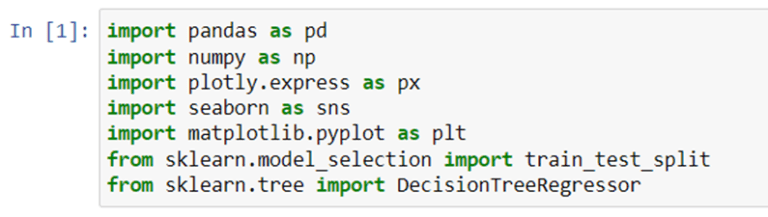
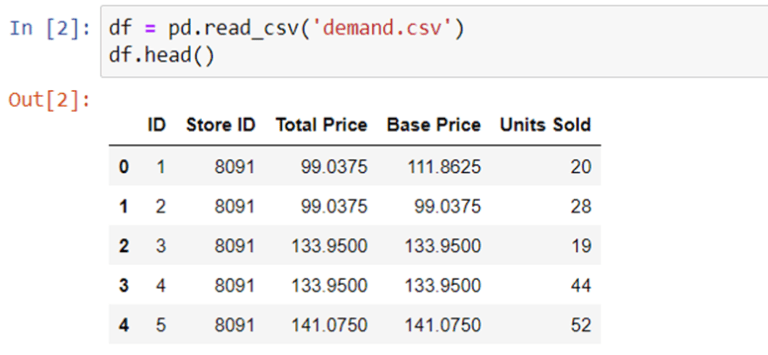
**PRODUCT DEMAND PREDICTION IN MACHINE LEARNING.**

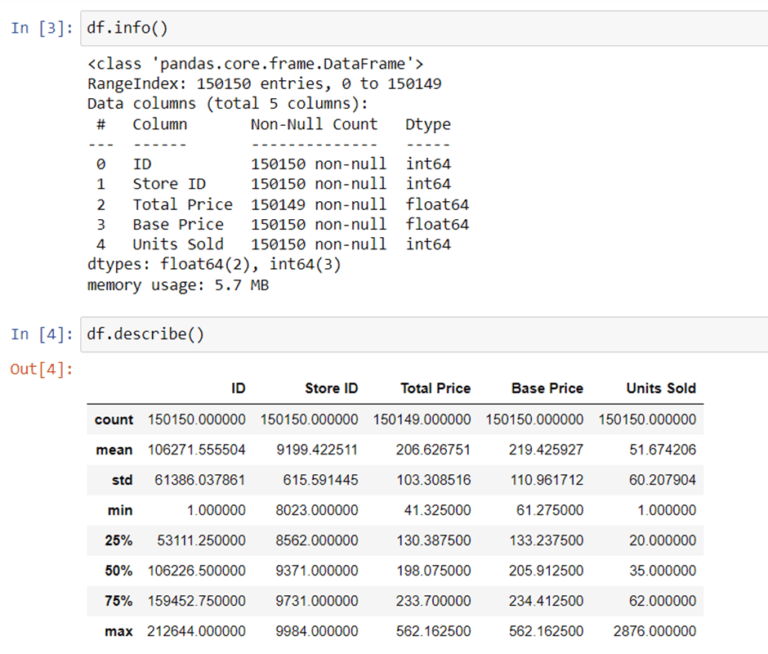
product company plans to offer discounts on their products during the upcoming holiday season. The company wants to find a price that makes its product cheaper than its competitors. For this task, the company provided records of historical sales changes based on price changes. I need to train a model that can predict the demand for a product in markets with different price points.

I hope you have understood what kind of question sentences you can get in the product demand forecasting task. In the next section, I’ll show you how to predict product demand with machine learning using Python.

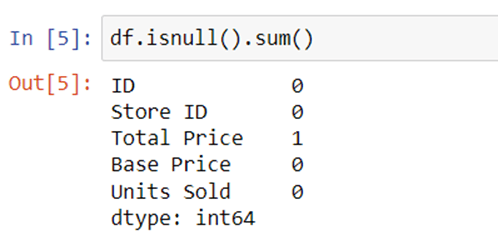
**Let’s start by importing the Python libraries and datasets required for the product demand forecasting task.**



Let’s get some information and description about our dataset.

Let’s check whether our data contains Null values or not.

