Data Collection: Gather historical sales data. This data should include information about sales over time, such as dates, product details, prices, and any relevant external factors (e.g., holidays, promotions, economic indicators).

Data Preprocessing: Clean and prepare your data. This involves handling missing values, outliers, and converting data into a suitable format for analysis.

Feature Engineering: Create meaningful features from your data. For sales prediction, this might include creating time-based features (e.g., day of the week, month, season), lag features (e.g., previous month’s sales), and any domain-specific features.

Data Splitting: Split your data into training and testing sets. Typically, you’d use the training data to train your machine learning model and the testing data to evaluate its performance.

Model Selection: Choose a machine learning algorithm suitable for time-series forecasting. Common choices include linear regression, decision trees, random forests, or more advanced models like ARIMA, LSTM, or Prophet.

Model Training: Train your chosen model using the training data. Ensure you handle time-series data properly, considering issues like autocorrelation and seasonality.

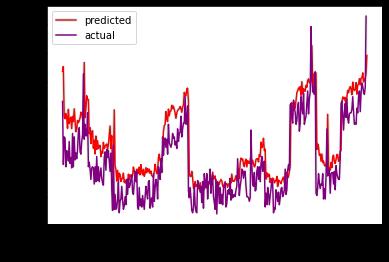
Hyperparameter Tuning: Optimize your model’s hyperparameters to improve its performance. You can use techniques like grid search or random search.



Model Evaluation: Evaluate your model’s performance using appropriate metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE)

Prediction: Use your trained model to make future sales predictions. You can use this to forecast sales for the desired time period.

Visualization: Visualize your predictions and compare them to actual sales data. Matplotlib or Seaborn can be helpful for this.

Deployment: If needed, deploy your model in a production environment to make real-time predictions.

Monitoring and Maintenance: Continuously monitor your model’s performance and update it as necessary, especially if your sales data changes significantly over time.

Remember that the specific implementation details and tools may vary depending on your project’s requirements and the complexity of your data. You may also consider incorporating external factors or economic indicators for more accurate predictions.

**Future sales prediction code:**

Import numpy as np

Import pandas as pd, datetime

Import seaborn as sns

From statsmodels.tsa.stattools import adfuller

Import matplotlib.pyplot as plt

Get\_ipython().run\_line\_magic(‘matplotlib’, ‘inline’)

From time import time

Import os

From math import sqrt

From statsmodels.tsa.seasonal import seasonal\_decompose

From statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf

Import itertools

Import statsmodels.api as sm

From statsmodels.tsa.stattools import acf,pacf

From statsmodels.tsa.arima\_model import ARIMA

From sklearn import model\_selection

From sklearn.metrics import mean\_squared\_error, r2\_score

From pandas import DataFrame

Import xgboost as xgb

From fbprophet import Prophet

Import warnings

Warnings.filterwarnings(‘ignore’)

Store = pd.read\_csv(‘../input/rossmann-store-sales/store.csv’)

* Train = pd.read\_csv(‘../input/rossmann-store-sales/train.csv’, index\_col=’Date’, parse\_dates=True’