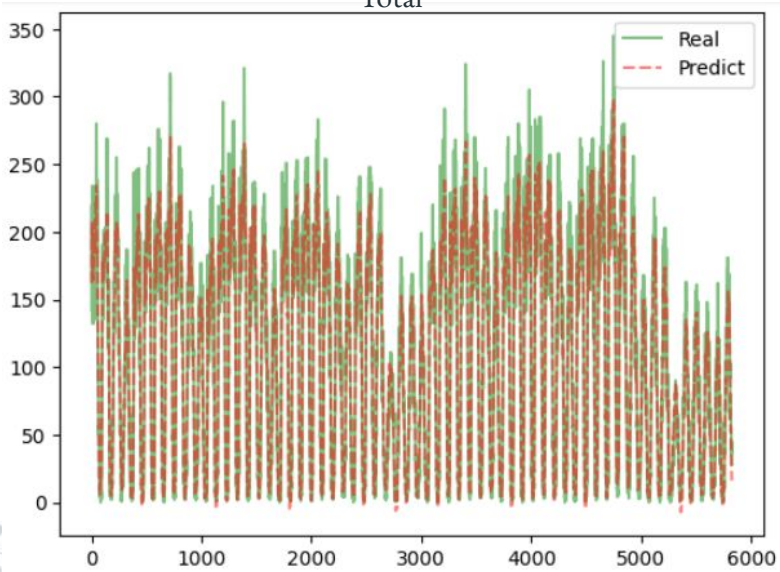
VALIDAÇÃO DO MODELO

Para validação do modelo LSTM, é testado com uma zona (237) para validar os resultados e depois compilar com o restante das zonas.

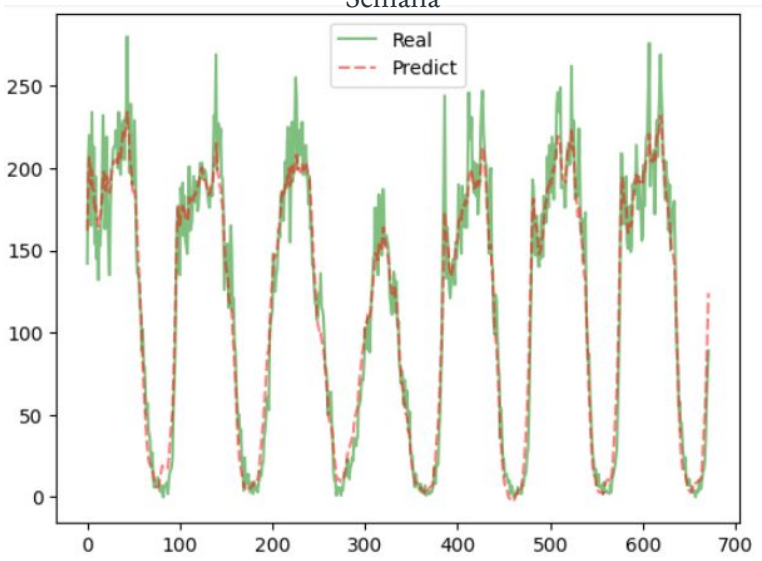
Utilizando:

- Período: 15
- Porcentagem de Treino: 15%
- Unidades de Neurônios: 32
- Epochs: 4

