Desafio GAVB

Vaga - Cientista de Dados

Candidato - Raphael Crespo Pereira
rcp@cin.ufpe.br

Descrição

Esse desafio utiliza como base de dados as ocorrências de acidentes de trânsito na cidade de Recife.

Objetivo

O objetivo deste desafio é prever a quantidade de ocorrências de acidentes de trânsito do próximo dia.

Base de Dados

Ocorrências de acidentes de trânsito na cidade de Recife, contendo ocorrências entre os anos de 2015 e 2019;

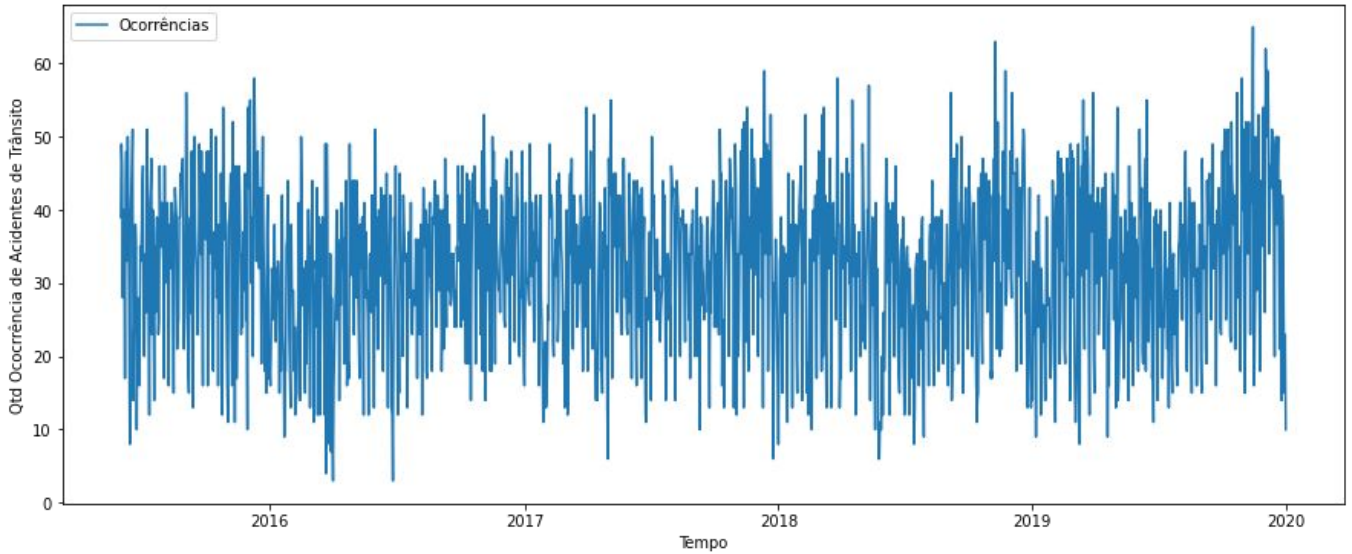
Os dados vieram em **5 arquivos separados** sendo 1 referente a cada ano de registros.

Onde os dados referentes ao ano de 2015 começam em 1 de Junho e os demais possuem registros durante todo o ano.

Os dados em cada um das bases são registros de acidentes de trânsito ou seja **cada linha da base representa 1 ocorrência**. Sendo assim, para transformar em uma base de séries temporais foi **agrupado e somado a quantidade de registros** para cada um dos dias do ano.

