ProductSalesAnalysisUsingPython

TEAM MEMBER

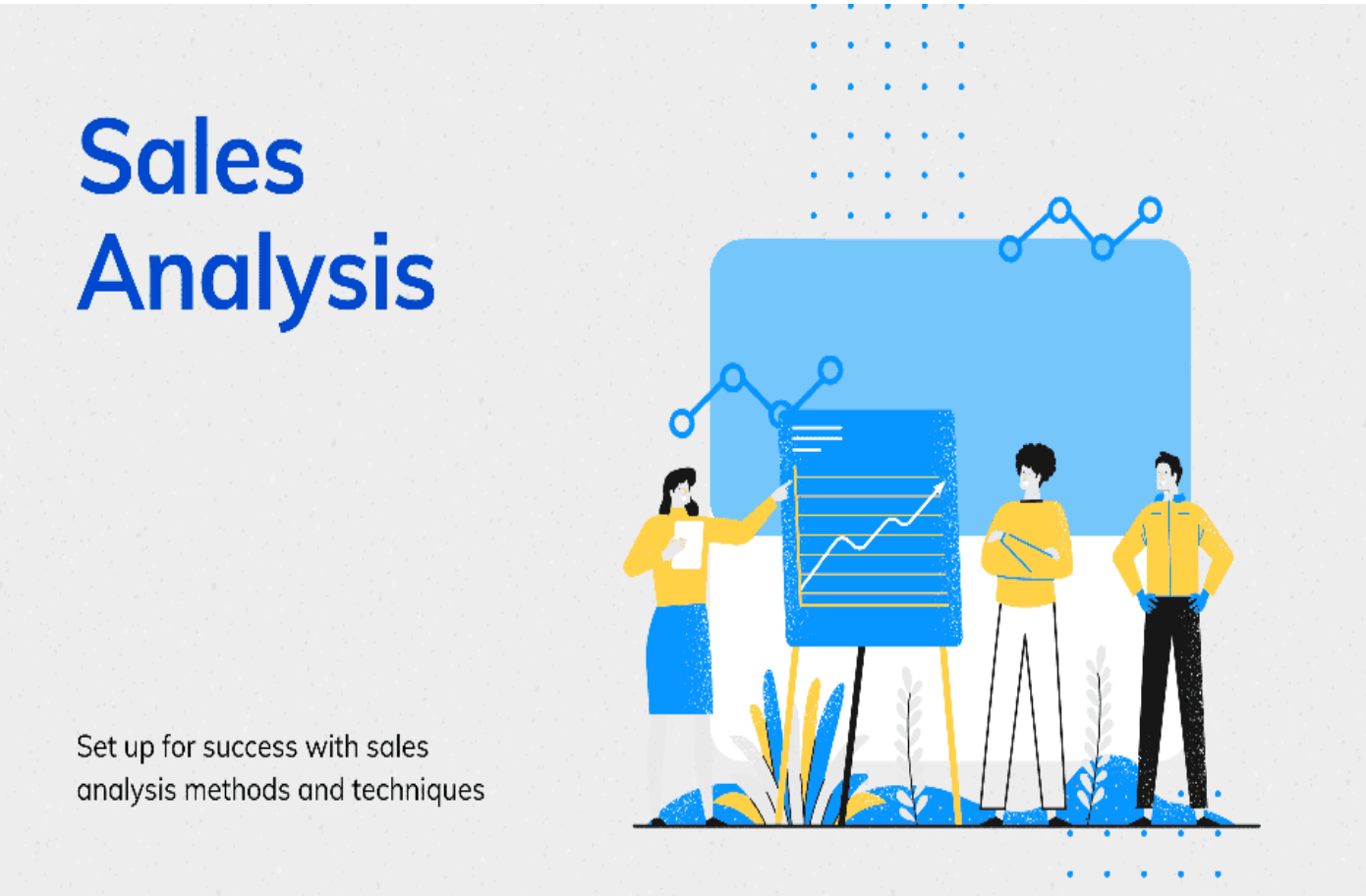
621521104098: praveenkumar.A

Phase-2submissiondocument

ProjectTitle:Product SalesAnalysisUsing PythonPhase 2:

Innovation

Topic:Inthissectionyouneedtoputyourdesignintoinnovationtosolvetheproblem



ProductSalesAnalysisUsingPython

Datapreprocessingisacrucialstepin productsalesanalysis.

