

Predição de Séries Temporais

O que é?

Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo. (https://pt.wikipedia.org/wiki/S%C3%A9rie_temporal). Para a análise e predição de uma série temporal, a sequência dos dados é fator fundamental para sua utilização. Em uma série temporal, os dados vizinhos são dependentes e um dos objetivos é identificar estas relações de dependência. Podemos observar as séries temporais em diversas áreas, muitas destas áreas relacionadas a negócios. Exemplos: finanças, marketing, economia, seguros, ciências sociais, meteorologia, energia, medicina, engenharia, etc. “Uma série temporal é uma sequência de realizações (observações) de uma variável ao longo do tempo. Dito de outra forma, é uma sequência de pontos (dados numéricos) em ordem sucessiva, geralmente ocorrendo em intervalos uniformes. Portanto, uma série temporal é uma sequência de números coletados em intervalos regulares durante um período de tempo.” Para exemplificarmos esta última frase retirada do link acima, podemos mostrar a seguinte tabela:

tempo	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9
valor	10,8	9,99	7,34	8,68	9,43	12,01	10,07	7,9	14,01

A tabela acima mostra os valores em Reais de uma ação da bolsa de valores no decorrer de 9 dias consecutivos.

Podemos citar alguns exemplos de aplicações em negócios:

- Predição do consumo de ovos de páscoa para o próximo Natal;
- Predição do número de demissões em um determinado setor industrial;
- Predição de vendas de bilhetes aéreos para o próximo Carnaval;
- Predição de ocupação dos hotéis de uma cidade turística;
- Predição do número de pacientes em um pronto socorro na próxima sexta-feira de carnaval;
- Predição da quantidade necessária de caixas de um supermercado para atender os clientes em um final de semana;
- Predição de aluguel de carros no próximo Carnaval;
- Predição de estoque de medicamentos para uma determinada época do ano;
- Etc.

Qualquer variável monitorada ao longo do tempo é passível de predição. Quanto mais próximo dos valores históricos for a predição, maior o índice de acerto desta. Por outro lado, quando mais distante for a predição dos últimos dados armazenados, maior será o erro.

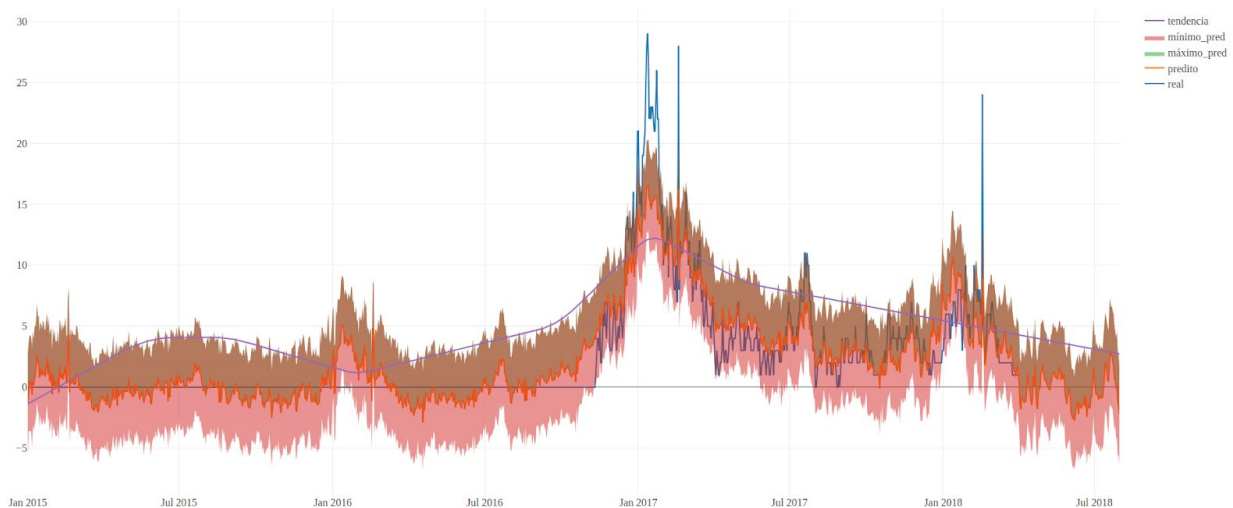
FERRAMENTAS

Para a predição de séries temporais dispomos de algumas ferramentas matemáticas sendo que as mais utilizadas são ARIMA (<https://pt.wikipedia.org/wiki/ARIMA>) e FACEBOOK PROPHET (<https://research.fb.com/prophet-forecasting-at-scale/>).

Aqui utilizaremos o Facebook Prophet.

Eventos que deixam as predições mais complexas

Algumas predições são, necessariamente, problemas de difícil solução tal como prever fenômenos da natureza. Fenômenos que dependem de aspectos psicológico-sociais tal como o valor de ações da bolsa de valores (uma simples palavra de algum político pode alterar significativamente o valor de uma ação) e fenômenos sazonais tal como eventos anuais (Carnaval, Natal, etc).