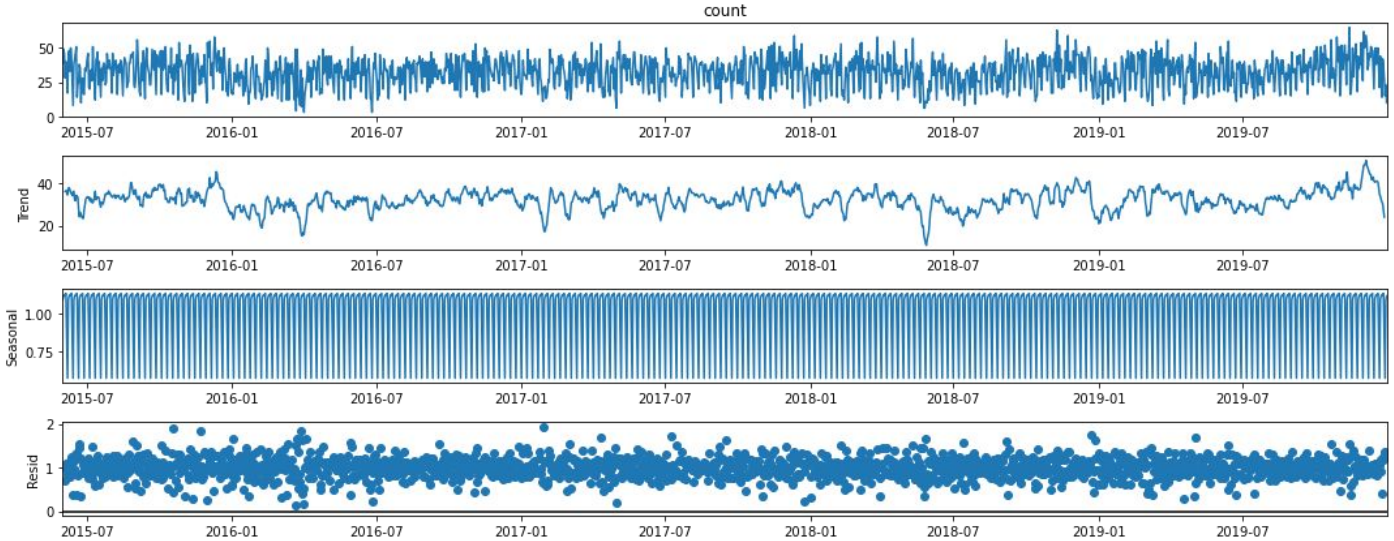
Base de Dados

Após o agrupamento dos dados, pode-se observar a série temporal a ser analisada.



Análise Descritiva

Como primeira etapa foi realizada uma análise exploratória dos dados e a decomposição da série para identificar os padrões de tendência e sazonalidade. Onde se observou que a série tem uma sazonalidade bem definida

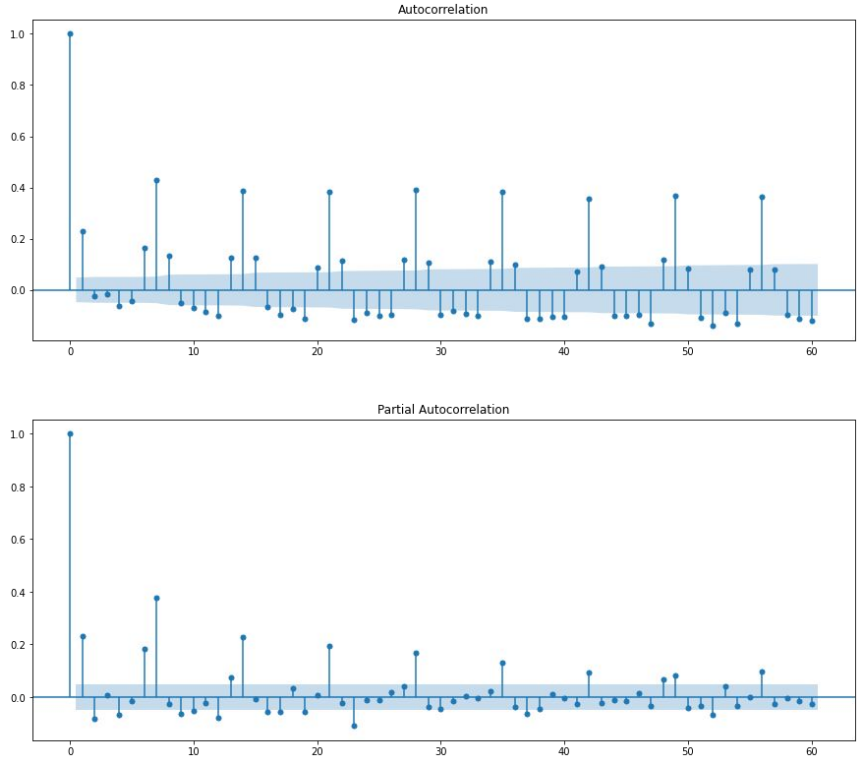


Análise Descritiva

1.1 Existem lags relevantes ?

Para responder esta pergunta foi utilizado os gráficos de autocorrelação e de autocorrelação parcial. E se observou que a série possui uma sazonalidade de 7 dias e possui. O gráfico de autocorrelação parcial indica a importância dos lags.