DataCollection:Gatherdatafromvarioussources,suchassalesrecords,databases,or onlineplatforms.

DataCleaning:

- Handlemissingvaluesbyimputingor removingthem.
- Removeduplicates.
- Correcterrors or inconsistenciesinthedata.

DataTransformation:

- Convertdata types(e.g., dates,categoricalvariables).
- Normalizeor scalenumericalfeatures.
- Createnewfeaturesif necessary(e.g., calculatetotal sales perproduct).

DataIntegration:

- Mergedatafrom differentsourcesifneeded(e.g.,combining salesdatawithproductinformation).

FeatureEngineering:

Createrelevantfeatureslikecalculatingsalesperday,week,ormonth.Usedomainknowledgetoengineerfeaturesthatmayimpactsales.

Datasetlink:https://www.kaggle.com/datasets/ksabishek/product-sales-data

MachinelearningModeling:

DecisionTree:

Adecisiontreeisasupervisedmachinelearningalgorithmthatcanbeusedforbothclassificatio nand regression tasks. It creates a tree-like structure where each internal node represents a feature,eachbranchrepresentsadecisionrule,andeachleafnoderepresentsanoutcomeorclas s.

RandomForest:

Arandomforestisanensemblelearningmethodthatbuildsmultipledecisiontreesduringtrainin gandcombinestheirpredictionstoimproveaccuracyandreduceoverfitting.Itoperatesbycreatin gaforest of decision trees, where each tree is trained on a different subset of the data and uses arandomselectionoffeatures.

Algorithm:

Sure!Decisionoftreesareapopularalgorithmforclassificationandregressiontasks.Random forestsareanensemblemethodthat combinesmultipledecisiontreesforbetter performance.

Coding:

```
#Importthenecessarylibraries

Fromsklearn.treeimport

DecisionTreeRegressor

Fromsklearn.ensembleimport

RandomForestRegressor

#Createadecisiontreeregressor decision_tree=DecisionTreeRegressor()

#Createarandomforestregressor random_forest=RandomForestRegressor()

#Trainthemodelswithyourdata
```

Decision_tree.fit(X_train,y_train)

Random_forest.fit(X_train,y_train)

#Makepredictions

Decision_tree_predictions=

Decision_tree.predict(X_test)

Random_forest_predictions =

Random_forest.predict(X_test)

Remembertoreplace`X_train,‘y_train,andX_testwithyouractualtrainingandtestingdata.Letme knowifyouneedfurtherassistance!

Importnecessarylibraries:

importpandasaspd

importmatplotlib.pyplotaspltimportnumpyasnp

LoadthedataintoaPandasDataFrame:

Load the sales data into a

DataFramesales_data=pd.read_csv('sale

s_data.csv')

#DisplaythefirstfewrowsoftheDataFrametounderstandthestructureofthedataprint(sales_data.head())

Calculatekeymetrics:

Totalrevenue

total_revenue=sales_data['Revenue'].sum

()#Totalquantitysold

total_quantity_sold=sales_data['Quantity'].sum()

#Averagesellingpriceaverage_selling_price=total_revenue/total_quantity_sold

#Displaythecalculatedmetrics

print('TotalRevenue:',total_revenue)print('TotalQuantitySold:',total_quantity_sold)print('Avera

geSellingPrice:',average_selling_price)

Analyzeproductperformance:

#Groupdatabyproductandcalculatetotalrevenueandtotalquantitysoldforeachproductproduc
t_performance = sales_data.groupby('Product').agg({'Revenue': 'sum',
'Quantity':'sum'}).reset_index()

#Sortproductsbyrevenueindescendingorderproduct_performance=pr
oduct_performance.sort_values(by='Revenue',ascending=False)

#Displaythetop-
performingproductsprint('TopPerformingProducts:')print(product_perf
ormance.head())

Visualize product performance# Plot top performing products plt.figure(figsize=(12,
6))plt.bar(product_performance['Product'],product_performance['Revenue'])plt.xlabel('Product')

plt.ylabel('TotalRevenue')

plt.title('TopPerformingProductsbyRevenue')plt.xticks(rotation=45)plt.t

ight_layout()

plt.show()