# **1.sales prediction project**

1.**Data collection**:Download the Bigmart sales

Dataset(2013) from the kaggle website.

**2.Data preprocessing:**

-convert categorial variables into numerical variables(e.g.,one-hot encoding

-handle missing values(if any)

-Normalize/scale numerical variables

**3.Exploratory Data Analysis(EDA):**

-Analyze sales trends,product categories,and store locations

-visualize data using plots(e.g.,bar charts,scatter plots)

**4.Feature Engine :**

-Extract relevant features from product and store attributes

-create new features (e.g.,average sales per month)

**5.Model selection and training:**

-choose regression models(e.g.,Linear Regression,random Forest,Gradient Boosting)

-Train models using the preprocessed data

**6.Model Evaluation:**

-Evaluate model performance using metrices(e.g.,RMSE,MAE,R-squared)

-compare model performance

**7.Hyperparameter** grid search,cross-validation) to optimize model hyperparameters

**8.Model Deployment:**

-use the trained model to predict sales for the next year

-visualize predictions using plots(e.g., bar charts,line plots)

## code

import numpy as np

import pandas as pd

from sklearn.model\_selection import

train\_test\_split

from sklearn.linear\_model import

LinearRegression

From sklearn.metrics import

Mean\_squared\_error

Import matplotlib.pyplot as plt

#Load dataset(replace ‘your\_dataset.csv’ with your actual dataset)

Data-pd.read\_csv(‘your\_dataset.csv’)

#assume the dataset has columns ‘weather’,’soil\_condition’,and ‘yield’

Features=data[[‘weather’,’soil\_condition’]]

Target=data[‘yield’]

#split data into training and test sets

X-train,X\_test,y\_train,y\_test=train\_test\_split(features,target,test\_size=0.2,random\_states=42)

#Initialize and train the model

Modle=LinearRegression()

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

#make predictions

Predictions=model.predict(X-test)

#Evaluate the model

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

Print(“Mean Squared Error:{mse}”)

#plot predictions vs actual values

Plt.scatter(y\_test,predictions)

Plt.xlabel(‘Actual Values’)

Plt.ylabel(‘Predicted Values’)

Plt.title(‘Actual vs Predicted Values’)

Plt.show()

**2.Iris Flower classification project**

1**.Data collection**:

-Download the iris dataset from the UCI ML Repository

**2.Data preprocessing:**

-Handle missing values

-Analyze sales trends,product categories,and store locations

**3.Exploratory Data Analysis(EDA):**

**-Analyze Iris flower characteristics (e.g., petal and sepal measurements)**

**-visualize data using plots(e.g., scatter plots,histograms)**

**4.Feature Selection:**

-choose relevant features (e.g., petal and sepal length/width)

**5.Model selection and training:**

-choose classification models(e.g.,Logistic Regression,Decision Trees,SVM)

-Train models using the preprocessed data

**6.Model Evaluation:**

-Evaluate model performance using metrices

-compare model performance

7.Hyperparameter Turning:

-use techniques (e.g.,grid serach, cross-vadilation)to optimize model hyperparameters

**8.Model Deployment:**

-use the trained model to classify Iris flowers into species

-visualize predictions using plots(e.g., confusion matrix,classification report)

## Code

Import pandas as pd

From sklearn.model\_selection import

train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import accurancy\_store,

classification\_report

#Load the Iris dataset

Data=pd.read\_csv(‘iris.csv’)

#preprocess the data

Data.dropna(inplace=True)

#Handle missing values

#split the data into training and testing sets

X=data.drop([‘Species’],axis=1)

Y=data[‘Species’]

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

#Train an SVM model

Model=SVC()

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

#Make predictions and evaluate the model

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

Accuracy=accuracy\_score(y\_test,y\_pred)

Print(f’Accuracy:{accuracy}’)

Print(classification\_report(y\_test,y\_pred))