Pelo gráfico PACF podemos observar que os lags: **1, 6, 7, 13, 14 e os demais múltiplos de 7 até o lag 56** são significativos por estarem acima do intervalo de confiança.



Análise Descritiva

1.2 A série é estacionária ? baseado em qual análise ?

Para verificar a estacionariedade da série é recomendado utilizar o teste Kwiatkowski-Phillips-Schmidt-Shin de estacionariedade, e se o p-valor encontrado for menor do que o nível de significância de 5% a série é considerada estacionária.

```
kpss_test = kpss(serie,regression='ct', nlags='auto')
```

```
print('kpss = %f.' % kpss_test[0] ,'P-value = %f.' % kpss_test[1],'\n')
```

```
kpss = 0.169701. P-value = 0.030249.
```

Como pode-se observar o p-valor encontrado foi de aproximadamente 0.03 e é menor do que os 5% sendo assim, podemos considerar a série estacionária.

Análise Descritiva

1.3 Quais outras estatísticas podem ajudar no entendimento da distribuição?

Para confirmar o resultado é recomendado realizar o teste de Dickey-Fuller Aumentado em combinação com o teste realizado anteriormente KPSS.

A hipótese nula do Dickey-Fuller Aumentado é que existe uma raiz unitária, com hipótese alternativa de que não existe uma raiz unitária. Se o valor de p estiver acima de um tamanho crítico, não podemos rejeitar a existência de uma raiz unitária.A hipótese nula do Dickey-Fuller Aumentado é que existe uma raiz unitária, com hipótese alternativa de que não existe uma raiz unitária. Se o valor de p estiver acima de um tamanho crítico, não podemos rejeitar a existência de uma raiz unitária.

```
adf = adfuller(serie, regression='ct', autolag="t-stat")
print('ADF = %f.' % adf[0] ,'P-value = %f.' % adf[1],'LAGS = %.i' % adf[2],'\n')
```

```
ADF = -6.663148. P-value = 0.000000. LAGS = 22
```

Como pode-se observar o teste tambem apresentou o p-valor menor do que 5% corroborando pela estacionariedade.

Outras análises possiveis são verificar a normalidade da série, e normalidade do resíduo para identificação se o mesmo é um ruído branco ou se possui padrões possíveis de serem extraídos.

