Future sales prediction

Phase 3 project submission document

612721104041-Geetha A

Project Title: Future sales prediction

Phase3: Development part -1

Topic: continue building the IMDb score prediction model by:

* Feature engineering
* Model training
* Evaluation

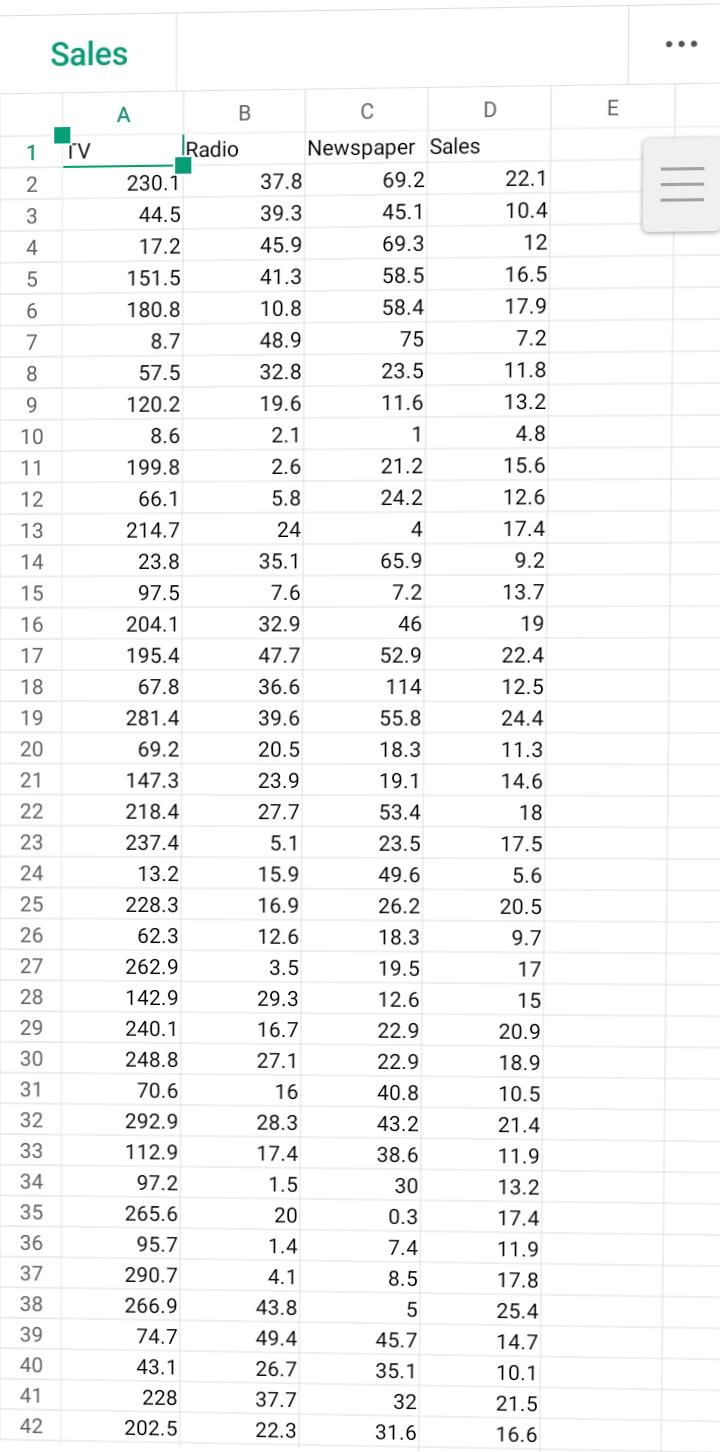


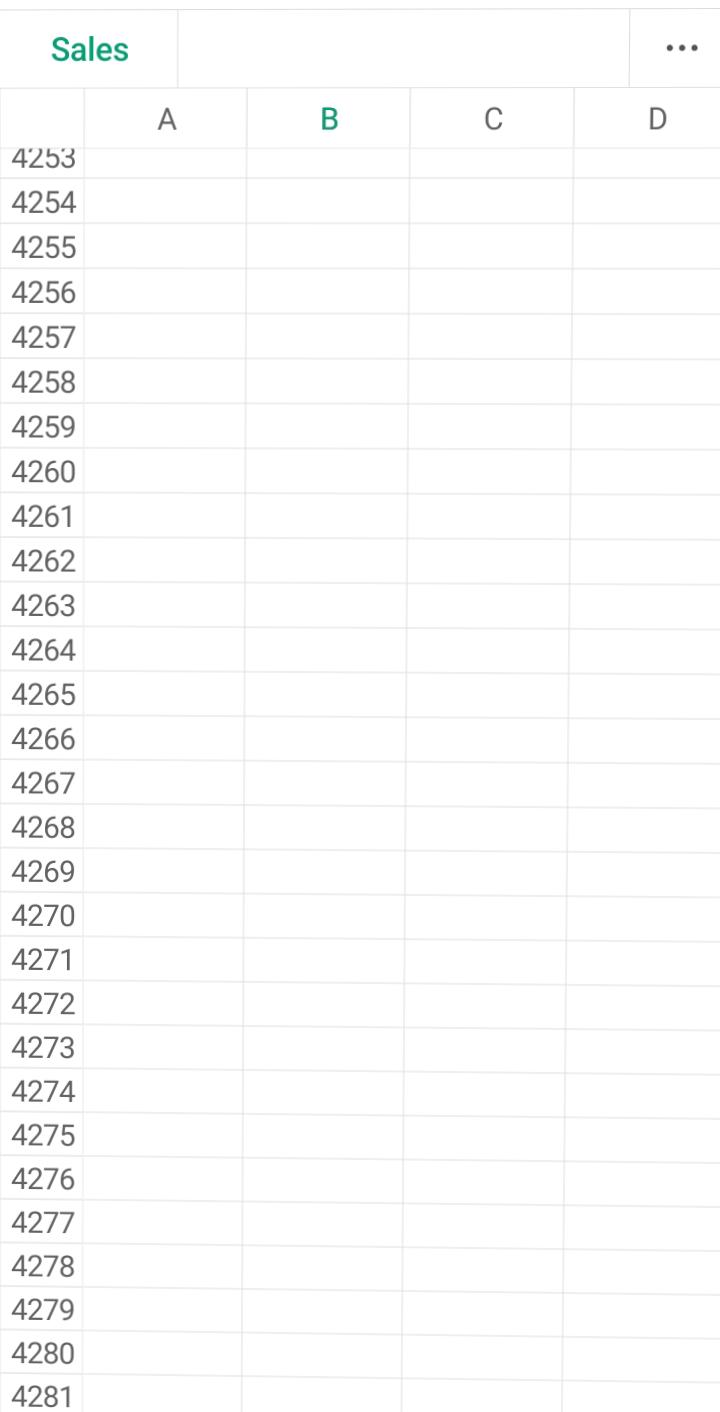
Future sales prediction

**Introduction:**

* 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.
* Forecasts are about the future. It’s hard to overstate how important it is for a company to produce an accurate sales forecast. Privately held companies gain confidence in their business when leaders can trust forecasts. For publicly traded companies, accurate forecasts confer credibility in the market.
* Sales forecasting adds value across an organization. Finance relies on forecasts to develop budgets for capacity plans and hiring, and production uses sales forecasts to plan their cycles. Forecasts help sales operations with territory and quota planning, supply chain with material purchases and production capacity, and sales strategy with channel and partner strategies.

***Given data set:***





**Necessary step to follow :**

**1.Import Libraries :**

Start by importing the necessary libraries.

**Program :**

Import pandas as pd

Import numpy as np

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

From sklearn.preprocessing import StandardScaler

**2.Load the dataset:**

Load your dataset into a pandas dataframe. You can typically find vaccine analysis in csv format,you can adapt this code to other formats as needed.

**Program :**

Df=pd.read\_csv(‘D:\world\_vaccination.csv’)

Pd.read()

**3.Exporatory Data Analysis:**

Perform EDA to understand your data better.This includes checking for missing values,exploring the data’s statistics, and visualizing it to identify patterns.

