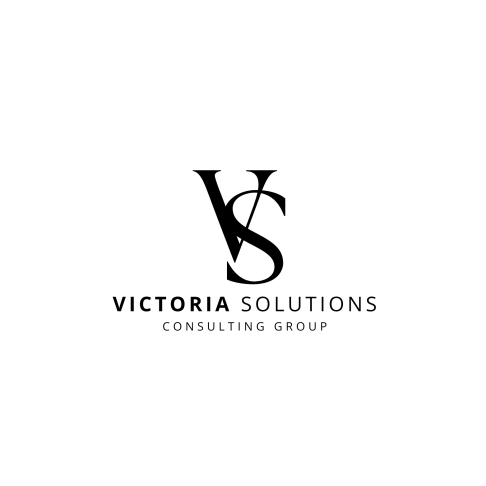
|  |  |
| --- | --- |
| Name | Asad |
| Contact Number | 07443892214 |
| Project Title (Example – Week1, Week2, Week3) | Week 4 project |



**Project Guidelines and Rules**

1. **Formatting and Submission**
   * **Format:** Use a readable font (e.g., Arial/Times New Roman), size 12, 1.5 line spacing.
   * **Title:** Include Week and Title (Example - Week 1: TravelEase Case Study.)
   * **File Format:** Submit as PDF or Word file to contact@victoriasolutions.co.uk
   * **Page Limit:** 4–5 pages, including the title and references.
2. **Answer Requirements**
   * **Word Count:** Each answer should be 100–150 words; total 800–1,200 words.
   * **Clarity:** Write concise, structured answers with key points.
   * **Tone:** Use formal, professional language.
3. **Content Rules**
   * Answer all questions thoroughly, referencing case study concepts.
   * Use examples where possible (e.g., risk assessment techniques).
   * Break complex answers into bullet points or lists.
4. **Plagiarism Policy**
   * Submit original work; no copy-pasting.
   * Cite external material in a consistent format (e.g., APA, MLA).
5. **Evaluation Criteria**
   * **Understanding:** Clear grasp of business analysis principles.
   * **Application:** Effective use of concepts like cost-benefit analysis and Agile/Waterfall.
   * **Clarity:** Logical, well-structured responses.
   * **Creativity:** Innovative problem-solving and examples.
   * **Completeness:** Answer all questions within the word limit.
6. **Deadlines and Late Submissions**
   * **Deadline:** Submit on time; trainees who submit fail to submit the project will miss the “Certificate of Excellence”

1. **Additional Resources**
   * Refer to lecture notes and recommended readings.
   * Contact the instructor or peers for clarifications before the deadline.

**START YOUR PROJECT FROM HERE:**

**Week 4 project**

**1. Introduction**

The objective of this project is to apply AI-driven analytics to customer and sales data in order to generate actionable business insights. The dataset includes demographic, financial, and behavioural variables such as Age, Income, Credit Score, Loan Amount, Marketing Spend, and Purchase Frequency, with target variables Sales, Customer Churn, and Loan Default. Data cleaning and preprocessing were performed to ensure quality. Power BI was used for descriptive and visual analytics, highlighting key trends and patterns, while Spyder/Python was employed to build predictive models and prescriptive analytics, enabling evidence-based decision-making across sales growth, churn reduction, and risk management.

**2. Data Exploration (with Visuals)**

The dataset contains both categorical variables (e.g., Gender, Seasonality) and numerical variables (e.g., Age, Income, Sales). Descriptive statistics show variation across spending patterns, credit scores, and default history. Visual exploration in Power BI reveals important trends: Sales vs. Marketing Spend highlights spending efficiency, Churn by Demographics shows customer segmentation, and Loan Default by Income uncovers risk distribution. A time series forecast of Sales offers predictive insights into future trends. Among these, the time series graph is the most valuable, as it not only captures historical sales behaviour but also projects future performance, supporting both strategic and operational decisions.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**3. Predictive & Prescriptive Analytics**

**Sales Prediction (Regression)**

* **Best Model**: Linear Regression
* **Performance**: Good fit with reasonable error scores (MSE, MAE, R²).
* **Key Drivers**: Seasonality (high, medium, low), previous defaults, gender, purchase frequency, age, spending score, credit score.
* **Insight**: Seasonal marketing campaigns and loyalty programs should be prioritized to boost sales, while managing credit risks among default-prone customers.