Modelagem

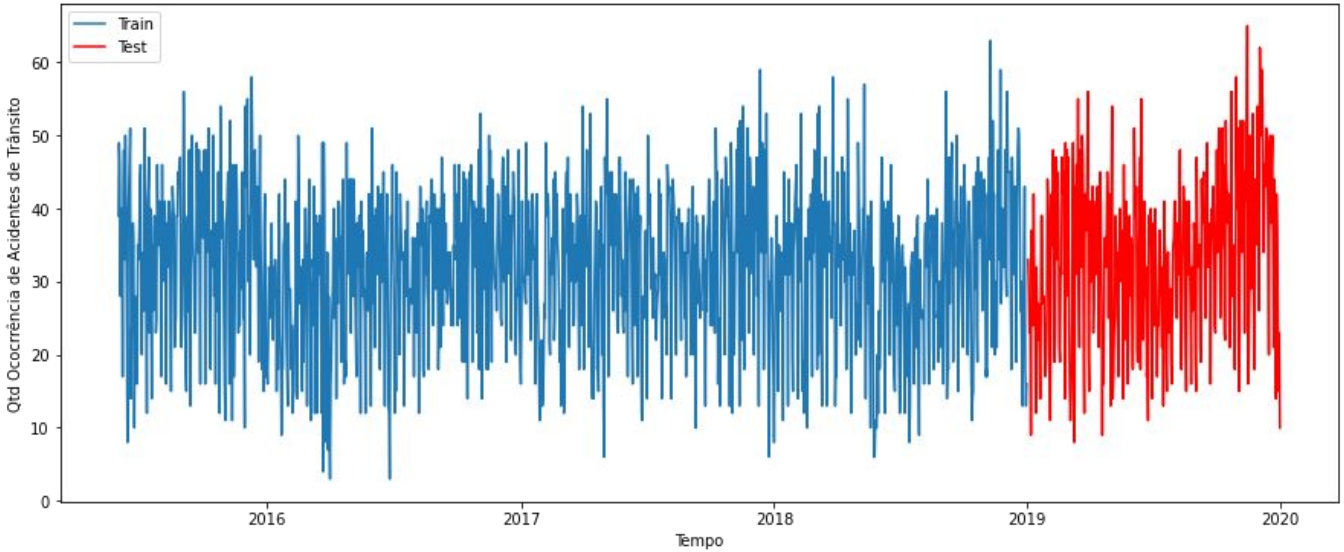
2.1. Avaliar diferentes modelos

Durante a análise foram testados 2 modelos, o primeiro foi o modelo clássico e um dos mais utilizados para previsão de séries temporais ARIMA e o outro foi um modelo de aprendizagem de máquina utilizando uma etapa de otimização para definir os hiperparâmetros e a quantidade de lags a serem utilizados pelo modelo.

Modelagem

MODELO ARIMA

O Conjunto de teste definido pelo desafio foi o ano de 2019. Sendo assim para esta modelagem foi utilizada todos os dados de 2015 a 2018 para treinamento como pode ser observado na imagem abaixo.



Modelagem

MODELO ARIMA

Para realizar a modelagem ARIMA foi utilizado a biblioteca python pmdarima com o método auto_arima que utiliza o metodo de box jenkins para estimar os parâmetros minimizando o Critério de Informação Akaike.

O melhor modelo encontrado foi um modelo SARIMA(3,0,0)(1,0,1)

Onde o algoritmo corroborou com os resultados encontrados na analise descritiva de que a série temporal é estacionária. Já que o modelo possui parametro de diferencião I = 0.