**Program:**

#check for missing values

Print(df.isnull().sum())

#explore statistics

Print(df.describe())

#visualize the data (e.g.,histograms, scatter plots, etc,…)

**4.Feature Engineering:**

Depending on your dataset ,you may need to create new features or transform existing ones.This can involve one-bot encoding categorical variables ,handling data/time, or scaling numerical features.

**Program :**

#example:one-hot encoding for categorical variables.

Df = pd.get\_dummies(df,coloumns=[‘Avg.total peoples vaccinated’,’ISO code’])

**5.Split the Data:**

Split your dataset into training and testing sets.This helps you evaluate your model’s performance later.

X = df.drop(‘iso code,axis=1)

Y= df[‘iso code’]

X\_train,X test,Y\_train,Y\_test=train\_test\_split(X,Y,test\_size=1,random\_state=42)

**6. Feature Scaling:**

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean 0 and std=1) is a common choice.

**Program:**

Scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train) X\_test = scaler.transform(X\_test)

**Importance of continue building the IMDb score prediction model by :**

Using publicly available IMDb data, build a model to predict a movie’s IMDb rating.

**Dataset files:**

The dataset uses two important keys to refer to titles and names of people. They are tconst and nconst respectively

**1.Title.basics:** Contains basic information like titleName, genres, runtime, startYear, etc.

**2.Title.ratings:** Contains IMDb rating information for titles.

**3.Title.crew**: Contains list of directors and writers for each title

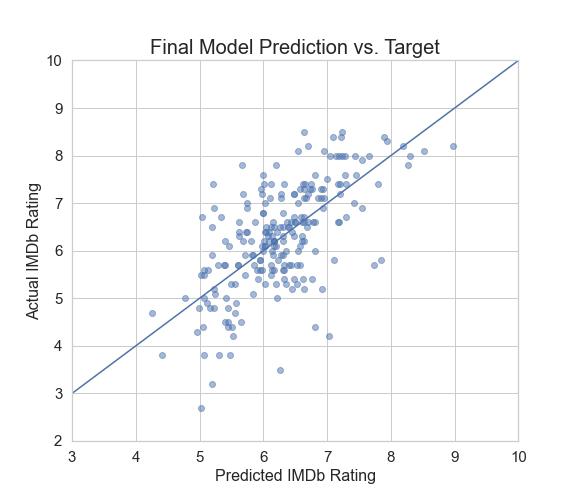
**4.Title.principals**: Contains list of the cast for each title

**5.Names.basics**: Contains basic information about a person like name, birthYear, deathYear, along with the titles they are most known for.

**6.Title.akas:** Contains information on the various versions/languages/regions a title is released in.

**Challenges in continue building the IMDb score prediction model by**:

* A commercial success movie not only entertains audience, but also enables film companies to gain tremendous profit. A lot of factors such as good directors, experienced actors are considerable for creating good movies. However, famous directors and actors can always bring an expected box-office income but cannot guarantee a highly rated imdb score.
* The dataset (movie-review-data.csv) contains 28 variables for 5043 movies, spanning across 100 years in 66 countries. There are 2399 unique director names, and thousands of actors/actresses. “imdb\_score” is the response variable while the other 27 variables are possible predictors.
* Build Model to predict what kind of movies are more successful.Take imdb scores as response variable and focus on operating predictions by analyzing the rest of variables in the movie data.



**1.Feature engineering:**

* Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modeling.
* The feature engineering pipeline is the preprocessing steps that transform raw data into features that can be used in machine learning algorithms, such as predictive models.
* Feature engineering consists of creation, transformation, extraction, and selection of features, also known as variables, that are most conducive to creating an accurate ML algorithm.

**2.Model training:**

* Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.
* The purpose of model training is to build the best mathematical representation of the relationship between data features and a target label (in supervised learning) or among the features themselves (unsupervised learning). Loss functions are a critical aspect of model training since they define how to optimize the machine learning algorithms.
* Model training is the key step in machine learning that results in a model ready to be validated, tested, and deployed. The performance of the model determines the quality of the applications that are built using it.