O gráfico acima mostra os valores preditos mínimos, máximo, tendência, valor predito e o valor real. Observem que para eventos sazonais, a predição não consegue acompanhar a curva real sem a necessária computação dos parâmetros.

Para instalarmos o Facebook Prophet devemos executar o seguinte comando:

pip install fbprophet

(<https://facebook.github.io/prophet/docs/installation.html#installation-in-python>)

Utilizando o Facebook Prophet

Esta ferramenta exige um formato de entrada bem definido que é : `ds['ds','y']`, ou seja, para utilizarmos o Prophet temos de entrar com um pandas dataframe com exatamente estas duas colunas sendo que `ds` é a data no padrão ISO e o `y` são os valores a serem preditos tais como cotação de ações, tamanho de frota, preço, etc.

	DS	Y
0	2007-12-10	9.590761
1	2007-12-11	8.519590
2	2007-12-12	8.183677
3	2007-12-13	8.072467
4	2007-12-14	7.893572

(https://facebook.github.io/prophet/docs/quick_start.html)

No código abaixo temos um exemplo de predição de passageiros de um aeroporto no decorrer dos anos.

```

1 import numpy as np
2 import pandas as pd
3 from fbprophet import Prophet
4 from matplotlib import pyplot
5
6
7 def faz_predicao(X, tempo_em_dias):
8     '''
9     Faz predição da serie temporal
10    :param X: Pandas Dataframe com [data,ds]
11    :param tempo_em_dias: dias a serem preditos para frente
12    :return: Arquivo com a predicao
13    '''
14    #df_national_holidays_events = pd.read_csv('national_holidays_events.csv')
15    # df_state_holidays_events = pd.read_csv('state_holidays_events.csv')
16    # df_city_holidays_events = pd.read_csv('city_holidays_events.csv')
17
18    #model = Prophet(seasonality_prior_scale=50, holidays_prior_scale=50, interval_width=0.95,
19    #                n_changepoints=150)
20    #model.add_seasonality(name='yearly', period=365, fourier_order=365)
21    #model.add_seasonality(name='monthly', period=30.5, fourier_order=30)
22    #model.add_seasonality(name='weekly', period=7, fourier_order=14)
23    model = Prophet()
24    model.fit(X)
25    future = model.make_future_dataframe(periods=tempo_em_dias, include_history=True)
26    future.tail()
27    forecast = model.predict(future)
28    forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
29    RMSE = np.sqrt(np.mean((forecast.loc[:, 'yhat'] - X['y']) ** 2))
30    forecast['RMSE'] = pd.Series(RMSE, index=forecast.index)
31    print('RMSE: %f' % np.sqrt(np.mean((forecast.loc[:, 'yhat'] - X['y']) ** 2)))
32    model.plot(forecast, uncertainty=True)
33    model.plot_components(forecast)
34    return forecast
35
36 def main():
37     # load
38     df = pd.read_csv('airline-passengers.csv')
39     ds = pd.DataFrame()
40     ds['ds'] = df['Month']
41     ds['y'] = df['Passengers']
42     previsao = faz_predicao(ds, 10)
43     print(previsao.columns)
44     print(previsao)
45
46 if __name__ == '__main__':
47     main()
48

```

Este código faz predição para 10 dias a frente.

INFO:fbprophet.forecaster:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
 INFO:fbprophet.forecaster:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
 Initial log joint probability = -2.46582

Iter	log prob	dx	grad	alpha	alpha0	# evals	Notes
99	401.66	0.000126253	82.0746	0.8557	0.8557	137	
191	401.842	0.000218722	100.877	3.114e-06	0.001	315	LS failed, Hessian reset
199	401.852	1.80114e-05	70.4554	1	1	325	
299	401.862	8.70985e-05	98.4062	1	1	454	
326	401.92	0.00068046	72.8916	8.808e-06	0.001	526	LS failed, Hessian reset
399	401.965	8.99555e-07	84.3993	1	1	628	
422	401.965	1.33566e-06	93.0775	2.084e-08	0.001	689	LS failed, Hessian reset
444	401.965	2.74061e-08	54.0265	1	1	715	

Optimization terminated normally:

Convergence detected: relative gradient magnitude is below tolerance

RMSE: 22.445723

Index(['ds', 'trend', 'yhat_lower', 'yhat_upper', 'trend_lower', 'trend_upper',
 'additive_terms', 'additive_terms_lower', 'additive_terms_upper',
 'yearly', 'yearly_lower', 'yearly_upper', 'multiplicative_terms',
 'multiplicative_terms_lower', 'multiplicative_terms_upper', 'yhat',
 'RMSE'],

O Erro Quadrático Médio (RMSE) é 22.44 ou aproximadamente, 22%.

Exercício

Entre no site da BOVESPA, baixe um arquivo de histórico das cotações e faça a predição. Tente minimizar o RMSE.