GRT INSTITUTE OF
ENGINEERING AND
TECHNOLOGY, Tiruttani

1103-GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PHASE 2

PROJECT TITLE

PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

I PRAMEELA
Collage code : 1103
3rd yr, 5th sem
Reg no.:110321104015
prameela0430@gmail.co

2.1 SHORT EXPLANATION ABOUT PRODUCT DEMAND

PREDICTION WITH MACHINE LEARNING

Product demand prediction with machine learning involves using algorithms and historical data to forecast how many units of a product will be sold in the future. It's a crucial task for businesses to optimize inventory, production, and sales strategies. Here's a simplified overview of the process:

1. Data Collection : Gather historical data, including sales records, pricing information, promotional activities, and other relevant factors.

2. Data Preprocessing : Clean and prepare the data by handling missing values, encoding categorical variables, and scaling numerical features.

3. Feature Engineering : Create new features or transform existing ones to capture patterns and seasonality in the data.

4. Model Selection : Choose a machine learning model or algorithm (e.g., Linear Regression, Decision Trees, or more advanced models like XGBoost) to make predictions.

5. Training : Train the selected model using the historical data, with product attributes and external factors as input features and past sales (demand) as the target variable.

6. Evaluation : Assess the model's performance by using metrics like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE) on a separate test dataset.

7. Hyperparameter Tuning : Fine-tune the model's parameters to improve prediction accuracy.

8. Deployment : Deploy the trained model in a production environment to generate real-time predictions or automate demand forecasting.

9. Monitoring and Maintenance : Continuously monitor the model's performance and update it as demand patterns change.

Product demand prediction with machine learning helps businesses make informed decisions, reduce excess inventory costs, avoid stockouts, and enhance customer satisfaction by ensuring products are available when and where they are needed

2.2 WHERE I GOT THE DATASETS AND ITS DETAILS

You can find datasets for customer segmentation and various other data science projects from several reputable sources.

KAGGLE : Kaggle is a popular platform for data science competitions and dataset sharing. It hosts a wide range of datasets on various topics, including customer data. You can browse datasets, read their descriptions, and download them for free. Kaggle also provides a community where you can discuss and collaborate on data science projects.

Website : https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning

NAME OF THE DATASET : PRODUCT DEMAND PREDICTION WITH MACHINE LEARNING

DATA DESCRIPTION :

Customer segmentation is a common application of data science in the retail industry, including malls. To perform customer segmentation effectively, you need relevant data about mall customers. Once you have collected and cleaned the relevant data, you can apply various data science techniques such as clustering, classification, and regression to segment mall customers effectively. The goal is to identify groups of customers with similar characteristics and preferences to tailor marketing strategies, promotions, and store layouts to meet their needs and maximize the mall's revenue.

2.3 DETAILS ABOUT COLUMNS

1. Id: This column typically represents a unique identifier for each product or record in your dataset. It's a reference that helps distinguish one product from another.

2. Store Id: This column likely contains an identifier for the store or location where the product was sold or where the data was collected. It helps you associate each product with a specific store.

3. Total Price: This column contains the total price of the product when it was sold. It could represent the final price paid by the customer, inclusive of any discounts, taxes, or additional charges.

4. Base Price: The base price is the original or regular price of the product before any discounts or promotions. This column provides insight into how the product's pricing may have been altered.

5. Units Sold: This column records the number of units or items of the product that were sold during a specific transaction or time period. It's a critical measure of product demand, as it indicates how many customers purchased the product.

2.4 DETAILS OF LIBRARIES TO BE USED AND WAY TO DOWNLOAD

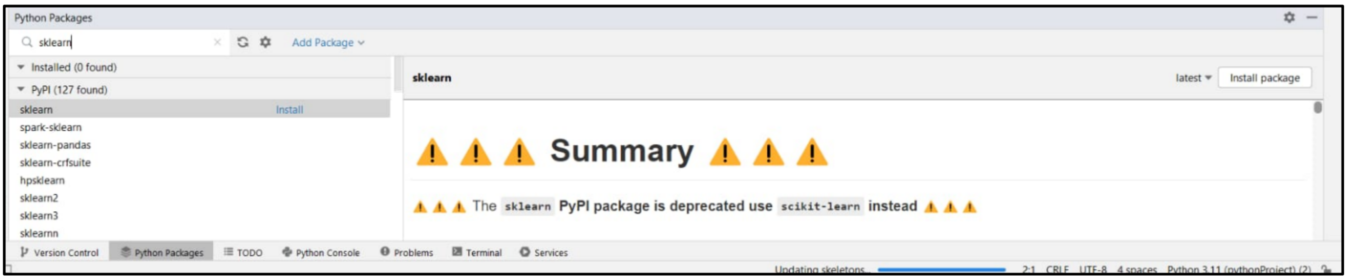
LIBRARIES TO BE USED

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
# Load your dataset
data = pd.read_csv('your_dataset.csv')
# Define the features (X) and the target (y)
```

