FUTURE SALES PREDICTION

IBM NAAN MUDHALVAN

PROJECT SUBMISSION PHASE-2

Future sales prediction is the process of predicting what the demand for certain products will be in the future. This helps manufracturers to decide what they should produce and guides retailers toward what they should stock.

Sales forecasting is the process of estimating future revenue by predicting how much of a product or service will sell in the next week, month, quarter, or year. At its simplest, a sales forecast is a projected measure of how a market will respond to a company's go-to-market efforts.

One of the most common methods used to predict sales is regression analysis. This method involves using historical sales data to train a model that can predict future sales.

The model can take into account factors such as past sales, marketing campaigns, and economic indicators to make its predictions.

Demand forecasting is aimed at improving the following process:

- ✓ Supplier relationship management
- ✓ Customer relationship management
- ✓ Order fulfillment and logistics
- ✓ Marketing campaigns
- ✓ Manufacturing flow management

DATA GATHERING:

Dataset is taken from kaggle competition and be downloaded from here:

https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction

DATA DESCRIPTION:

You are provided with daily historical sales data. The task is to forecast the total amount of products sold in every shop for the test set. Note that the list of shops and products slightly changes every month. Creating a robust model that can handle such situations is part of the challenge.

Machine Learning Algorithms:

Decision Trees and Random Forests: Useful for capturing complex relationships in the data.

Gradient Boosting Models (e.g., XGBoost, LightGBM): Excellent for predictive modeling and handling non-linear relationships.

Predicting future sales involves using data science techniques. A simple algorithmic approach could involve:

1. Data Collection:

Gather historical sales data, including timestamps.

Include relevant features like promotions, holidays, and other factors affecting sales.

2. Data Preprocessing:

Handle missing data and outliers.

Convert categorical variables into numerical representations.

Normalize or scale numerical features.

3. Feature Engineering:

Create new features, like moving averages or seasonality indicators.

Extract relevant information from timestamps, such as day of the week or month.

4. Model Selection:

Choose a regression model suitable for time-series data, like linear regression, decision trees, or more advanced models like ARIMA, SARIMA, or Prophet.

5. Training the Model:

Split data into training and validation sets.

Train the model on historical data.

6. Model Evaluation:

Evaluate the model's performance using metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) on the validation set.

7. Hyperparameter Tuning:

Fine-tune model parameters to improve performance.

8. Prediction:

Use the trained model to predict future sales based on new data.

9. Monitoring and Updating:

Regularly monitor the model's performance and update it as needed with new data.

ALGORITHMS:

1. List out the goods and services you sell.
2. Estimate how much of each you expect to sell.
3. Define the unit price or dollar value of each good or service sold.
4. Multiply the number sold by the price.
5. Determine how much it will cost to produce and sell each good or service

Importing Libraries:

EDA Libraries:

```
import pandas as pd
import numpy as np

import matplotlib.colors as col
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import datetime
from pathlib import Path
import random
```

Scikit-Learn models:

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,
mean_absolute_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from xgboost.sklearn import XGBRegressor
from sklearn.model_selection import KFold, cross_val_score,
train_test_split

from sklearn.model_selection import KFold, cross_val_score, train_test_split
```

LSTM:

```
import keras
from keras.layers import Dense
from keras.models import Sequential
from keras.callbacks import EarlyStopping
from keras.utils import np_utils
from keras.layers import LSTM
```

ARIMA Model:

```
import statsmodels.tsa.api as smt
import statsmodels.api as sm
from statsmodels.tools.eval_measures import rmse

import pickle
import warnings
```

2. Loading and Exploration of the Data
3. EDA (Exploratory Data Analysis)
4. Determining Time Series Stationary
5. Differencing
6. Scaling Data
7. Prediction Dataframe