**Title:**

Future Sales Prediction

**Abstract:**

Sales prediction plays a crucial role in business planning and decision -making processes . This abstract provides an overview of the future sales prediction , highlighting its significance and potential applications . By leveraging advanced technologies and data analysis techniques , business can gain valuable insights into future sales trends , enabling them to optimize inventory management , marketing strategies , and resource allocation . This abstract emphasizes the importance of accurate sales prediction models and the potential benefits they offer to businesses across various industries .

**Problem Statement:**

* The problem is to develop a predictive model that uses historical sales data to forecast future sales for a retail company. The objective is to create a tool that enables the company to optimize inventory management and make informed business decisions based on data driven sales predictions. This project involves data preprocessing, feature engineering, model selection, training, and evaluation .

**Phase Of Development:**

**1. Data collection**

- Identify the data sources

- Historical sales data (time series data)

- External factors (eg economic indicators , promotions)

-Acquire and gather the data from relevant sources .

-Ensure data quality and consistency

**2. Data Preprocessing:** Start by collecting historical sales data from the retail company. Clean the data by removing any duplicates, missing values, or outliers. Perform exploratory data analysis to gain insights into the data and identify patterns or trends.  
  
 **3. Feature Engineering:** Create relevant features that can help improve the predictive power of the model. This can include variables such as time of year, day of the week, promotional events, holidays, etc. These features can be derived from the existing data or external sources

**4. Model Selection:** Choose an appropriate machine learning algorithm for the sales prediction task. Some popular algorithms for time series forecasting include ARIMA, SARIMA, Prophet, and LSTM. Evaluate multiple models and select the one that performs the best based on evaluation metrics such as mean absolute error (MAE) or root mean squared error (RMSE).

**5. Model Training**: Split the data into training and testing sets. Use the training set to train the selected model on the historical sales data. Adjust hyperparameters if necessary to optimize the model's performance

**6. Model Evaluation:** Evaluate the trained model on the testing set to assess its predictive accuracy. Calculate evaluation metrics such as MAE, RMSE, or R-squared. Compare the model's performance with baseline models or benchmarks to validate its effectiveness.

**7. Forecasting:** Once the model is trained and evaluated, use it to make future sales prediction

**8.Deployment**

-Deploy the chosen model to a production environment.

-Implement APIs or scripts to automate the prediction process.

-Implement APIs or scripts to automate the prediction process

**9.Monitoring and Management**

-Continuously monitor the model's performance in production.

-Retrain the model periodically with new data.

-Update the model as needed to account for changing business conditions

**Dataset:**

The dataset used for the future sales prediction with machine learning project is a collection of historical sales . The dataset contains the following key features :

1. TV
2. Radio
3. Newspaper
4. Sales

**Data Preprocessing**

Import pandas as pd

Import numpy as np

From sklearn.preprocessing import StandardScaler

From sklearn.model\_selection import train\_test\_split

Data = pd.read\_csv(r”D:\Visual studio\Course\Naan Mudhalvan\Data Science\course\Sales.csv”)

X = data[[‘TV’,’Radio’,’Newspaper’]]

Y = data[‘Sales’]

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,train\_size=0.3,random\_state=42)

**Model Training Process:**

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.

**Evaluation metrices for regression:**

1. **Mean Squared Error (MSE):**

- MSE measures the average of the squared differences between the predicted and actual values.

- Lower MSE indicates better model performance.

- It penalizes large errors more than smaller ones.

- Formula: MSE = (1/n) \* Σ(yi - ŷi)^2, where n is the number of data points, yi is the actual value, and ŷi is the predicted value.

2. **Root Mean Squared Error (RMSE):**

- RMSE is the square root of the MSE and provides the error in the same units as the target variable.

- It offers a more interpretable measure of error.

- Formula: RMSE = √MSE.

3. **Mean Absolute Error (MAE):**

- MAE measures the average of the absolute differences between the predicted and actual values.

- It's less sensitive to outliers compared to MSE.

- Formula: MAE = (1/n) \* Σ|yi - ŷi|.

4**. R-squared (R^2):**

- R-squared represents the proportion of the variance in the dependent variable (demand) that is predictable from the independent variables (features).

- It ranges from 0 to 1, where a higher value indicates a better fit.

- Formula: R^2 = 1 - (SSR / SST), where SSR is the sum of squared residuals and SST is the total sum of squares.

**Code:**

Import pandas as pd

Import numpy as np

From sklearn.preprocessing import StandardScaler

From sklearn.model\_selection import train\_test\_split

From sklearn.ensemble import RandomForestRegressor

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

Data = pd.read\_csv(r”D:\Visual studio\Course\Naan Mudhalvan\Data Science\course\Sales.csv”)

X = data[[‘TV’,’Radio’,’Newspaper’]]

Y = data[‘Sales’]

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,train\_size=0.3,random\_state=42)

Model = RandomForestRegressor(n\_estimators=400,random\_state=42)

Model.fit(X\_train,y\_train)

Score = model.score(X\_test,y\_test)

Prediction = model.predict(X\_test)

Predicted\_score = r2\_score(y\_test,prediction)

Mse = mean\_squared\_error(y\_test,prediction)

Mae = mean\_absolute\_error(y\_test,prediction)

Rmse = mean\_squared\_error(y\_test,prediction,squared=False)

Print(f’Score : {score}’)

Print(f’R squared score : {predicted\_score}’)

Print(f’Mean squared error : {mse}’)

Print(f’Mean absolute error :{mae}’)

Print(f’Root mean squared error :{rmse}’)

**With user input:**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

data = pd.read\_csv(r"D:\Visual studio\Course\Naan Mudhalvan\Data Science\course\Sales.csv")

X = data[['TV','Radio','Newspaper']].values

y = data['Sales']

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state=42)

model = RandomForestRegressor(n\_estimators=400,random\_state=42)

model.fit(X\_train, y\_train)

user\_input =[8.6,2.1,1]

prediction = model.predict(X\_test)

score = model.score(X\_test,y\_test)

random\_forest\_regression = model.predict(np.array(user\_input).reshape(1,-1))

print(random\_forest\_regression[0])

print('Score :',score\*100)

**Output:**

Score : 0.9093587380521827

R squared score : 0.9093587380521827

Mean squared error : 2.5192362491071543

Mean absolute error :1.2580928571428582

Root mean squared error :1.5872102094893272

**Conclusion:**

These techniques enhances the data optimization along with improving the efficiency with better results and greater predictability. After predicting the purchase amount, the companies can apply some marketing strategies for certain sections of customers so that the profit could be enhanced.