

Nostradamus: plataforma de aprendizado de máquina como um serviço para tratamento, análise, visualização e previsão de séries temporais providas pelo usuário

Jayme T. Anchante¹, André R. A. Grégio²,

¹Aluno do programa de Especialização em Data Science & Big Data, jayme.anchante@disroot.org;

²Professor do Departamento de Informática - DINF/UFPR, gregio@ufpr.br;

Resumo

O presente trabalho apresenta uma proposta de sistema automatizado de previsão de séries temporais chamado Nostradamus. Inicia-se o trabalho apresentando os principais conceitos de séries temporais, suas principais características e formas de tratamento. Em seguida, expõe-se o que se chamou de abordagem da estatística inferencial e de abordagem da estatística preditiva e as formas de tratamento e previsão de séries temporais de cada uma. Posteriormente, propõe-se o sistema automatizado Nostradamus, discutindo a forma de otimização e as etapas que o sistema percorre para alcançar a previsão final. No benchmark em sete bases de dados de séries temporais, o sistema Nostradamus alcançou o melhor resultado em cinco delas.

Palavras chaves: aprendizado de máquina, automl, séries temporais.

Introdução

- Série temporal são dados pontuais ordenados temporalmente. Apesar do tempo ser contínuo, normalmente uma série temporal envolve dados discretos, tomados em sequência de períodos igualmente espaçados e sucessivos de tempo. Além da análise de séries temporais, ou seja, métodos para extrair estatísticas úteis e outras características dos dados, outro tema bastante relevante e objeto de estudos do presente trabalho é a predição de séries temporais, processo que envolve o uso de modelos que predigam valores futuros com base em valores passados observados.
- O objetivo do trabalho é a criação de um sistema automatizado de previsão de séries temporais que abstraia completamente questões técnicas como processamento dos dados, engenharia de características, modelagem, validação do resultado e realizar uma previsão que extrapole o período de amostra inicial.

Material e Métodos

- O sistema Nostradamus pode ser definido numa função objetivo tal que

$$\begin{aligned} \min \quad & MAE = f(FE(p), Algo(HP)) \\ \text{tal que} \quad & p \leq n/20 \\ & Algo(HP) \rightarrow Data \end{aligned} \quad (1)$$

o objetivo é minimizar o erro absoluto médio (*MAE* em inglês) das previsões de acordo com uma função de *FE* que é a engenharia de características e de um *Algo* ou algoritmo que tem *HP* ou hiperparâmetros. A primeira restrição significa que a ordem *p* da engenharia de características não deve exceder a vigésima parte do número de dados da série temporal. A segunda restrição afirma que os hiperparâmetros deve estar de acordo com a distribuição dos dados.

- Como engenharia de características, é feita uma transformação da série temporal em um processo autorregressivo $AR(p)$, em que *p* ou ordem máxima de defasagens da variável alvo utilizada
- Os algoritmos preditivos implementados pela plataforma são: regressão linear, *elasticnet* (uma regressão linear com regularização L1 e L2), florestas aleatórias, *k* vizinhos próximos, rede neural profunda, implementados pela biblioteca *scikit-learn*; um algoritmo de impulso de gradiente extremo baseado em árvore de decisão com *boosting* chamado de XGBoost.
- A forma de minimização é feita por meio de um processo iterativo em que a cada etapa um valor de *p* é escolhido para a engenharia de características, assim como um dos algoritmos mencionados anteriormente com uma configuração de hiperparâmetros amostrados de forma que faça sentido com os dados obedecendo a segunda restrição.
- Um algoritmo de busca Bayesiana é responsável pela otimização da função objetivo. Ele funciona da seguinte maneira: inicialmente são feita *x* iterações amostradas de forma totalmente aleatória para que o sistema gere dados, as amostragens de parâmetros são as características que serão inputadas e o erro absoluto médio é o alvo do algoritmo Bayesiano, assim após *x* iterações, o algoritmo começará a testar espaços amostrais mais promissores de forma mais consistente, mas mantendo a busca aleatória a cada *y* iterações para que o sistema não caia em um mínimo local.

Resultados e discussões

- As séries temporais utilizadas como *benchmark* são: *sunspots*, número de manchas solares por ano entre 1700 e 2008; *airpassengers*, número mensal de passageiros de vôos de avião; *austres*, dados residenciais trimestrais; *heartrate*, frequência cardíaca; *lynx*, número de linces capturados por ano no Canadá entre 1821 e 1934; *wineind*, vendas de vinho da indústria australiana; e *woolynrnq*, produção trimestral de lã na Austrália.
- Foram separados os dez por cento dos pontos mais recentes de cada conjunto de dados como base de teste e cada algoritmo recebeu os noventa por cento de dados restantes para treinamento.

