INSTITUTO FEDERAL
São Paulo
Câmpus Salto

Tópicos Avançados de Automação Industrial

TAAE9

Prof° José W. R. Pereira

Ciência de dados

Análise de Dados

A análise de dados é o **processo de manipulação de dados** através de **ferramentas computacionais e estatísticas**, de modo a buscar **informações relevantes** que auxiliam à **tomada de decisão.**

Tipos de análise de dados

Analista de Dados

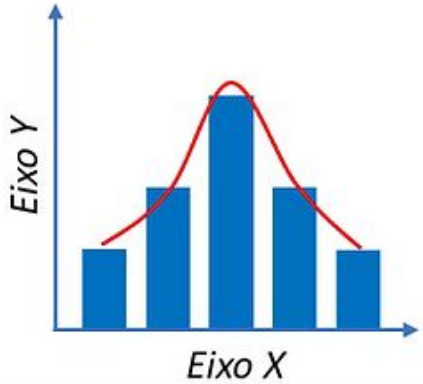
Cientista de Dados

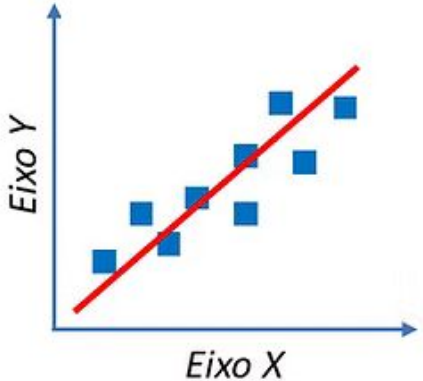
Descritiva

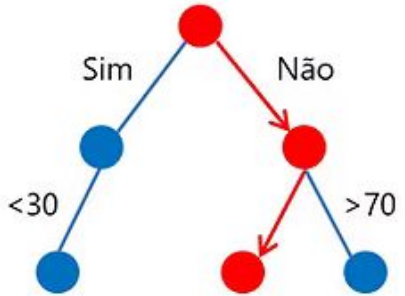
Diagnóstica

Preditiva

Prescritiva







Desenvolvimento da Análise

1. **Problema de negócio:**
    1. O resultado da ação foi positivo ou negativo?
    2. Por que os resultados foram esses?
    3. Como tomar uma decisão?

Desenvolvimento da Análise

2. **Descrever** o comportamento dentro de um escopo;
3. **Diagnosticar** os motivos que produziram os comportamentos;
4. **Prever** o comportamento baseado em histórico;
5. **Prescrever/orientar** ações para alcançar as metas do negócio.

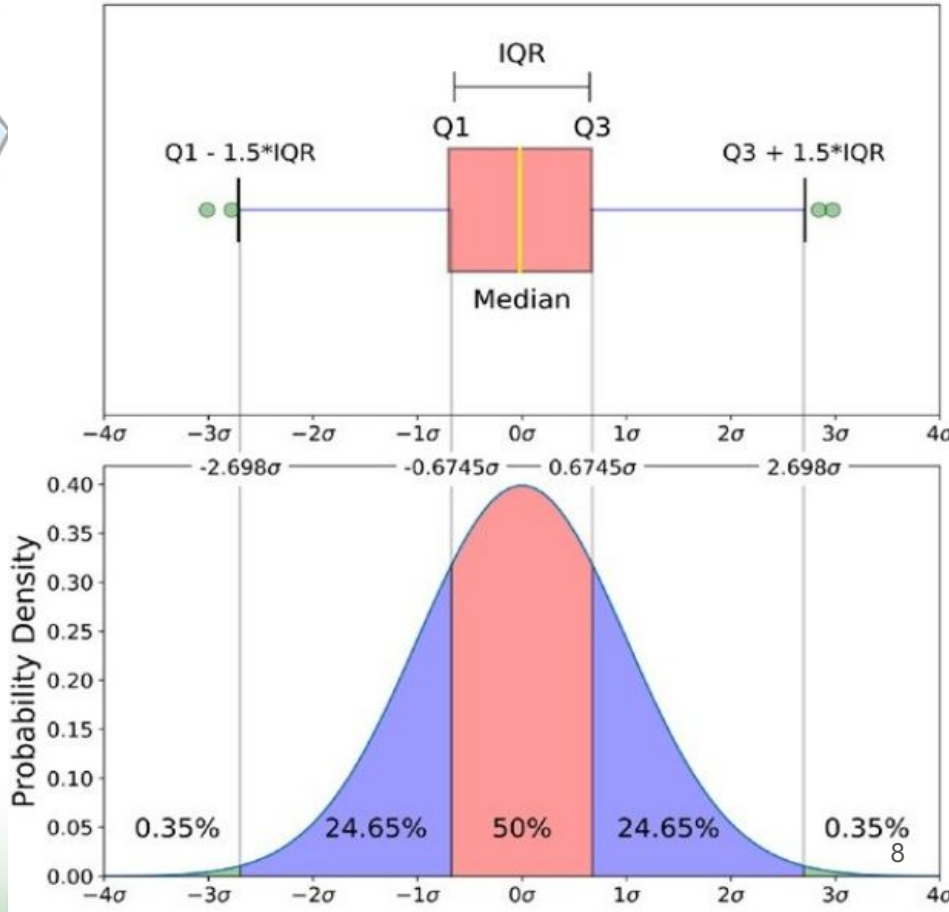
Análise Descritiva

Fornece um resumo simples de uma planilha de dados, através de **indicadores, gráficos e tabelas.**

