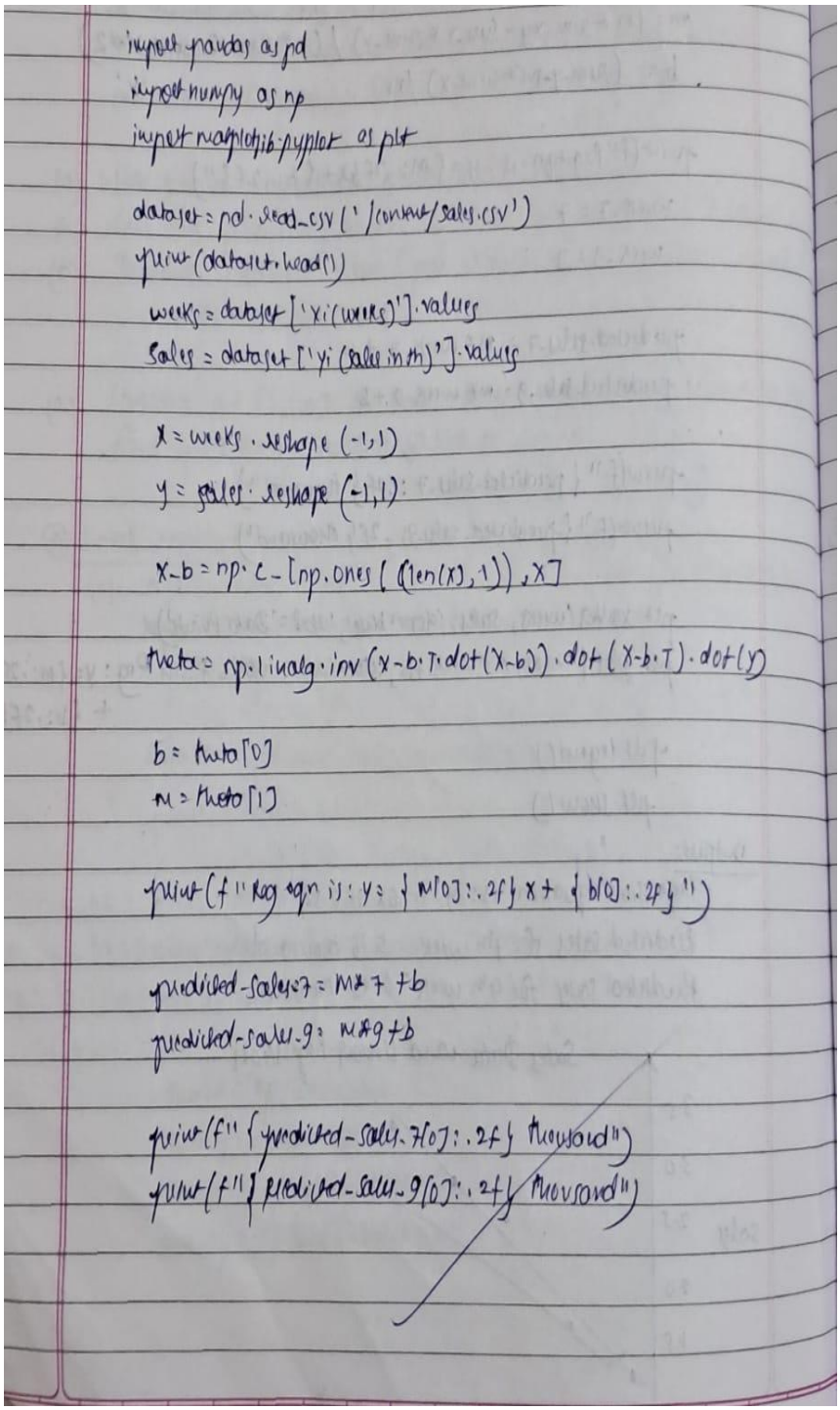
Program 7

Demonstrate various data pre-processing techniques for a given dataset

Screenshot:



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dataset = pd.read_csv('/content/sales.csv')
print(dataset.head())
weeks = dataset['x:(weeks)'].values
Sales = dataset['y: (Sales in th)'].values

X = weeks.reshape(-1,1)
y = Sales.reshape(-1,1)

X_b = np.c_[np.ones((len(X),1)),X]

theta = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)

b = theta[0]
m = theta[1]

print(f"Reg eqn is: y= {m[0]:.2f} x + {b[0]:.2f}")

predicted_sales_7 = m*7 +b
predicted_sales_9 = m*9+b

print(f"{predicted_sales_7[0]:.2f} thousand")
print(f"{predicted_sales_9[0]:.2f} thousand")
```

Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load the dataset from CSV
df = pd.read_csv("/content/iris (1).csv")

# Features and target
X = df[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
y = df['species']

# Encode species labels (setosa → 0, versicolor → 1, virginica → 2)
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)


# Train the SVM model

svm_model = SVC(kernel='rbf', C=1.0, gamma='scale')

svm_model.fit(X_train_scaled, y_train)


# Predictions

y_pred = svm_model.predict(X_test_scaled)


# Evaluation

print("Accuracy:", accuracy_score(y_test, y_pred))

print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

print("Classification Report:\n", classification_report(y_test, y_pred,
target_names=label_encoder.classes_))
```