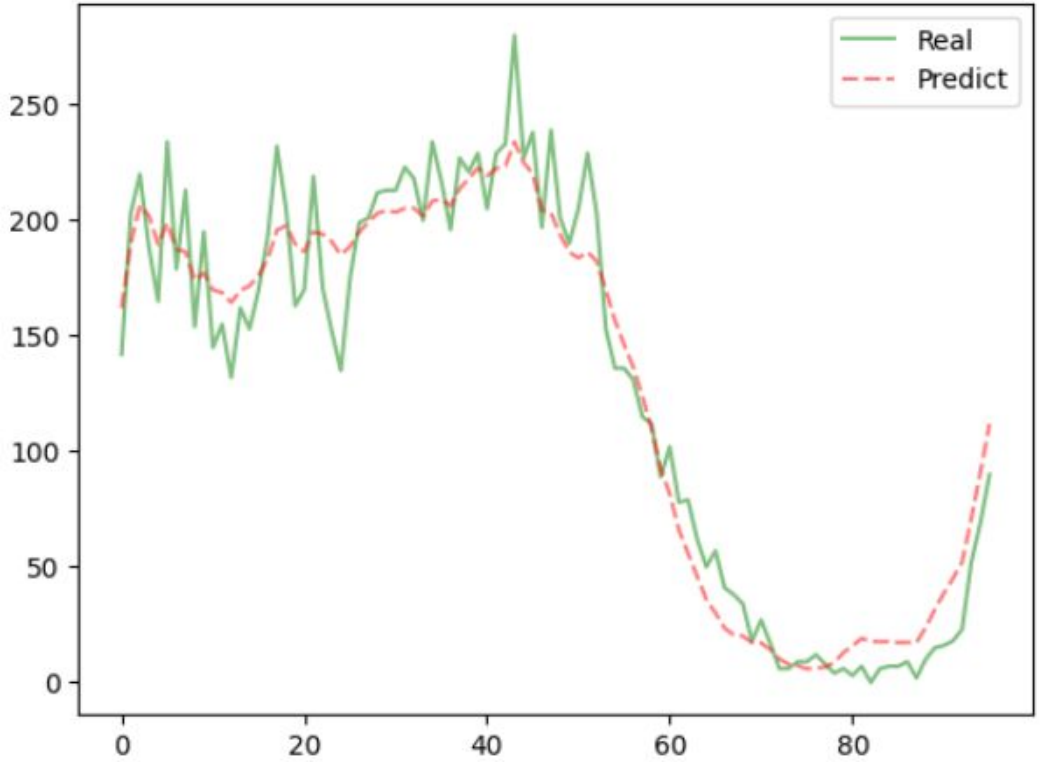
Total



Semana



VALIDAÇÃO DO MODELO



R2: 0.97%

Loss: 0.0075

RMSE: 0.08

Erro Absoluto: 11.5

Desvio Padrão: 15.4

Tempo: 20 Minutos

Dessa maneira, é possível perceber que o resultado é satisfatório com um tempo razoável de processamento, algo importante uma vez que serão realizados 263 modelos, um para cada zona.

VALIDAÇÃO COMPLETA

Após usar a mesma arquitetura para todas as zonas, obtem-se os resultados a serem validados:

PROPORTIONAL MAE MEAN: 0.2839
BEST TO WORST PROPORTIONAL MAE

Zone 107	0.091991
Zone 68	0.099148
Zone 90	0.099447
Zone 186	0.100049
Zone 164	0.101097
...	...
Zone 93	0.685492
Zone 61	0.727791
Zone 82	0.747463

PROPORTIONAL STD MEAN: 0.2364
BEST TO WORST PROPORTIONAL STD

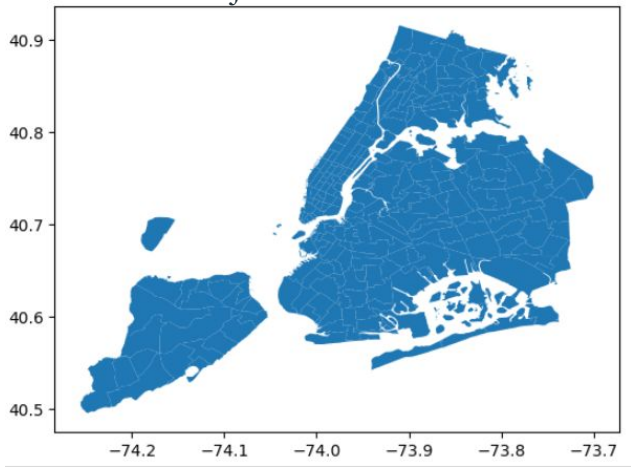
Zone 107	0.066722
Zone 186	0.077964
Zone 231	0.078831
Zone 263	0.084811
Zone 87	0.088428
...	...
Zone 49	0.522715
Zone 66	0.552117
Zone 61	0.562510