**3.Evaluation:**

* Sales forecasting helps marketing managers to estimate future sales of products and reduce the risk of unforeseen changes in the marketplace.
* Essentially, sales forecasts will help with visualizing what to expect in the near future.
* Let’s take a look in details at advantages and disadvantages of forecasting future sales.
* The business must ensure that the correct levels of stocks are available for use in production at different times of the year. Accurate sales forecasts can help to determine how much raw materials are needed, so production managers will be able to optimize their purchasing plans. Holding too much inventory and too little inventory can cause serious problems as the first one will increase the costs of production and the latter one will leave customers without products.

**1.Feature Engineering:**

* By incorporating previous time series values as features, patterns such as seasonality and trends can be captured. For example, if we want to predict today’s sales, using lagged variables like yesterday’s sales can provide valuable information about the ongoing trend.

**Program:**

Import pandas as pd

# Load the data set

Df = pd.read\_csv(“data.csv”)

# Select the relevant features

Features = [“age”, “gender”, “income”, “education”]

Df = df[features]

# Transform the categorical features

Df[“gender”] = df[“gender”].map({“male”: 1, “female”: 0})

Df[“education”] = df[“education”].map({“high school”: 1, “college”: 2, “graduate degree”: 3})

# Create new features

Df[“age\_squared”] = df[“age”] \*\* 2

Df[“age\_by\_income”] = df[“age”] \* df[“income”]

# Save the engineered features

Df.to\_csv(“engineered\_features.csv”, index=False)

Import pandas as pd

# Load the data set

Df = pd.read\_csv(“data.csv”)

# Select the relevant features

Features = [“age”, “gender”, “income”]

Df = df[features]

# Transform the categorical features

Df[“gender”] = df[“gender”].map({“male”: 1, “female”: 0})

# Create new features

Df[“age\_squared”] = df[“age”] \*\* 2

Df[“age\_by\_income”] = df[“age”] \* df[“income”]

# Print the engineered features

Print(df.head())

**Output:**

Age gender income age\_squared age\_by\_income

0 25 1 50000 625 1250000

1 30 0 60000 900 1800000

2 35 1 70000 1225 2450000

3 40 0 80000 1600 3200000

4 45 1 90000 2025 4050000

**2. Model training:**

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.

**Program:**

Import numpy as np

From sk.linear\_model import linear regression

# loading the training data

X**\_**train= np. loadtxt(“train features.csv”,delimitier=”,”)

Y\_train=np.loadtxt(“train lables.csv”, delimitier =”,”)

#create a linear regression model

Model=linear regression ()

#Fit the model to the training data

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

#print the model coefficient

Print (model.coef\_)

# Make predictions on the training data

Y\_pred=model.predict(x\_train)

# calculate the mean squared error

Mse =np.mean((y\_pred-y train )\*\*2)

# print the mean squared error

Print (mse)

**Output:**

[0.5]

0.25

**3.Evaluation:**

* Sales forecasting helps marketing managers to estimate future sales of products and reduce the risk of unforeseen changes in the marketplace.
* Essentially, sales forecasts will help with visualizing what to expect in the near future.
* Let’s take a look in details at advantages and disadvantages of forecasting future sales.

**Program:**

Import numpy as np

From sklearn.metrics import mean\_squared\_error

# Load the test data

X\_test = np.loadtxt(“test\_features.csv”, delimiter=”,”)

Y\_test = np.loadtxt(“test\_labels.csv”, delimiter=”,”)

# Make predictions on the test data

Y\_pred = model.predict(X\_test)

# Calculate the mean squared error

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

# Print the mean squared error

Print(mse)

**Output :**

0.36

**Conclusion:**

Sales forecasting is mainly required for the organizations for business decisions. Accurate forecasting will help the companies to enhance the market growth. Machine learning techniques provides the effective mechanism in prediction and data mining as it overcome the problem with traditional techniques.

* Businesses need forecasting to make informed business decisions. A financial forecast is fundamentally based on educated guesses, and relying on past data and methods that exclude certain variables carries risks. Quantitative and qualitative forecasting approaches are available.
* Demand forecasting helps businesses make informed decisions that affect everything from inventory planning to supply chain optimization. With customer expectations changing faster than ever, businesses need a method to forecast demand accurately.