**Customer Churn (Classification)**

* **Best Model**: Random Forest
* **Key Drivers**: Sales, credit score, marketing spend, income, loan amount, age.
* **Insight**: Focus retention efforts on high sales customers with lower credit scores. Adjust marketing spend strategically to engage at-risk groups.

**Loan Default (Classification)**

* **Best Model**: Random Forest
* **Key Drivers**: Marketing spend, income, sales, age, loan amount, credit score.
* **Insight**: Strengthen credit screening for low-income/high-loan customers and adjust marketing campaigns to attract financially stable profiles.

**Risk Buckets (Segmentation)**

* **Approach**: Customers grouped into deciles based on probability of default.
* **Findings**: Highest-risk bucket had ~38% default probability, while lowest was ~9%.
* **Insight**: Intervene early with customers in high-risk deciles through tailored repayment plans, while cross-selling to low-risk customers.

from pathlib import Path

import numpy as np

import pandas as pd

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

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.metrics import (

accuracy\_score, precision\_score, recall\_score, f1\_score,

roc\_auc\_score, brier\_score\_loss,

mean\_squared\_error, mean\_absolute\_error, r2\_score

)

from sklearn.linear\_model import LogisticRegression, LinearRegression

from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor

# ====== CONFIG ======

DATA = Path(r"C:\Users\sssss\OneDrive\Documents\internship projects bootcamp\cleaned\_dataset\_week4\_v2.csv")

NUMERIC = ["Age","Income","Spending\_Score","Credit\_Score","Loan\_Amount",

"Previous\_Defaults","Marketing\_Spend","Purchase\_Frequency","Sales"]

CATEG = ["Gender","Seasonality"]

SEED, TEST\_SIZE = 42, 0.25

# =====================

def header(txt):

print("\n" + "="\*len(txt)); print(txt); print("="\*len(txt))

def get\_preprocessor(df, feats):

num = [c for c in NUMERIC if c in feats and c in df.columns]

cat = [c for c in CATEG if c in feats and c in df.columns]

prep = ColumnTransformer([

("cat", OneHotEncoder(handle\_unknown="ignore", sparse\_output=False), cat),

("num", "passthrough", num),

])

return prep, num, cat

def feature\_names(prep, num\_cols, cat\_cols):

try:

# Modern scikit-learn: includes both OHE + passthrough numerics

return np.array(prep.get\_feature\_names\_out())

except Exception:

# Fallback generic names

return np.array([f"f{i}" for i in range(len(num\_cols) + len(cat\_cols))])

def top\_k\_importances(model\_pipe, k=10):

pre = model\_pipe.named\_steps["prep"]

clf = model\_pipe.named\_steps["clf"]

names = feature\_names(pre, [], [])

if hasattr(clf, "coef\_"):

imp = np.abs(clf.coef\_.ravel())

elif hasattr(clf, "feature\_importances\_"):

imp = clf.feature\_importances\_

else:

imp = np.zeros(len(names))

order = np.argsort(imp)[::-1][:k]

return pd.DataFrame({"feature": names[order], "importance": imp[order]})

def print\_prescriptive(fi\_df, label):

print("\nPrescriptive notes:")

feats = " | ".join(fi\_df["feature"].head(5).tolist())

print(f"- Top drivers for {label}: {feats}")

print("- Act on high-impact drivers: scale helpful levers (e.g., Marketing\_Spend),")

print(" and mitigate risky ones (e.g., low Credit\_Score, high Previous\_Defaults).")

print("- Use risk buckets below to prioritize outreach to highest-risk deciles.")