Tabela 1: Erro absoluto médio nos 10% de cada série temporal utilizados para validação para a previsão de cada um dos três sistemas, Nostradamus, ARIMA e Prophet

	Base	Nostradamus	ARIMA	Prophet
<i>sunspots</i>	18.93	44.41	45.50	
<i>airpassengers</i>	88.74	39.35	57.93	
<i>austres</i>	41.74	56.43	115.42	
<i>heartrate</i>	1.64	2.01	2.16	
<i>lynx</i>	198.49	652.95	967.96	
<i>wineind</i>	5939.42	3817.89	3762.36	
<i>woolnrq</i>	426.8	427.38	592.50	

- No gráfico abaixo podemos ver a forma como cada sistema fez as previsões. A série temporal analisada é *sunspots*. Enquanto o sistema Nostradamus tenta verdadeiramente ajustar o padrão das séries, os modelos ARIMA e Prophet optam por previsões mais "conservadoras" em torno da média dos dados. O sistema Nostradamus obtém uma performance melhor que os demais sistemas, pois possui um erro absoluto médio menor que os modelos ARIMA e Prophet.

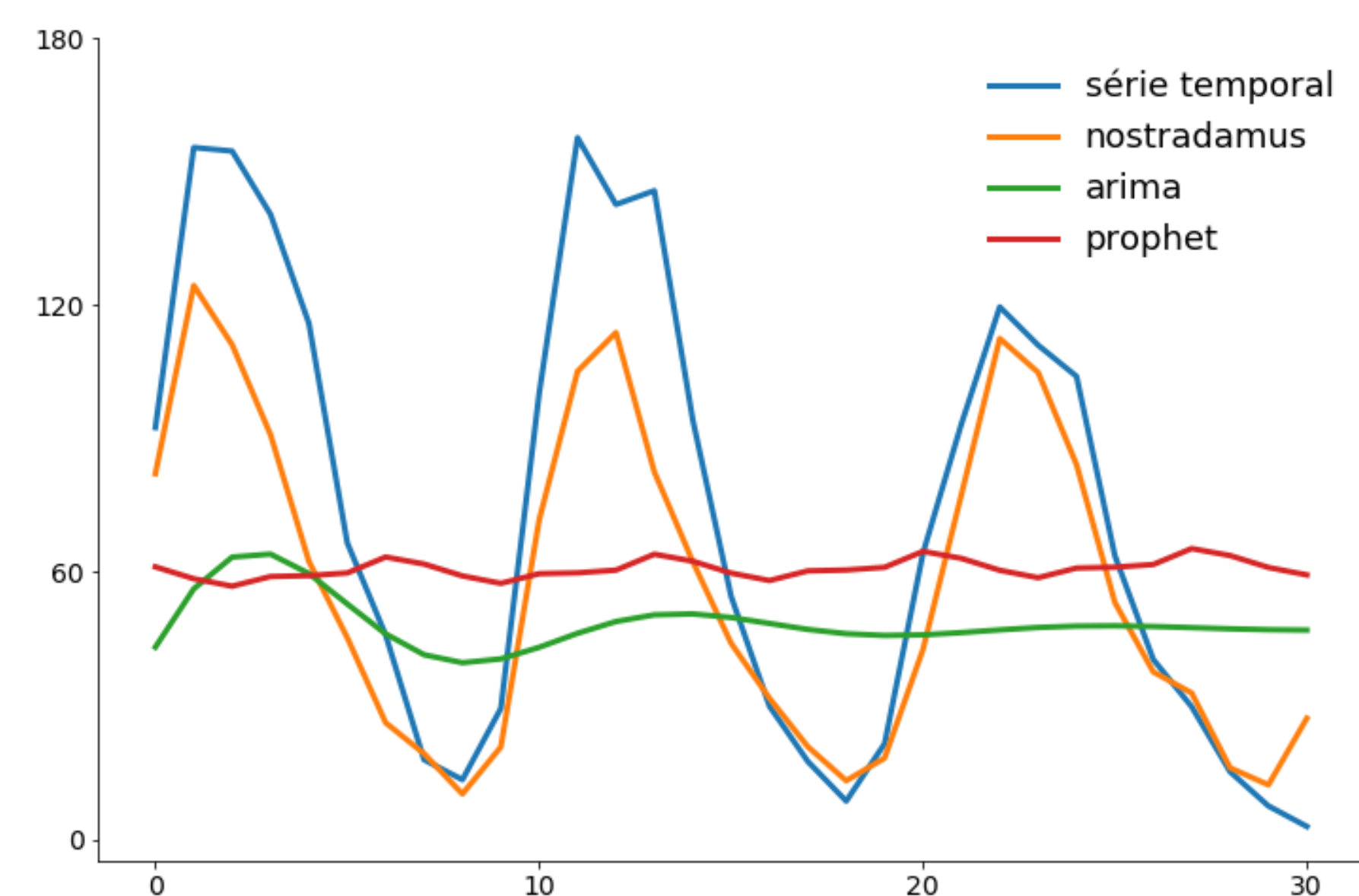


Figura 1: Previsão de cada sistema na base sunspots nos 10% da base reservados para teste

Conclusões

Algumas conclusões preliminares alcançadas pelo trabalho são:

- É possível criar um sistema automatizado de previsão de séries temporais
- O sistema Nostradamus demonstrou razoável poder preditivo frente aos demais algoritmos, alcançando a melhor performance em 5 das 7 séries temporais diversas utilizadas para *benchmark*

Principais Referências

- GUJARATI, D. N.; PORTER, D. *Basic Econometrics*. 5ª ed. Nova Iorque: Mc Graw-Hill/Irwin, 2009.
- PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825-2830, 2011.
- CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. São Francisco, CA, USA: ACM, 13-17 Agosto de 2016. p. 785-794. arXiv:1603.02754.