



CASE – Bolsista IA Londrina/PR - Remoto

Requerente

Fábrica de Software do SENAI

Candidato

Tiago Dutra Galvão

Londrina, Janeiro de 2024

1. Introdução:

A predição de vendas é um processo crucial para as empresas, permitindo antecipar a demanda futura com base em dados históricos e variáveis relevantes. Ao empregar modelos preditivos, como ARIMA, SARIMAX, LSTM e Neural Prophet, as organizações podem tomar decisões informadas sobre estoques, produção e estratégias de marketing. Essa abordagem visa otimizar o desempenho, minimizar custos e melhorar a eficiência operacional, contribuindo para o sucesso a longo prazo no mercado.

A presente análise surge no contexto de uma parceria com uma empresa especializada em gestão de vendas, que busca a implementação de um sistema de previsão de vendas. O desafio em questão envolve o desenvolvimento de um MVP (Minimum Viable Product), abrangendo não apenas o controle de acesso ao sistema, mas também um módulo crucial de previsão de vendas, fundamentado em dados históricos fornecidos pela empresa.

O arquivo fornecido para experimentação e desenvolvimento (Sample - Superstore.xls) serve como ponto de partida para a execução de experimentos e validação do MVP. A visão geral do projeto contempla a criação de um sistema web, utilizando o robusto framework Python Django, que incorpora não apenas a autenticação de usuários e gerenciamento destes, mas também um módulo de previsão capaz de antecipar as vendas para os próximos 12 meses, após a inserção de dados no formato especificado.

É essencial destacar que a abordagem adotada visa garantir a eficiência do sistema. O processamento dos dados ocorre de maneira independente, separado do sistema principal, assegurando que a solicitação de previsão não comprometa o tempo de resposta ao usuário. Este relatório busca avaliar e comparar quatro modelos distintos - ARIMA, SARIMAX, LSTM e Neural Prophet - na busca pela otimização das previsões de vendas para atender às expectativas da empresa em questão.

Ao longo deste documento, detalharemos a implementação, resultados e considerações de cada modelo, com o objetivo final de identificar aquele que melhor atende às necessidades específicas do projeto, contribuindo assim para a eficácia do sistema de previsão de vendas.

2. Descrição dos Modelos:

ARIMA (AutoRegressive Integrated Moving Average):

O modelo ARIMA (AutoRegressive Integrated Moving Average) é uma técnica estatística que combina componentes autoregressivos, integrativos e de

médias móveis. Sua aplicação implica a análise de padrões temporais nos dados de vendas, sendo configurável para lidar com tendências e sazonalidades. A escolha das ordens (p , d , q) é essencial para adequar o modelo às características específicas do conjunto de dados, permitindo previsões precisas.

O ARIMA (AutoRegressive Integrated Moving Average) é um modelo estatístico amplamente utilizado para a previsão de séries temporais univariadas. Ele incorpora três componentes principais: AutoRegressivo (AR), Integrado (I) e de Médias Móveis (MA). Vamos explicar cada um desses componentes:

AutoRegressivo (AR):

O componente AR refere-se à parte autoregressiva do modelo. Ele modela a relação entre uma observação atual e suas observações anteriores em termos de uma equação de regressão. O termo "AutoRegressivo" indica que a previsão é uma combinação linear das observações passadas.

Integrado (I):

O componente I representa a ordem de diferenciação necessária para tornar a série temporal estacionária, ou seja, remover tendências e padrões que podem tornar a análise mais difícil. A diferenciação envolve subtrair observações consecutivas para tornar a série estacionária.

Médias Móveis (MA):

O componente MA refere-se à parte de médias móveis do modelo. Ele modela a relação entre uma observação atual e um erro residual da média móvel de observações anteriores. Isso ajuda a capturar padrões não explicados pelo componente AR.

A notação padrão para um modelo ARIMA é $ARIMA(p, d, q)$, onde:

p : Ordem do componente AR (ordem dos termos autoregressivos).

d : Ordem do componente I (número de diferenciações necessárias para tornar a série estacionária).

q : Ordem do componente MA (ordem dos termos de médias móveis).

O processo de ajuste de um modelo ARIMA geralmente envolve a identificação adequada de p , d e q com base na análise gráfica da série temporal, como funções de autocorrelação e autocorrelação parcial. Uma vez ajustado, o modelo é capaz de fazer previsões para valores futuros com base nas relações identificadas nos dados históricos.

Em resumo, o ARIMA é uma ferramenta poderosa para modelar e prever séries temporais, permitindo a consideração de padrões autoregressivos, efeitos

sazonais e tendências temporais, o que o torna adequado para o contexto de previsão de vendas.

SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors):

SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors) é uma extensão do modelo ARIMA, projetado para lidar com séries temporais que exibem padrões sazonais e podem ser influenciadas por variáveis externas, chamadas de regressores exógenos. Este modelo é capaz de modelar padrões sazonais nas séries temporais. Ele inclui termos adicionais para capturar variações recorrentes ao longo do tempo, proporcionando uma melhor adaptação a dados com influências sazonais por meio de ordens sazonais (P, D, Q, s) para lidar com esses tipos de variações. Essas ordens são análogas às ordens não sazonais em um modelo ARIMA, representando os termos autoregressivos, de diferenciação e de médias móveis sazonais, respectivamente. O parâmetro "s" indica a periodicidade sazonal, ou seja, o número de observações em uma temporada.

Uma das características distintivas do SARIMAX é a capacidade de incorporar variáveis exógenas no modelo. Essas variáveis são fatores externos que podem influenciar a série temporal, mas não são previstas pelo modelo. A inclusão de regressores exógenos permite capturar melhor a complexidade dos dados e melhorar a precisão das previsões. Ao integrar sazonalidade e variáveis exógenas, o SARIMAX oferece uma abordagem mais flexível e abrangente para a modelagem de séries temporais em comparação com o ARIMA padrão. Isso o torna uma escolha valiosa em situações em que a sazonalidade e influências externas desempenham um papel significativo na dinâmica das vendas ou de outros fenômenos temporais.

LSTM (Long Short-Term Memory):

A Long Short-Term Memory (LSTM) é uma arquitetura de rede neural recorrente (RNN) projetada para superar as limitações das RNNs tradicionais ao lidar com dependências temporais de longo prazo. As LSTMs foram introduzidas para resolver o problema de desvanecimento do gradiente, que ocorre quando as RNNs têm dificuldade em capturar relações de dependência a longo prazo devido à rápida diminuição dos gradientes durante o treinamento.

As LSTMs possuem unidades de memória chamadas células de memória, que são capazes de armazenar informações por longos períodos. Essas células são protegidas de atualizações desnecessárias por meio de portões. Para

controlar o fluxo de informações na célula de memória, as LSTMs utilizam portões sigmoid.

Esses portões decidem quais informações entrarão, sairão ou serão mantidas na célula de memória. Existem três tipos principais de portões em uma unidade LSTM: Portão de Esquecimento (Forget Gate): Decide quais informações antigas na célula de memória devem ser descartadas. Portão de Entrada (Input Gate): Decide quais novas informações devem ser adicionadas à célula de memória. Portão de Saída (Output Gate): Decide qual parte da célula de memória deve ser exposta como a saída do bloco LSTM.

A função de ativação tanh é comumente usada para modular os valores que fluem através da unidade LSTM. Isso ajuda a normalizar os valores entre -1 e 1, facilitando o controle do fluxo de informações na célula de memória.

A característica principal das LSTMs é a presença de conexões recorrentes que permitem que as informações fluam por toda a sequência de entrada e saída. Essa capacidade de reter informações de entradas anteriores é crucial para capturar dependências temporais de longo prazo.

Durante cada passo de tempo, a LSTM toma como entrada a observação atual e o estado oculto anterior. O estado oculto e a célula de memória são atualizados usando os portões sigmoid e a função de ativação tanh. O novo estado oculto é então usado na próxima iteração do modelo.

A arquitetura LSTM é especialmente eficaz em situações em que as dependências temporais são complexas e de longo prazo, como em séries temporais de vendas, linguagem natural ou dados sequenciais em geral. Sua capacidade de capturar e lembrar padrões temporais faz dela uma escolha valiosa para problemas de previsão e modelagem de séries temporais.

Neural Prophet:

O Neural Prophet é um modelo de previsão de séries temporais baseado em redes neurais projetado para simplificar o processo de modelagem e previsão. Ele combina a flexibilidade das redes neurais com uma implementação intuitiva, permitindo que os usuários obtenham previsões precisas sem a necessidade de ajustes complexos de hiperparâmetros. Breve descrição do modelo Neural Prophet.

Facilidade de uso é uma de suas características distintivas do Neural Prophet é a sua interface amigável, que simplifica o processo de previsão de séries temporais. Isso é particularmente benéfico para usuários que não têm experiência avançada em redes neurais, permitindo uma implementação eficiente.

Outra característica é o manuseio automático de tendências e sazonalidades, neste caso, o modelo Neural Prophet é capaz de identificar automaticamente e

modelar tendências e sazonalidades nos dados temporais. Ele evita a necessidade de intervenção manual para especificar componentes sazonais e tendências, tornando a modelagem mais automatizada.

Algo que se torna muito vantajoso é a não necessidade de tratamento de dados ausentes. O Neural Prophet é projetado para lidar com dados ausentes de maneira eficaz, preenchendo lacunas na série temporal de forma inteligente. Isso contribui para uma maior robustez na modelagem, especialmente quando se lidam com conjuntos de dados do mundo real.