# ---------- LOAD ----------

df = pd.read\_csv(DATA)

header("Loaded dataset")

print(df.head(3)); print(f"\nShape: {df.shape}")

# ============================================================

# 1) SALES REGRESSION (LinearReg + RandomForestRegressor)

# ============================================================

if "Sales" in df.columns:

header("Analytic 1 — Sales Regression (console only)")

target = "Sales"

feats = [c for c in NUMERIC + CATEG if c != target and c in df.columns]

X = df[feats].copy()

y = df[target].values

prep, num\_cols, cat\_cols = get\_preprocessor(df, feats)

models = {

"LinearRegression": Pipeline([("prep",prep),("clf",LinearRegression())]),

"RandomForestReg": Pipeline([("prep",prep),("clf",RandomForestRegressor(n\_estimators=500, random\_state=SEED))]),

}

Xtr, Xte, ytr, yte = train\_test\_split(X, y, test\_size=TEST\_SIZE, random\_state=SEED)

rows, best = [], {"mse": 1e18, "name": None, "pipe": None}

for name, pipe in models.items():

pipe.fit(Xtr, ytr)

pred = pipe.predict(Xte)

mse = mean\_squared\_error(yte, pred)

rmse = mse\*\*0.5

mae = mean\_absolute\_error(yte, pred)

r2 = r2\_score(yte, pred)

rows.append([name, mse, rmse, mae, r2])

if mse < best["mse"]:

best.update({"mse": mse, "name": name, "pipe": pipe})

print(pd.DataFrame(rows, columns=["Model","MSE","RMSE","MAE","R2"]).round(4))

fi = top\_k\_importances(best["pipe"], k=10)

print("\nTop drivers of Sales (best model =", best["name"], ")")

print(fi.to\_string(index=False))

print\_prescriptive(fi, "Sales")

# =================================================================

# 2) CUSTOMER CHURN CLASSIFICATION (LogReg + RandomForest)

# =================================================================

best\_binary = None # keep best binary model for deciles

if "Customer\_Churn" in df.columns:

header("Analytic 2 — Customer Churn (console only)")

target = "Customer\_Churn"

feats = [c for c in NUMERIC + CATEG if c != target and c in df.columns]

X = df[feats].copy()

y = df[target].astype(int).values

prep, num\_cols, cat\_cols = get\_preprocessor(df, feats)

models = {

"LogReg": Pipeline([("prep",prep),("clf",LogisticRegression(max\_iter=2000))]),

"RandomForest": Pipeline([("prep",prep),("clf",RandomForestClassifier(n\_estimators=400, random\_state=SEED))]),

}

Xtr, Xte, ytr, yte = train\_test\_split(X, y, test\_size=TEST\_SIZE, random\_state=SEED, stratify=y)

rows, best = [], {"auc": -1, "name": None, "pipe": None, "proba": None}

for name, pipe in models.items():

pipe.fit(Xtr, ytr)

proba = pipe.predict\_proba(Xte)[:,1]

pred = (proba >= 0.5).astype(int)

acc = accuracy\_score(yte, pred)

prec = precision\_score(yte, pred, zero\_division=0)

rec = recall\_score(yte, pred, zero\_division=0)

f1 = f1\_score(yte, pred, zero\_division=0)

auc = roc\_auc\_score(yte, proba)

brier = brier\_score\_loss(yte, proba) # probability MSE

rows.append([name, acc, prec, rec, f1, auc, brier])

if auc > best["auc"]:

best.update({"auc": auc, "name": name, "pipe": pipe, "proba": proba})

print(pd.DataFrame(rows, columns=["Model","Accuracy","Precision","Recall","F1","ROC\_AUC","Brier\_MSE"]).round(4))

fi = top\_k\_importances(best["pipe"], k=10)

print("\nTop drivers of Churn (best model =", best["name"], ")")

print(fi.to\_string(index=False))

print\_prescriptive(fi, "Churn")

best\_binary = ("Churn", yte, best["proba"])

# ===============================================================

# 3) DEFAULTED CLASSIFICATION (only if varied 0/1)

# ===============================================================

if "Defaulted" in df.columns and df["Defaulted"].nunique() > 1:

header("Analytic 3 — Defaulted (console only)")

target = "Defaulted"

feats = [c for c in NUMERIC + CATEG if c != target and c in df.columns]

X = df[feats].copy()

y = df[target].astype(int).values

prep, num\_cols, cat\_cols = get\_preprocessor(df, feats)

models = {

"LogReg": Pipeline([("prep",prep),("clf",LogisticRegression(max\_iter=2000))]),

"RandomForest": Pipeline([("prep",prep),("clf",RandomForestClassifier(n\_estimators=400, random\_state=SEED))]),