Cia Aérea	Horário
Latam	11:10
Gol	11:15
Azul	11:20
Gol	12:40
Latam	12:45

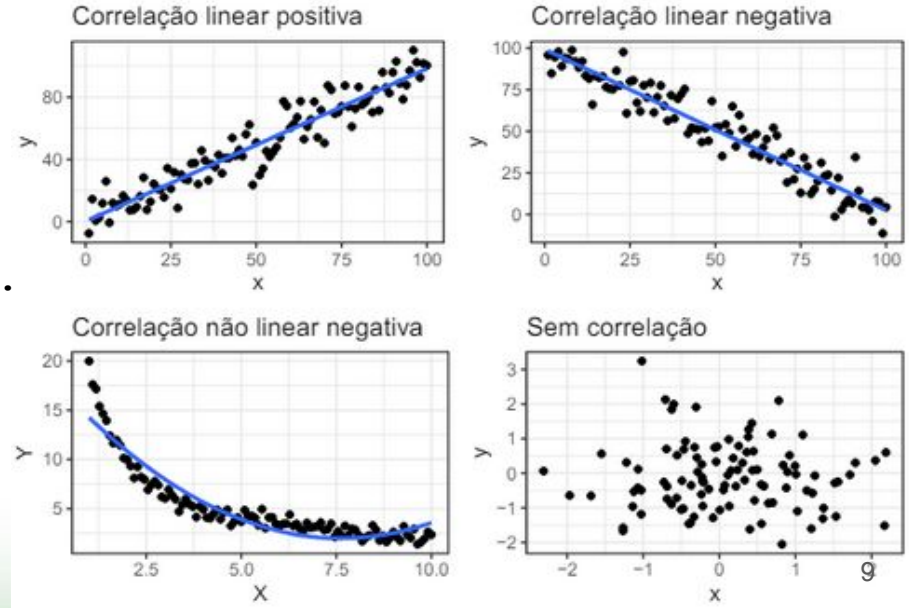
Análise Descritiva

Realiza a exploração inicial dos dados, permitindo a compreensão da **distribuição, valor central e dispersão dos dados**, além da presença de possíveis ***outliers***.



Características

- Resumo de dados:
  - Medidas de tendência central:
    - Média, Moda e Mediana;
  - Medidas de dispersão:
    - Desvio padrão e Variância.
- Visualização de dados:
  - Relação entre variáveis.
- Identificação de padrões.



Ferramentas

1. Planilha de cálculos: Excel, Google Sheets;
2. Softwares estatísticos: SPSS, Stata;
3. Software de visualização de dados: Tableu, Power BI
4. Linguagem de programação: Python, R

Análise Diagnóstica

Busca:

- **entender os motivos ou causas** por trás de determinado fenômeno ou resultado.
- **identificar padrões e relações** nos dados que podem explicar mudanças ou tendências observadas.

Características

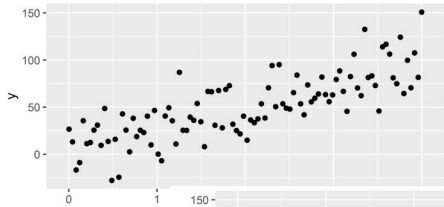
- Investigativa:
  - Buscando entender as causas subjacentes de um fenômeno ou problema.
- Baseada em Hipóteses:
  - Uma hipótese é proposta e a análise é usada para testar ela.

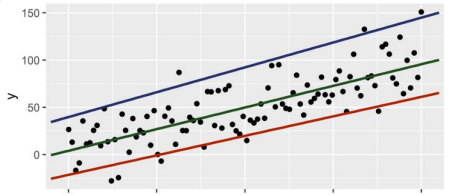
Características

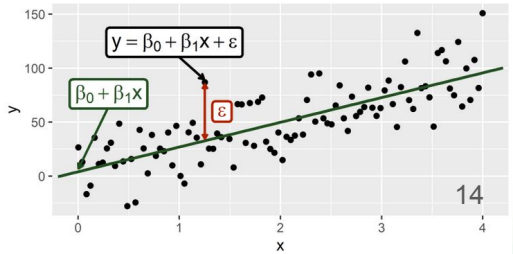
- Uso de dados históricos:
  - Identificar tendências, padrões e relações que podem explicar os resultados atuais.
- Análise Multivariada:
  - Pode envolver múltiplas variáveis ao mesmo tempo, com interações complexas que podem estar influenciando um resultado.
- Interpretação cuidadosa.