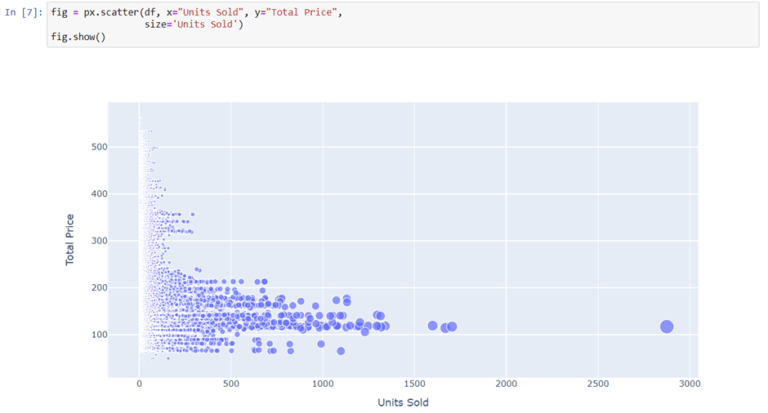
Let’s check whether our data contains Null values or not.



**, since there is only one value in the “Total Price” column for this record, here we delete the entire row.**



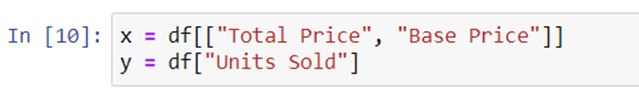
**Now let’s analyze the relationship between product price and demand. Here we use a scatterplot to see how demand for a product change with price changes.**



can be seen that, with a few exceptions, most of the data points indicate that product sales increase as prices decrease. **Now let’s look at the correlations between the characteristics of the dataset.**

## **Modelling**

Now let’s move on to the task of training a machine learning model to predict demand for a product at various prices. Select the ‘Total Price’ and ‘Base Price’ columns as the functions to train the model on, and the ‘Units Sold’ column as the labels for the model.



Now let’s split the data into training and test sets and use the decision tree regression algorithm to train our model:

xtrain, xtest, ytrain, ytest = train\_test\_split(x, y,

test\_size=0.2,

random\_state=42)

from sklearn.tree import DecisionTreeRegressor

model = DecisionTreeRegressor()

model.fit(xtrain, ytrain)

Now let’s input the features **(Total Price, Base Price)** into the model and predict how much quantity can be demanded based on those values:

#features = [["Total Price", "Base Price"]]

features = np.array([[133.00, 140.00]])

model.predict(features)

**array([27.])**

**CONCLUSION:**

1. **Data Quality Is Crucial:** High-quality historical data is essential for accurate demand prediction. Ensure your data is clean, complete, and representative of the market conditions.
2. **Feature Engineering Matters:** The choice and engineering of features (variables) play a significant role in the accuracy of your model. Factors like seasonality, promotions, and external events can impact demand.
3. **Model Selection:** Various machine learning models can be used for demand prediction, including linear regression, time series models, and more complex algorithms like random forests or deep learning. The choice of model depends on your specific problem and data.
4. **Evaluation Metrics:** Select appropriate evaluation metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) to measure the performance of your model.
5. **Time Series Analysis:** If demand data exhibits time-dependent patterns, consider using time series analysis techniques such as ARIMA or Prophet.
6. **Demand Forecast Horizon:** The prediction horizon (short-term or long-term) can impact the choice of model and the quality of predictions.
7. **Data Scaling:** Depending on the model, you may need to scale or normalize your data to ensure the features have similar ranges.
8. **Cross-Validation:** Use cross-validation techniques to assess your model's generalization performance and to avoid overfitting.
9. **Ensemble Methods:** Ensemble methods like bagging and boosting can improve prediction accuracy by combining multiple models.
10. **Domain Knowledge:** Incorporating domain expertise can help fine-tune models and interpret results effectively.
11. **Continuous Monitoring:** Product demand can change over time, so it's essential to retrain your model periodically to adapt to new data.
12. **External Factors:** Consider external variables like economic indicators, weather, or social events that may impact demand and incorporate them into your model if relevant.
13. **Interpretability:** Understand and communicate the factors that drive your predictions, especially if you need to make strategic decisions based on them.
14. **Communication:** Effective communication of predictions to stakeholders is critical to leverage the insights from machine learning for decision-making.
15. **Sensitivity Analysis:** Assess how sensitive your predictions are to changes in input variables and model parameters.
16. **Cost-Benefit Analysis:** Evaluate the costs and benefits of implementing a demand prediction model and measure its impact on inventory management, production planning, and overall business performance.