```
X = data[['Store Id', 'Total Price', 'Base Price']]
y = data['Units Sold']
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Initialize and train the model (Linear Regression)
model = LinearRegression()
model.fit(X_train, y_train)
```

WAY TO DOWNLOAD THE LIBRARIES

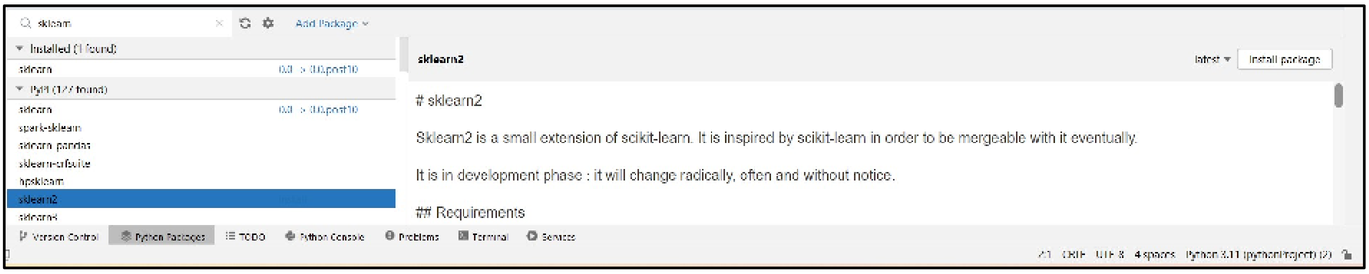
1.Click the python packages in the bottom of your project in pycharm



⚠ ⚠ ⚠ Summary ⚠ ⚠ ⚠

⚠ ⚠ ⚠ The `sklearn` PyPI package is deprecated use `scikit-learn` instead ⚠ ⚠ ⚠

2.Type the required library in the search box and click install package in the right end top of the python packages.



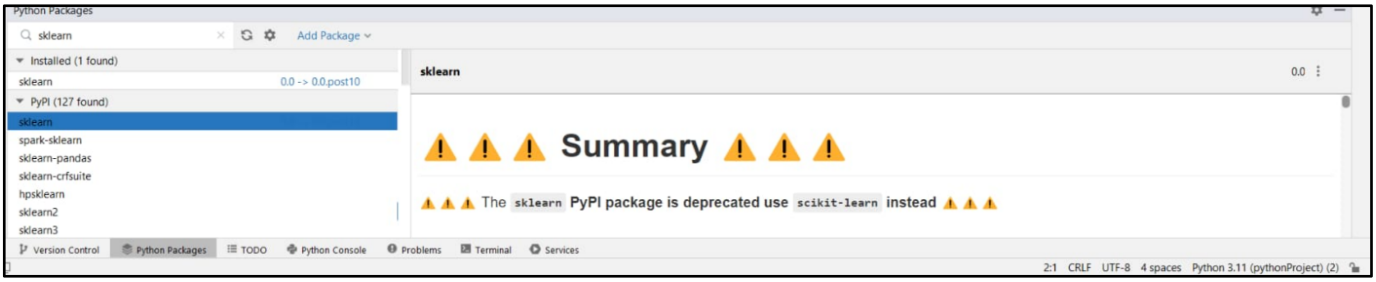
sklearn2

Sklearn2 is a small extension of scikit-learn. It is inspired by scikit-learn in order to be mergeable with it eventually.

It is in development phase : it will change radically, often and without notice.

Requirements

3.After installation process finished it shows the package was installed in the python packages.



⚠ ⚠ ⚠ Summary ⚠ ⚠ ⚠

⚠ ⚠ ⚠ The `sklearn` **PyPI package is deprecated use** `scikit-learn` **instead** ⚠ ⚠ ⚠

2.5 HOW TO TRAIN AND TEST THE DATASET

Training and testing a product demand prediction model with machine learning involves the following steps:

1. Data Preparation:

- Gather historical data on product sales, pricing, promotions, and relevant features.

- Preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features.

2. Data Splitting:

- Split your dataset into two parts: a training set and a testing set. A common split is 80% for training and 20% for testing, but the exact split ratio may vary based on your dataset size and requirements.

3. Feature Selection:

- Choose the relevant features (input variables) for your model, such as 'Store Id', 'Total Price', 'Base Price', and any others that may impact product demand. Ensure these features are available in both the training and testing sets.

4. Model Selection:

- Choose a machine learning algorithm suitable for demand prediction. Common choices include Linear Regression, Decision Trees, Random Forest, XGBoost, or neural networks for more complex tasks.

5. Model Training:

- Train the selected model using the training dataset. Use the chosen features ('X_train') and the target variable ('y_train').

6. Model Testing:

- Use the trained model to make predictions on the testing dataset. Pass the testing features ('X_test') to the model to obtain demand predictions ('y_pred').

7. Evaluation:

- Compare the predicted demand values ('y_pred') with the actual demand values from the testing dataset ('y_test').

- Use evaluation metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), or others to assess the model's performance.

8. Tuning and Optimization:

- If the model's performance is not satisfactory, consider hyperparameter tuning and feature engineering to improve predictions.

9. Deployment:

- Once you are satisfied with the model's performance, you can deploy it in a production environment to make real-time demand predictions.

10. Monitoring and Maintenance:

- Continuously monitor the model's performance in production. Update the model as necessary to account for changing demand patterns or external factors.

The process involves iterating through these steps, fine-tuning your model, and possibly exploring different machine learning algorithms to improve the accuracy of product demand predictions.

2.6 REST OF EXPLANATION

Certainly, here's the continuation of the explanation for product demand prediction with machine learning:

1. Hyperparameter Tuning :

- Depending on the model you choose, you may need to fine-tune its hyperparameters. Hyperparameters are settings that are not learned during training but can significantly impact the model's performance. Techniques like grid search or random search can help you find the best hyperparameters for your model.

2. Cross-Validation :

- To ensure your model's performance is robust, use cross-validation techniques like k-fold cross-validation. It involves splitting your data into multiple subsets and training/evaluating the model on different combinations. This helps you get a more reliable estimate of your model's performance.