Ferramentas

- Análise de regressão:
  - Identificar a relação entre variáveis dependentes e independentes;
- Análise de correlação:
  - Determinar a relação entre duas ou mais variáveis;



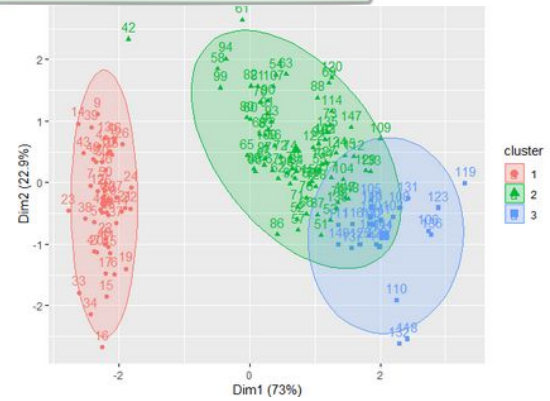


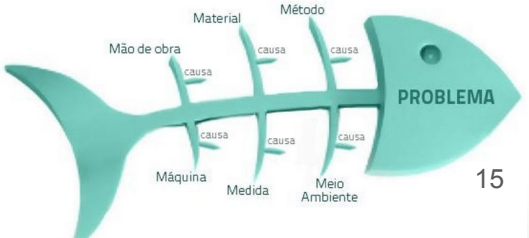


Ferramentas

Descritiva → **Diagnóstica** → Preditiva → Prescritiva

- Análise de *cluster*:
  - Agrupa por características semelhantes;
- Análise de causa e efeito (Diagrama de Ishikawa):
  - Proporciona uma visão ampla e identificação de possíveis causas de um problema.





Análise Preditiva

- Análise estatística que tem como objetivo prever resultados futuros com base em dados históricos e técnicas de modelagem.
- Coleta de dados >> modelo estatístico >> Previsão

Características

1. Baseada em modelos: Estatísticos e de **Machine Learning**.
2. Usa histórico de dados: Treinamento dos modelos.
3. Multivariável:
   Consideram a interação entre muitas variáveis.
4. Preditiva, mas não infalível:
   Sempre há um grau de incerteza.
5. Orientada para a ação:
   Resultados orientam a tomada de decisão.

Ferramentas

1. Python: Pandas, NumPy, Sci-kit Learn.
2. R: Análise estatística e gráficos.
3. SAS: *Statistical Analysis System* (financeiro e saúde).
4. SPSS: Software de fácil usabilidade e com ampla gama de ferramentas disponíveis.
5. Excel: Aplicado quando não se requer técnicas estatísticas avançadas.

Análise Prescritiva

- Fornece recomendações sobre o que deve ser feito, utilizando **técnicas avançadas** que levam em consideração uma **variedade de cenários** e **possíveis resultados**, para ajudar os **tomadores de decisão** a entender as implicações de diferentes cursos de ação.

Características

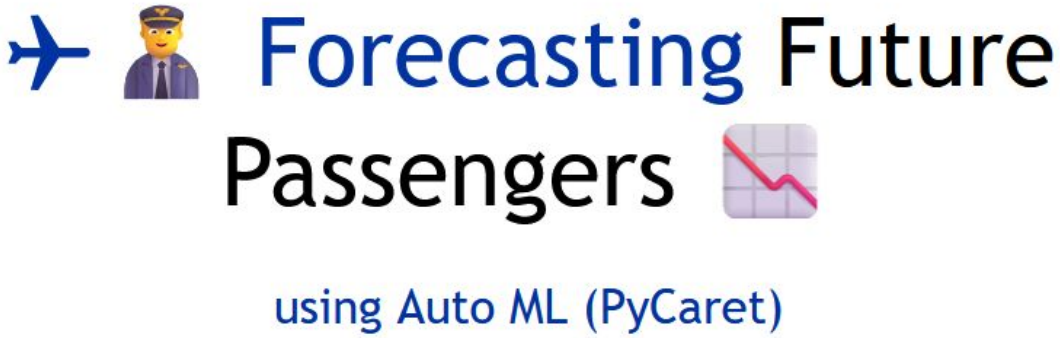
1. Recomendações de ação;
2. Consideração de diferentes cenários;
3. Otimização de operações;
4. Auxílio à tomada de decisão;
5. Adaptação a mudanças nas condições: pode incorporar novos dados à medida que se tornem disponíveis.

Ferramentas

1. Softwares de modelagem: Python, R, SAS, SPSS
2. Ferramentas de otimização: Gurobi, CPLEX
3. Ferramentas de simulação: Simul8, AnyLogic
4. Ferramentas de visualização de dados: Tableau, Power BI
5. Plataformas de machine learning: Azure ML, Amazon SageMaker

Previsão de passageiros futuros

Forecasting Future Passengers



Previsão de passageiros futuros
Forecasting Future Passengers

Problema:

é necessário prever passageiros futuros após 1960.

Previsão de passageiros futuros

Forecasting Future Passengers

Dataset Description 📃

👉 There are 2 variables in this data set namely:

- Month, and
- #Passengers.

👉 The following is the structure of the data set.

Variable Name	Description	Sample Data
Month	Date of records (yyyy-mm format).	1949-01; 1949-02; ...
#Passengers	Total air passengers	112; 118; ...

Previsão de passageiros futuros

Forecasting Future Passengers

PyCaret is an open-source machine learning package written in low-code that enables Data Scientists to automate their machine learning processes. It reduces the model experimentation process, allowing for the achievement of specific outcomes with less code.