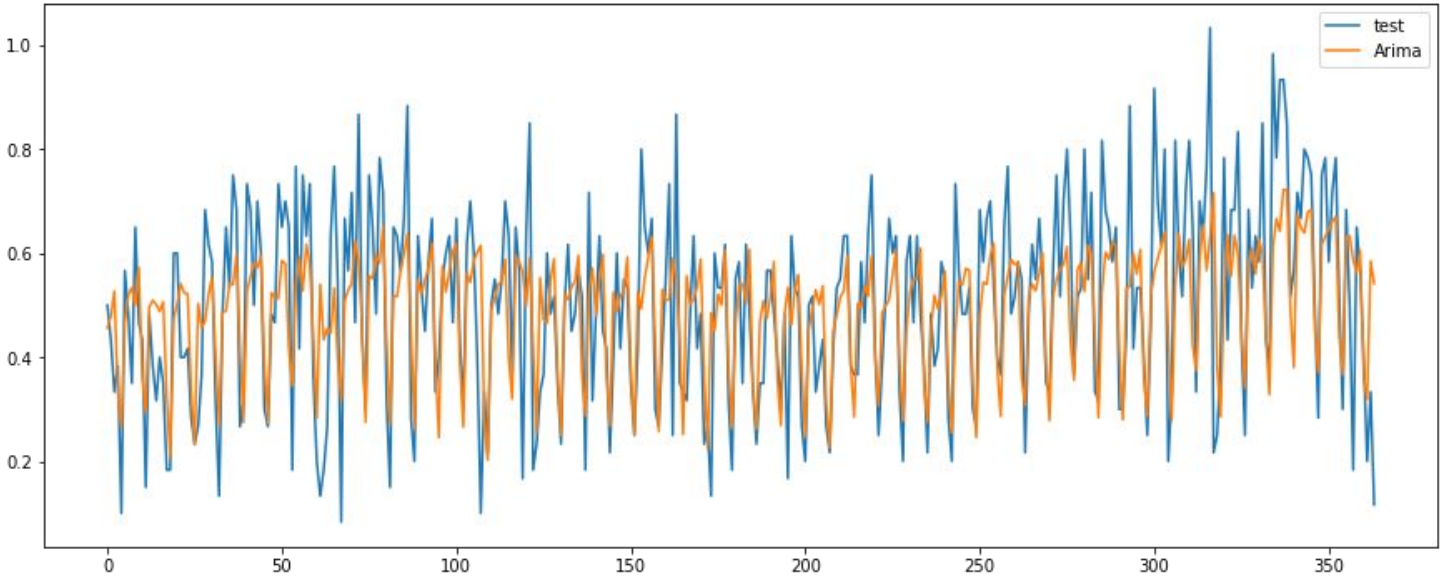
```
Performing stepwise search to minimize aic
 ARIMA(2,0,2)(1,0,1)[7] intercept   : AIC=inf, Time=6.60 sec
 ARIMA(0,0,0)(0,0,0)[7] intercept   : AIC=-817.133, Time=0.18 sec
 ARIMA(1,0,0)(1,0,0)[7] intercept   : AIC=-1086.948, Time=2.07 sec
 ARIMA(0,0,1)(0,0,1)[7] intercept   : AIC=-1022.163, Time=0.94 sec
 ARIMA(0,0,0)(0,0,0)[7]             : AIC=1966.759, Time=0.11 sec
 ARIMA(1,0,0)(0,0,0)[7] intercept   : AIC=-882.743, Time=0.24 sec
 ARIMA(1,0,0)(2,0,0)[7] intercept   : AIC=-1166.606, Time=6.05 sec
 ARIMA(1,0,0)(2,0,1)[7] intercept   : AIC=-1377.483, Time=10.48 sec
 ARIMA(1,0,0)(1,0,1)[7] intercept   : AIC=inf, Time=5.15 sec
 ARIMA(1,0,0)(2,0,2)[7] intercept   : AIC=inf, Time=13.60 sec
 ARIMA(1,0,0)(1,0,2)[7] intercept   : AIC=inf, Time=13.57 sec
 ARIMA(0,0,0)(2,0,1)[7] intercept   : AIC=-1320.956, Time=7.49 sec
 ARIMA(2,0,0)(2,0,1)[7] intercept   : AIC=-1384.137, Time=10.92 sec
 ARIMA(2,0,0)(1,0,1)[7] intercept   : AIC=-1387.835, Time=9.41 sec
 ARIMA(2,0,0)(0,0,1)[7] intercept   : AIC=-1019.450, Time=2.22 sec
 ARIMA(2,0,0)(1,0,0)[7] intercept   : AIC=-1085.684, Time=6.57 sec
 ARIMA(2,0,0)(1,0,2)[7] intercept   : AIC=inf, Time=15.41 sec
 ARIMA(2,0,0)(0,0,0)[7] intercept   : AIC=-888.352, Time=0.32 sec
 ARIMA(2,0,0)(0,0,2)[7] intercept   : AIC=-1074.673, Time=5.21 sec
 ARIMA(2,0,0)(2,0,0)[7] intercept   : AIC=-1164.669, Time=10.33 sec
 ARIMA(2,0,0)(2,0,2)[7] intercept   : AIC=inf, Time=10.54 sec
 ARIMA(3,0,0)(1,0,1)[7] intercept   : AIC=-1397.300, Time=8.69 sec
 ARIMA(3,0,0)(0,0,1)[7] intercept   : AIC=-1017.871, Time=2.90 sec
 ARIMA(3,0,0)(1,0,0)[7] intercept   : AIC=-1086.133, Time=7.88 sec
 ARIMA(3,0,0)(2,0,1)[7] intercept   : AIC=-1382.594, Time=20.42 sec
 ARIMA(3,0,0)(1,0,2)[7] intercept   : AIC=inf, Time=18.57 sec
 ARIMA(3,0,0)(0,0,0)[7] intercept   : AIC=-886.503, Time=0.33 sec
 ARIMA(3,0,0)(0,0,2)[7] intercept   : AIC=-1074.428, Time=5.32 sec
 ARIMA(3,0,0)(2,0,0)[7] intercept   : AIC=-1167.809, Time=16.45 sec
 ARIMA(3,0,0)(2,0,2)[7] intercept   : AIC=inf, Time=21.33 sec
 ARIMA(4,0,0)(1,0,1)[7] intercept   : AIC=-1394.425, Time=10.79 sec
 ARIMA(3,0,1)(1,0,1)[7] intercept   : AIC=-1395.469, Time=9.54 sec
 ARIMA(2,0,1)(1,0,1)[7] intercept   : AIC=-1385.778, Time=8.87 sec
 ARIMA(4,0,1)(1,0,1)[7] intercept   : AIC=-1393.214, Time=10.27 sec
 ARIMA(3,0,0)(1,0,1)[7]             : AIC=inf, Time=5.12 sec

Best model:  ARIMA(3,0,0)(1,0,1)[7] intercept
Total fit time: 283.972 seconds
```

Modelagem

MODELO ARIMA - Previsão conjunto de Teste

mse = 0.031

mae= 0.142



Modelagem

MODELO SVR

O modelo SVR foi implementado utilizando a biblioteca sklearn. Para melhorar a performance foi realizado uma validação com grid-search para encontrar os melhores hiperparâmetros do modelo e a melhor quantidade de lags a ser utilizado.