PROPORTIONAL R^2 MEAN: 0.7209
BEST TO WORST R^2

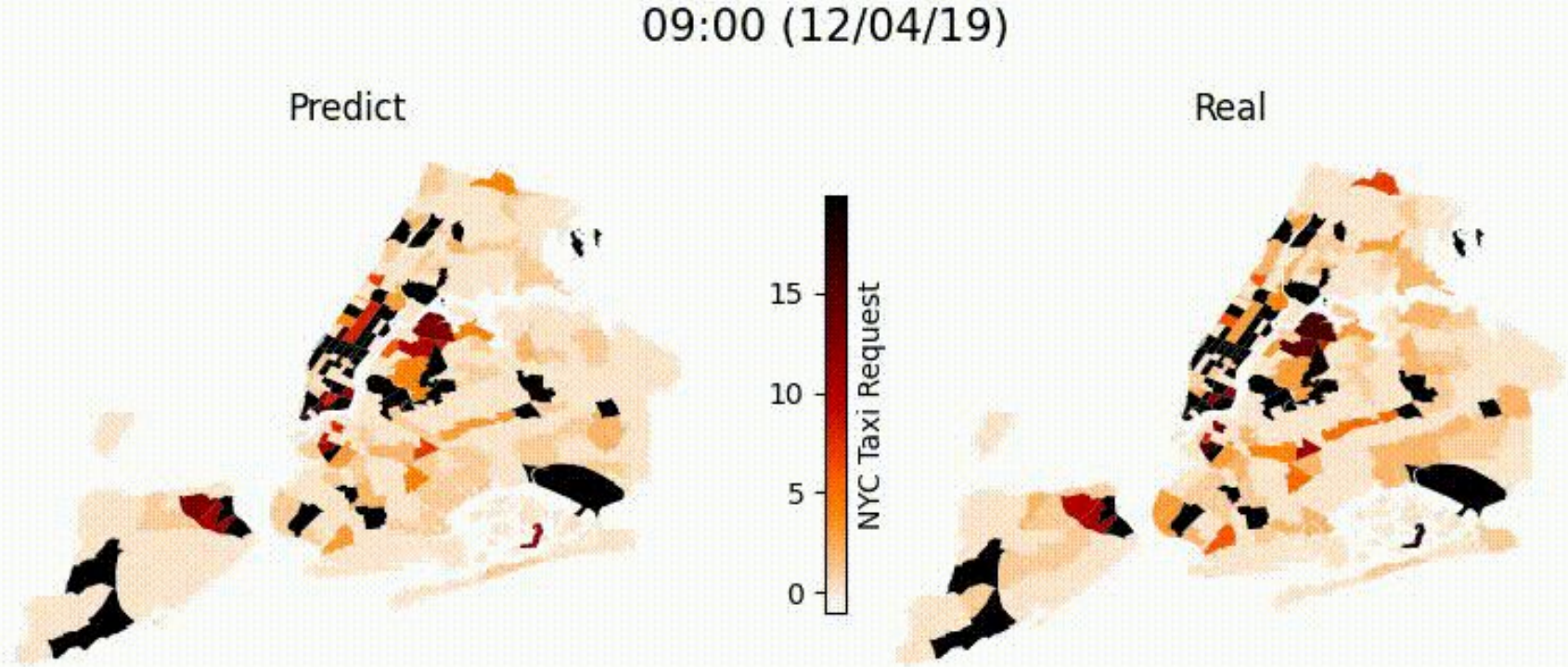
Zone 107	0.947459
Zone 236	0.940946
Zone 143	0.937237
Zone 231	0.936335
Zone 113	0.935573
...	...
Zone 93	0.220656
Zone 49	0.198079
Zone 129	0.149180

A plotagem do mapa de calor pode ser feita com os resultados utilizando o geopandas e obtendo o GeoJSON do NYC Open Data.

GeoJSON de Nova York



RESULTADO



CONCLUSÃO

Após a validação dos resultados é possível concluir que o LSTM consegue ser uma grande ferramenta de aplicações para previsibilidade e compreensão das demandas de transportes privados em específicas localizações. Isso **abre margem para abordagens similares com outros tipos de dados semelhantes**, como demanda de transporte público, policiamento ou outros serviços de assistência momentânea.

Entretanto dois grandes desafios descobertos desse tipo de aplicação são:

- **Desafio da Coleta de Dados**: No caso deste projeto a coleta de dados é feita de forma eficiente pela prefeitura de Nova York, garantindo uma abertura para as metodologias, porém em outros casos nem sempre isso acontece, como por exemplo transportes públicos em cidades brasileiras.

- **Desafio de Processamento**: A quantidade de dados aumenta exponencialmente a necessidade computacional, para melhores resultados seria necessário uma entrada com mais períodos e uma arquitetura da rede neural mais robusta, além, claro, do volume total dos dados. Idealmente seria mais dinâmico um processamento distribuído em nuvem.

REFERÊNCIAS

- Engineering Extreme Event Forecasting at Uber with Recurrent Neural Networks
  uber.com/en-BR/blog/neural-networks

- Time-series Extreme Event Forecasting with Neural Networks at Uber
  cs.columbia.edu/~lierranli/publications/TSW2017_paper.pdf

- Open Data for All New Yorkers
  opendata.cityofnewyork.us