Previsão de passageiros futuros
Forecasting Future Passengers

1.3 | Quick Overview of PyCaret Classification Module

The time series forecasting module in PyCaret (`pycaret.time_series.setup`) is a machine learning module that is used to handle time series analysis problems. With PyCaret, a data scientist/user can do forecasting with several models, namely:

- Seasonal Naive Forecaster,
- Exponential Smoothing,
- ARIMA,
- Polynomial Trend Forecaster,
- K Neighbors w/ Cond. Deseasonalize & Detrending,
- Linear w/ Cond. Deseasonalize & Detrending,
- Elastic Net w/ Cond. Deseasonalize & Detrending,
- Ride w/ Cond. Deseasonalize & Detrending,
- Lasso Net w/ Cond. Deseasonalize & Detrending,
- Extreme Gradient Boosting w/ Cond. Deseasonalize & Detrending, and more.

Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Importing Libraries ---
import datetime
import numpy as np
import pandas as pd
import warnings
import pycaret
import kaleido
import plotly.express as px

from pycaret.time_series import *
from pycaret.utils import enable_colab

# --- Libraries Settings ---
warnings.filterwarnings('ignore')
enable_colab()
```

Previsão de passageiros futuros

Forecasting Future Passengers

```
# --- Reading Dataset ---
df = pd.read_csv('../input/air-passengers-forecast-dataset/AirPassengers.csv')
df.head().style.background_gradient(cmap='Blues').set_properties(**{'font-family': 'Segoe UI'}).h
ide_index()
```

Month	#Passengers
1949-01	112
1949-02	118
1949-03	132
1949-04	129
1949-05	121

```
.: Imported Dataset Info :.
*****************************
Total Rows: 144
Total Columns: 2
*****************************
```

```
.: Dataset Details :.
*****************************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144 entries, 0 to 143
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Month        144 non-null    object
 1   #Passengers  144 non-null    int64
dtypes: int64(1), object(1)
```

Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Descriptive Statistics
df.describe().T.style.background_gradient(cmap='GnBu').set_properties(**{'font-family': 'Segoe U
I'})
```

	count	mean	std	min	25%	50%	75%	max
#Passengers	144.000000	280.298611	119.966317	104.000000	180.000000	265.500000	360.500000	622.000000

Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Change `Month` Column Type to datetime ---
df['Month'] = pd.to_datetime(df['Month'])
```

```
# --- Set `Month` Column as Index ---
df.set_index('Month', inplace=True, drop=True)
df.head(3).style.background_gradient(cmap='Blues').set_properties(**{'font-family': 'Segoe U
I'}).hide_index()
```

#Passengers
112
118
132

Previsão de passageiros futuros
Forecasting Future Passengers

5. | PyCaret Setup ⚙

👉 This section will implement PyCaret by calling `TimeSeriesExperiment()` function.

👉 For experiment purposes, the number of folds that used in cross validation will be set to 3 and the forecast horizon used will be set to 12 (last 12 points in the dataset will be set as test).

```
# --- Setup PyCaret ---
s = setup(df, fh = 12, fold = 3, session_id = 123)
```

Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Setup PyCaret ---
s = setup(df, fh = 12, fold = 3, session_id = 123)
```

	Description	Value
0	session_id	123
1	Target	#Passengers
2	Approach	Univariate
3	Exogenous Variables	Not Present
4	Data shape	(144, 1)
5	Train data shape	(132, 1)
6	Test data shape	(12, 1)
7	Fold Generator	ExpandingWindowSplitter
8	Fold Number	3
9	Enforce Prediction Interval	False
10	Seasonal Period(s) Tested	12

Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Perform Statistical Test ---
check_stats()
```

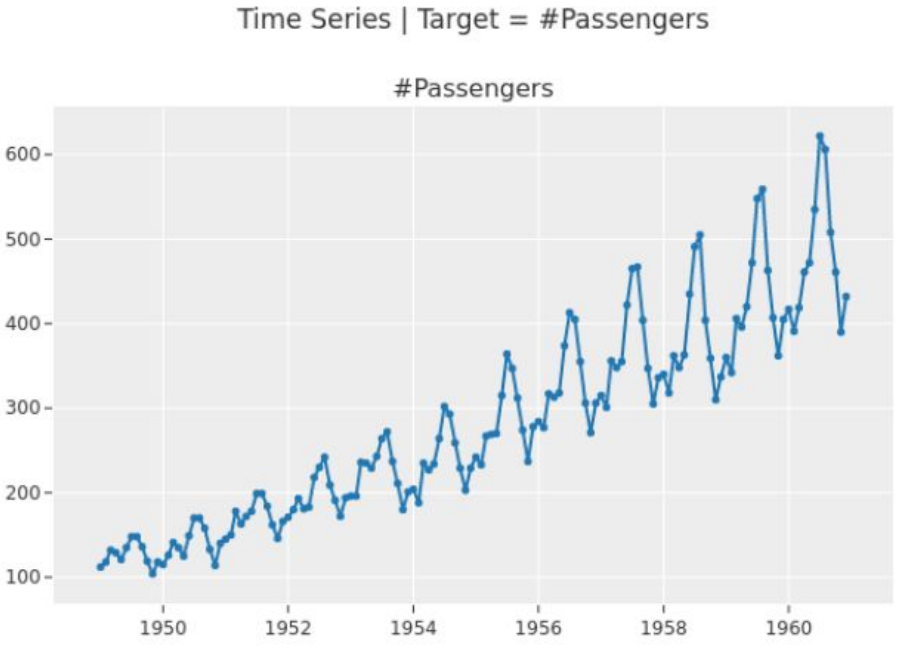
	Test	Test Name	Data	Property	Setting	Value
0	Summary	Statistics	Actual	Length		144.0
1	Summary	Statistics	Actual	Mean		280.298611
2	Summary	Statistics	Actual	Median		265.5
3	Summary	Statistics	Actual	Standard Deviation		119.966317
4	Summary	Statistics	Actual	Variance		14391.917201
5	Summary	Statistics	Actual	Kurtosis		-0.364942
6	Summary	Statistics	Actual	Skewness		0.58316

👉 From the statistical test results above, it can be concluded that:

- The series is not stationary (ADF p-value more than 0.05)
- The series is not adequate (Ljung-Box p-value less than 0.05)

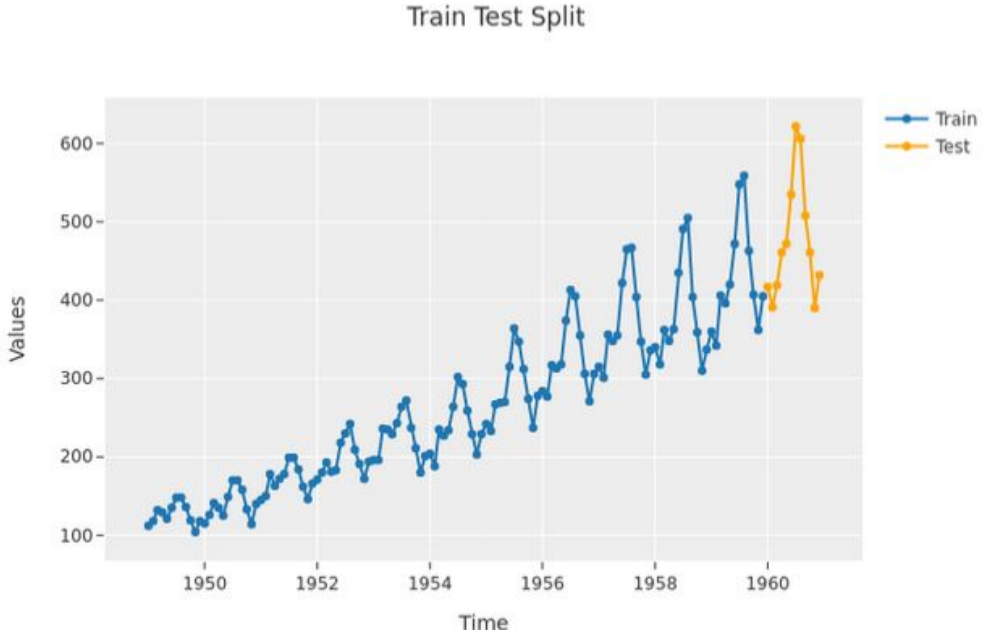
Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Time Series Plot ---
plot_model(plot = 'ts', fig_kwargs = {'hoverinfo': 'none', 'big_data_threshold': 15})
```