Os hiper parametros selecionados foram:

{'C': 0.1, 'epsilon': 0.01, 'gamma': 1, 'kernel': 'rbf'}

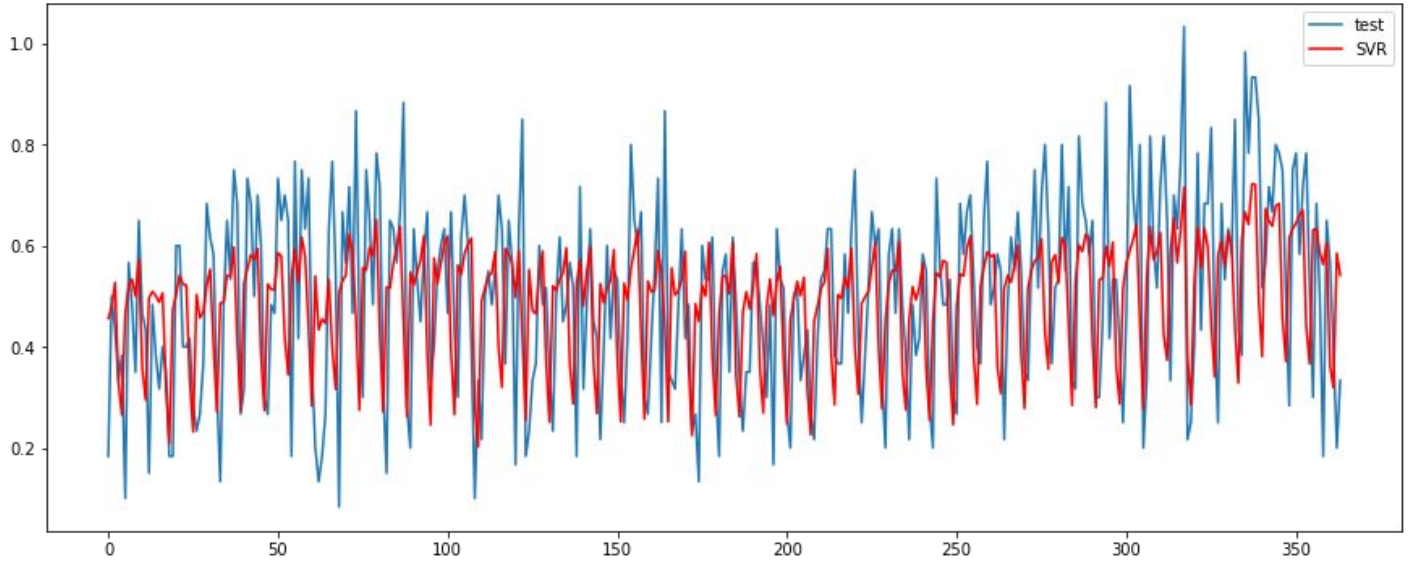
qtd_lags = 26

Modelagem

MODELO SVR - Previsão conjunto de Teste

mse = 0.023

mae= 0.117



Modelagem

2.5 É possível adicionar outras variáveis para ajudar no processo de previsão? Quais? melhoraram os resultados?

É possível avaliar as demais variáveis presentes no dataset e incluí-las como variáveis exógenas ou realizar um procedimento de feature engineering para encontrar variáveis que possam aumentar a assertividade do modelo.

2.6 Qual abordagem você usaria nesse contexto?

Realizar uma avaliação do impacto das variáveis a serem utilizadas no desempenho do modelo.

Conclusão

A previsão de séries temporais é uma atividade extremamente complexa e difícil de se modelar devido a natureza do problema de modelos de séries temporais univariadas. O dataset proposto se mostrou um desafio devido à alta variação dos dados, o que prejudica o ajuste de modelos lineares como ARIMA.

O modelo SVR com a utilização de uma etapa de otimização se mostrou com um desempenho superior quando comparado com as métricas encontradas no modelo ARIMA.