Este modelo também possui uma grande adaptabilidade a diferentes contextos de negócios. O modelo é versátil e pode ser aplicado a uma variedade de contextos de negócios, incluindo previsão de vendas, demanda de produtos ou qualquer outra aplicação que envolva séries temporais. Tem uma configuração intuitiva de hiperparâmetros, onde os usuários podem ajustar facilmente os parâmetros para otimizar o desempenho do modelo sem a necessidade de uma compreensão profunda de redes neurais. Além de tudo isso, tem um baixo custo em tempo para treinamento, o Neural Prophet é projetado para treinar rapidamente, tornando-o uma escolha eficiente para cenários onde o tempo de treinamento é uma consideração importante.

Em resumo, o Neural Prophet oferece uma abordagem acessível e eficaz para a previsão de séries temporais, destacando-se pela sua facilidade de uso, manuseio automático de padrões temporais e adaptabilidade a diferentes cenários de negócios. Isso o torna uma escolha atraente para usuários que buscam uma solução poderosa sem a complexidade associada a algumas técnicas de modelagem de séries temporais.

3. Metodologia:

Para a avaliação dos modelos ARIMA, SARIMAX, LSTM e Neural Prophet, a primeira etapa envolveu a divisão dos dados fornecidos no arquivo "Sample - Superstore.xls" em conjuntos de treinamento, validação e teste. A escolha das proporções entre esses conjuntos foi realizada considerando a quantidade de dados disponíveis e a necessidade de garantir robustez nas avaliações.

O Conjunto de Treinamento foi utilizado para treinar os modelos e aprender padrões a partir dos dados históricos, um Conjunto de Validação: empregado para ajustar hiperparâmetros e otimizar o desempenho dos modelos durante o processo de treinamento, quando necessário e, por fim, um conjunto de Teste: Reservado para avaliar o desempenho final e comparativo dos modelos, fornecendo uma avaliação objetiva de suas capacidades de previsão.

Já os critérios de avaliação para avaliação do desempenho dos modelos incluem métricas comuns em problemas de previsão de séries temporais:

Erro Médio Absoluto (MAE): Medida da média absoluta dos erros entre as previsões e os valores reais. Raiz do Erro Quadrático Médio (RMSE): Métrica que penaliza erros maiores de forma mais significativa, indicando a raiz quadrada da média dos erros quadráticos. Erro Percentual Absoluto Médio (MAPE): Calcula a média dos erros percentuais absolutos entre as previsões e os valores reais.

Cada modelo (ARIMA, SARIMAX, LSTM, Neural Prophet) foi implementado utilizando bibliotecas apropriadas, como statsmodels para ARIMA e SARIMAX, TensorFlow/Keras para LSTM, e a biblioteca NeuralProphet para o modelo Neural Prophet.

4. Resultados:

Apresentação dos resultados de desempenho para cada modelo serão mostrados nos arquivos .ipynb dos códigos de implementação. Lá é possível verificar que o modelo que possui o melhor desempenho é o modelo Neural Prophet, seguido do modelo SARIMAX. Ambos possuem uma pequena diferença. Esses modelos são capazes de levar em consideração a sazonalidade das séries temporais, no entanto, os modelos do tipo SARIMAX tem uma excelente performance para séries temporais estacionárias, o que não é o caso dos dados trabalhados. Esses resultados comparativos podem ser visualizados nos códigos desenvolvidos.

5. Conclusões:

Foi possível obter a identificação do modelo que obteve os melhores resultados, sendo escolhido para a elaboração do sistema web o modelo SARIMAX por ser um modelo em que é possível realizar melhorias ao longo do projeto pois pode-se alterar os parâmetros do modelo, uma vez que o modelo Neural Prophet tem os ajustes automáticos, o que pode dificultar tais avanços para novos dados.

6. Recomendações:

Para melhor a performance do modelo SARIMAX pode-se realizar transformações na série original deixando-a com distribuição normal e estacionária, contribuindo assim para um modelo preditivo satisfatório. Isso foi realizado.

7. Observações: Sistema Web

O sistema Web foi elaborado com o framework Django que utiliza um padrão de arquitetura de software semelhante ao conhecido como Model-View-Controller (MVC). No caso do Django, esse padrão é implementado como Model-View-Template (MVT). O "Model" lida com a manipulação de dados e regras de negócios, a0 "View" lida com a apresentação e interação com o usuário e o "Template" lida com a representação visual dos dados. O Django possui um ORM (Object-Relational Mapping) embutido, permitindo que os desenvolvedores interajam com o banco de dados usando objetos Python em vez de SQL direto. Isso facilita a manipulação de dados e a migração entre diferentes sistemas de banco de dados.

Uma praticidade para aplicações é que o Django fornece uma interface de administração automática gerada a partir dos modelos definidos. Isso facilita a administração do sistema, especialmente durante o desenvolvimento e a fase inicial de implementação.

Django é amplamente utilizado para desenvolvimento web, desde sites simples até aplicações complexas. Como no nesse caso, Django é uma escolha comum para construir a camada web de aplicações que envolvem modelos preditivos.