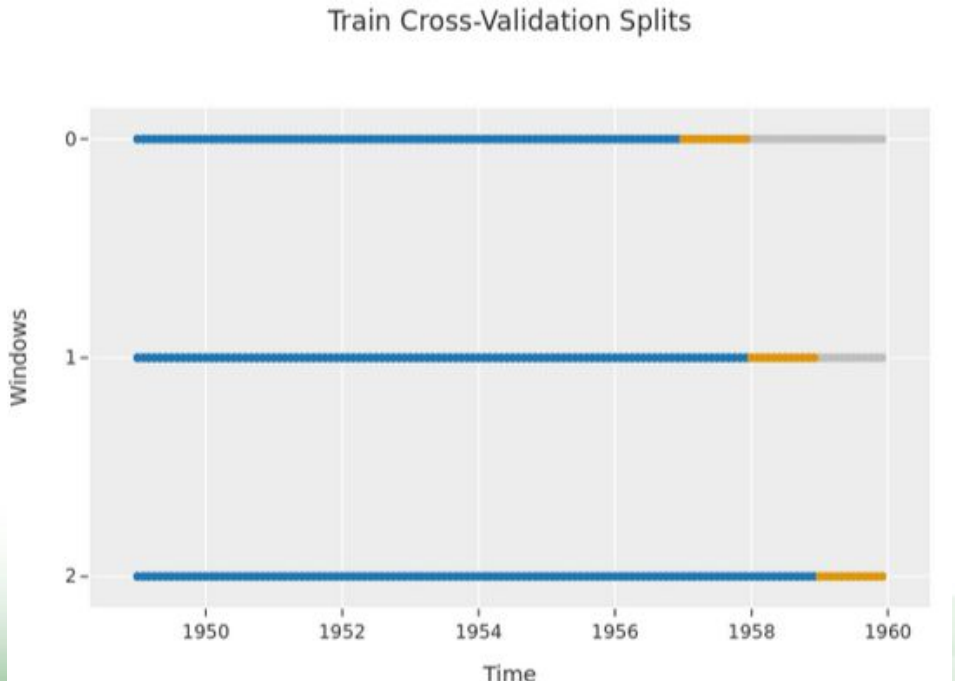
Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Train & Test Plot ---
plot_model(plot = 'train_test_split', fig_kwargs = {'hoverinfo': 'none', 'big_data_threshold': 15})
```



Previsão de passageiros futuros
Forecasting Future Passengers

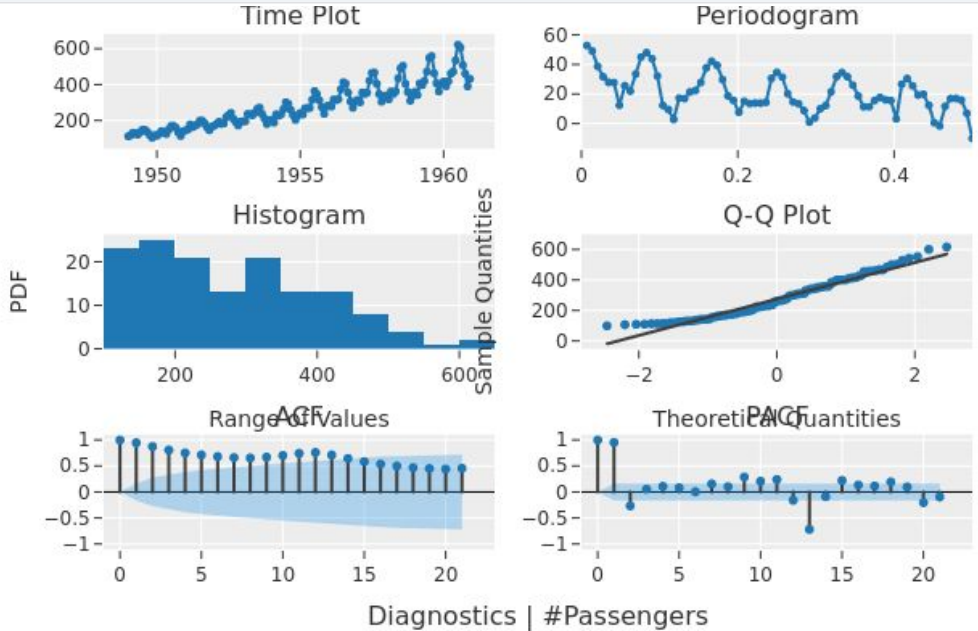
```
# --- CV Plot ---
plot_model(plot = 'cv', fig_kwargs = {'hoverinfo': 'none', 'big_data_threshold': 15})
```



Previsão de passageiros futuros

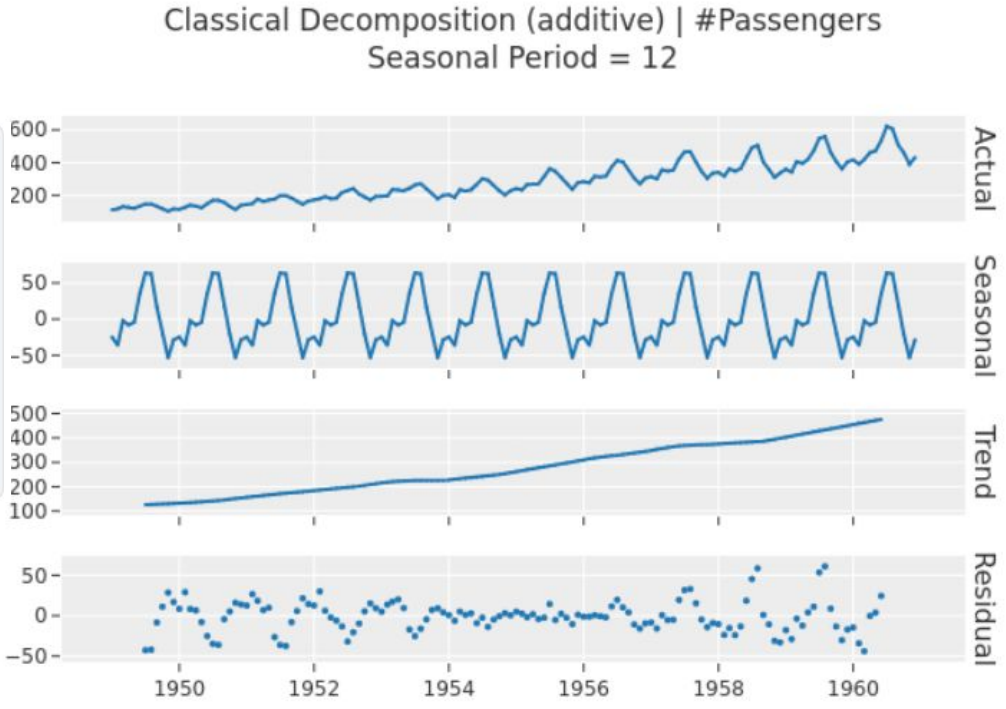
Forecasting Future Passengers

```
# --- Diagnostic Plot ---
plot_model(plot = 'diagnostics', fig_kwargs = {'hoverinfo': 'non
e', 'big_data_threshold': 15})
```



Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Showing Decomposition Plots ---
plot_model(plot = 'decomp', fig_kwargs = {'hoverinfo': 'none','big_data_threshold': 15})
plot_model(plot = 'decomp', data_kwargs={'type': 'multiplicative'},
           fig_kwargs = {'hoverinfo': 'none','big_data_threshold': 15})
plot_model(plot = 'decomp_stl', fig_kwargs = {'hoverinfo': 'none','big_data_threshold': 15})
```



Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Time Series Models ---
models()
```

	Name	Reference	Turbo
ID			
naive	Naive Forecaster	sktime.forecasting.naive.NaiveForecaster	True
grand_means	Grand Means Forecaster	sktime.forecasting.naive.NaiveForecaster	True
snaive	Seasonal Naive Forecaster	sktime.forecasting.naive.NaiveForecaster	True
polytrend	Polynomial Trend Forecaster	sktime.forecasting.trend.PolynomialTrendForeca...	True
arima	ARIMA	sktime.forecasting.arima.ARIMA	True
auto_arima	Auto ARIMA	sktime.forecasting.arima.AutoARIMA	True
exp_smooth	Exponential Smoothing	sktime.forecasting.exp_smoothing.ExponentialSm...	True
croston	Croston	sktime.forecasting.croston.Croston	True
ets	ETS	sktime.forecasting.ets.AutoETS	True
theta	Theta Forecaster	sktime.forecasting.theta.ThetaForecaster	True
tbats	TBATS	sktime.forecasting.tbats.TBATS	False
bats	BATS	sktime.forecasting.bats.BATS	False
lr_cds_dt	Linear w/ Cond. Deseasonalize & Detrending	pycaret.containers.models.time_series.BaseCdsD...	True
en_cds_dt	Elastic Net w/ Cond. Deseasonalize & Detrending	pycaret.containers.models.time_series.BaseCdsD...	True

Previsão de passageiros futuros
Forecasting Future Passengers

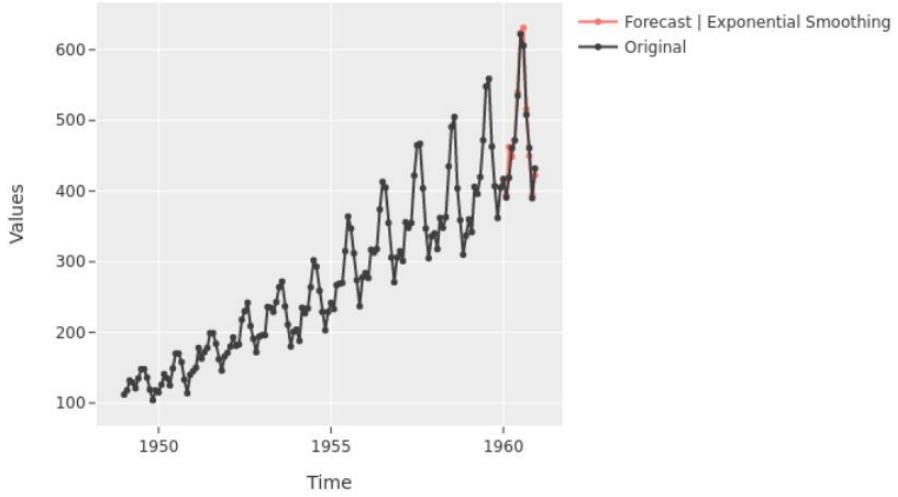
```
# --- Compare Models ---
best = compare_models()
```

	Model	MAE	RMSE	MAPE	SMAPE	MASE	RMSSE	R2	TT (Sec)
exp_smooth	Exponential Smoothing	17.1926	20.1633	0.0435	0.0439	0.5852	0.6105	0.8918	0.0933
et_cds_dt	Extra Trees w/ Cond. Deseasonalize & Detrending	19.4653	24.1050	0.0484	0.0484	0.6602	0.7288	0.8459	0.5300
huber_cds_dt	Huber w/ Cond. Deseasonalize & Detrending	20.0334	25.9670	0.0491	0.0499	0.6813	0.7866	0.8113	0.0400
arima	ARIMA	20.0069	22.2199	0.0501	0.0507	0.6830	0.6735	0.8677	0.4833
catboost_cds_dt	CatBoost Regressor w/ Cond. Deseasonalize & Detrending	20.9112	26.8907	0.0505	0.0509	0.7106	0.8146	0.8085	1.5933
ridge_cds_dt	Ridge w/ Cond. Deseasonalize & Detrending	20.6086	25.4405	0.0509	0.0514	0.7004	0.7703	0.8215	0.0300

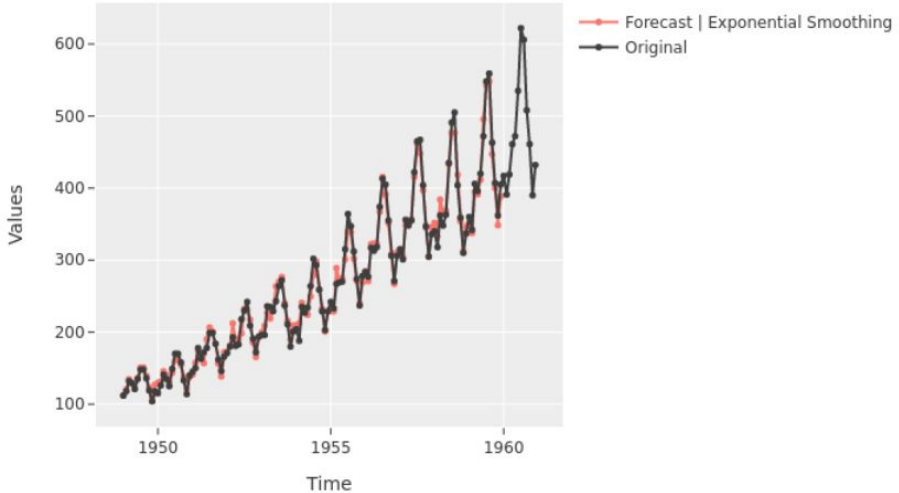
Previsão de passageiros futuros
Forecasting Future Passengers