3. Feature Engineering:

- Explore feature engineering techniques to create new features or transform existing ones. For instance, you can extract time-based features, create lag features, or engineer interaction features to capture complex relationships in your data.

4. Model Selection:

- If you're not satisfied with the performance of your initial model, consider trying different algorithms or more advanced models. Depending on the complexity of your problem, models like Random Forest, Gradient Boosting, or even deep learning techniques might be more suitable.

5. Model Interpretability:

- For some business use cases, it's essential to understand why your model makes certain predictions. Explore techniques for model interpretability to gain insights into the factors influencing your demand predictions.

6. Deployment:

- Once you're confident in your model's performance, deploy it to a production environment. This can involve creating APIs, integrating it into your business processes, and ensuring that it can make real-time predictions.

7. Monitoring and Maintenance:

- Continuously monitor the model in production. Track its performance and retrain or update it as needed to account for shifts in product demand patterns, changes in the market, or other relevant factors.

Remember that the specific steps and the level of complexity involved in product demand prediction can vary based on the nature of your business, the data available, and the goals of your forecasting task. The key is to approach it as an iterative process, constantly improving your model's accuracy and adaptability to changing market conditions.

2.7 WHAT METRICS USED FOR THE ACCURACY CHECK

For product demand prediction with machine learning, you can use several evaluation metrics to assess the performance of your predictive model. The choice of the metric depends on the specific characteristics of your problem and your business goals. Here are some commonly used metrics:

1. Mean Absolute Error (MAE):

- MAE measures the average absolute difference between the actual demand and the predicted demand. It provides a straightforward understanding of the model's prediction error.

2. Root Mean Square Error (RMSE):

- RMSE is similar to MAE but gives more weight to large errors. It is the square root of the average of the squared differences between the actual and predicted demand. RMSE penalizes larger errors more heavily.

3. Mean Absolute Percentage Error (MAPE):

- MAPE calculates the percentage difference between the actual and predicted demand. It's useful for understanding prediction accuracy relative to the actual demand values. However, it can be sensitive to small actual values.

4. R-squared (R2) Score:

- R-squared measures the proportion of the variance in the actual demand that is explained by the model. A higher R2 score indicates a better fit of the model to the data.

5. Root Mean Squared Logarithmic Error (RMSLE):

- RMSLE is particularly useful when dealing with data that has a wide range in the target variable. It calculates the logarithmic difference between actual and predicted values, which can reduce the impact of outliers.

6. Forecast Bias:

- Forecast bias measures the systematic error in your predictions. A positive bias indicates over-prediction, while a negative bias indicates under-prediction.

7. Percentage of Accurate Predictions:

- This metric measures the percentage of predictions that are within a certain percentage of the actual demand. For instance, you might assess the percentage of predictions within 10% of the actual values.

8. Custom Metrics:

- Depending on your specific business needs, you may define custom evaluation metrics. For example, you might create a metric that accounts for the cost of overstocking and understocking products.

The choice of metric depends on the characteristics of your data and the business context. Typically, MAE and RMSE are commonly used as they provide a clear measure of prediction error. However, you should consider the specific requirements of your problem and select the metric that aligns best with your business goals.