```
# --- Plot Forecasting Performance & Insample ---
plot_model(best, plot = 'forecast', fig_kwargs = {'hoverinfo': 'none','big_data_
threshold': 15})
plot_model(best, plot = 'insample', fig_kwargs = {'hoverinfo': 'none','big_data_
threshold': 15})
```

Actual vs. 'Out-of-Sample' Forecast | #Passengers

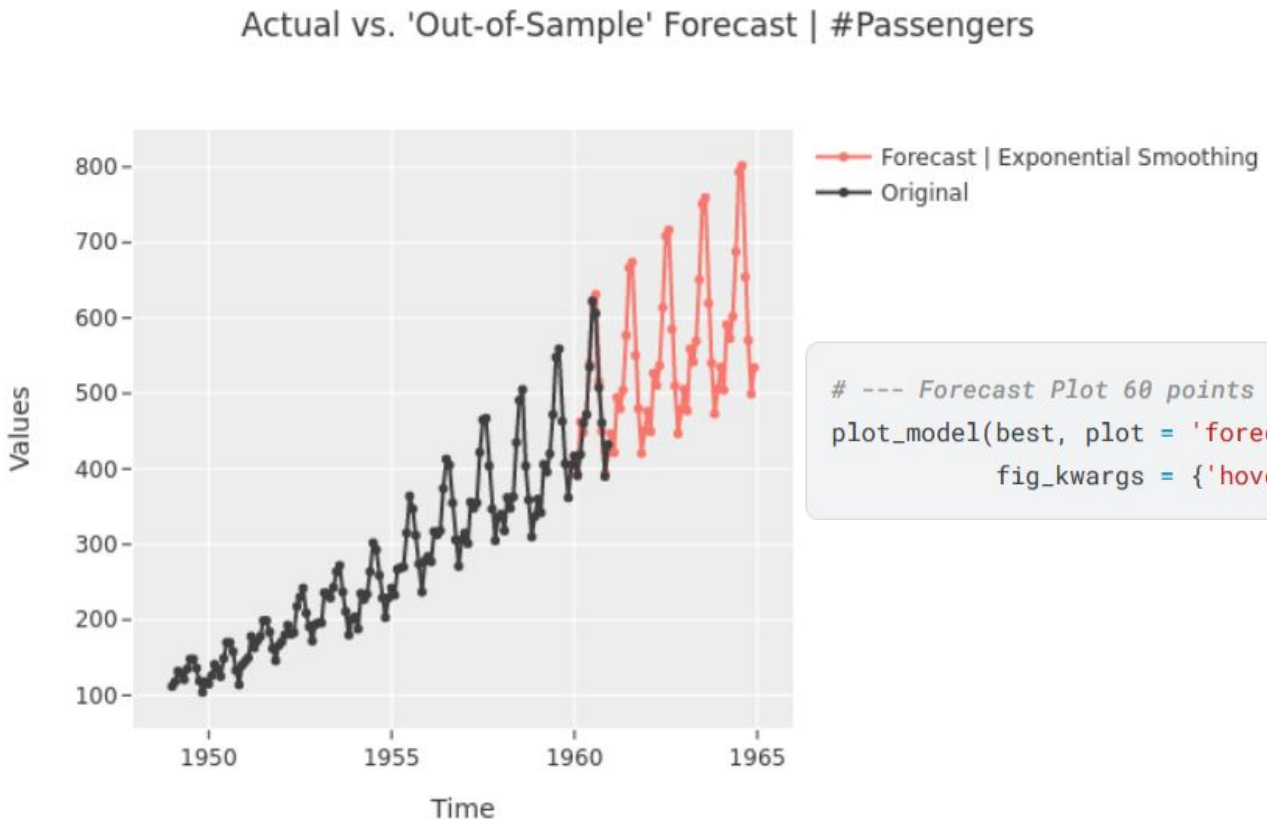


Actual vs. 'In-Sample' Forecast | #Passengers



Previsão de passageiros futuros

Forecasting Future Passengers




```
# --- Forecast Plot 60 points ---
plot_model(best, plot = 'forecast', data_kwargs = {'fh': 60},
           fig_kwargs = {'hoverinfo': 'none', 'big_data_threshold': 15})
```

Previsão de passageiros futuros
Forecasting Future Passengers

6. | Saving Model 💾

👉 Since exponential smoothing is the best forecasting model, will save the model in pickle (`.pkl`) file for future production.

```
# --- Save Model in Pickle File ---
save_model(best, 'exp_smooth_final_model')
```

INSTITUTO FEDERAL
São Paulo
Câmpus Salto

Tópicos Avançados de Automação Industrial

Prof° José W. R. Pereira
jose.pereira@ifsp.edu.br
josewrpereira.github.io/docs