}

Xtr, Xte, ytr, yte = train\_test\_split(X, y, test\_size=TEST\_SIZE, random\_state=SEED, stratify=y)

rows, best = [], {"auc": -1, "name": None, "pipe": None, "proba": None}

for name, pipe in models.items():

pipe.fit(Xtr, ytr)

proba = pipe.predict\_proba(Xte)[:,1]

pred = (proba >= 0.5).astype(int)

acc = accuracy\_score(yte, pred)

prec = precision\_score(yte, pred, zero\_division=0)

rec = recall\_score(yte, pred, zero\_division=0)

f1 = f1\_score(yte, pred, zero\_division=0)

auc = roc\_auc\_score(yte, proba)

brier = brier\_score\_loss(yte, proba)

rows.append([name, acc, prec, rec, f1, auc, brier])

if auc > best["auc"]:

best.update({"auc": auc, "name": name, "pipe": pipe, "proba": proba})

print(pd.DataFrame(rows, columns=["Model","Accuracy","Precision","Recall","F1","ROC\_AUC","Brier\_MSE"]).round(4))

fi = top\_k\_importances(best["pipe"], k=10)

print("\nTop drivers of Default (best model =", best["name"], ")")

print(fi.to\_string(index=False))

print\_prescriptive(fi, "Default")

# prefer default model for buckets if available

best\_binary = ("Defaulted", yte, best["proba"])

# ===============================================================

# 4) Risk Buckets (Deciles) — console table only

# ===============================================================

if best\_binary is not None:

label, y\_holdout, p\_holdout = best\_binary

header(f"Analytic 4 — Risk Buckets (Deciles) for {label}")

dfp = pd.DataFrame({"y": y\_holdout, "p": p\_holdout})

dfp["decile"] = pd.qcut(dfp["p"], 10, labels=False, duplicates="drop") # 0=lowest risk

agg = dfp.groupby("decile").agg(

customers=("y","count"),

positives=("y","sum"),

avg\_prob=("p","mean")

)

agg["rate"] = (agg["positives"] / agg["customers"].clip(lower=1)).round(4)

print(agg[["customers","positives","rate","avg\_prob"]].sort\_index())

print("\nAction: focus retention/collections on the highest deciles (largest rates/avg\_prob).")

else:

header("Analytic 4 — Risk Buckets")

print("Skipped: need a binary model with probability outputs (Churn or Defaulted).")

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**4. Business Recommendations**

Sales Growth Strategies

* Leverage seasonality by aligning campaigns with high-demand periods (e.g., festive or seasonal offers).
* Target repeat buyers with loyalty programs, as purchase frequency is a key sales driver.
* Personalize promotions using insights from age, gender, and spending score, ensuring marketing spend generates higher ROI.

Churn Management Strategies

* Focus on customers with high sales but lower credit scores, as they are more prone to churn.
* Implement proactive retention campaigns (discounts, tailored offers) for at-risk groups identified by Random Forest churn predictions.
* Use predictive churn scores to prioritize customer service interventions before customers leave.

Risk Management Policies

* Strengthen credit screening for low-income, high-loan applicants to reduce defaults.
* Introduce tiered risk policies: stricter terms for customers in high-risk deciles, flexible repayment options for medium-risk, and upsell offers for low-risk.
* Monitor default probability buckets regularly to update lending and collection policies.

Customer Segmentation Actions

* Segment customers into risk-based buckets (low, medium, high) for tailored actions.
* Cross-sell premium products to low-risk, high-income customers.
* Provide financial advisory or repayment support for high-risk groups to reduce defaults and improve customer trust.
* Use demographic segmentation (age, gender, seasonality) to refine marketing campaigns.

**5. Conclusion**

This project successfully integrated Power BI for descriptive analytics with Spyder for predictive and prescriptive modelling, creating a full analytics loop. Power BI provided clear and interactive visualizations of sales, churn, and customer behaviour, helping identify trends and anomalies. Spyder extended this by applying machine learning models, which not only predicted outcomes such as sales performance, churn likelihood, and default risk, but also generated prescriptive insights on how to act on these predictions. Together, the tools enabled a comprehensive data-driven strategy: descriptive analytics explained what happened, predictive analytics forecasted what could happen, and prescriptive analytics recommended what actions to take. This end-to-end approach supports smarter decision-making, minimizes risks, and maximizes business growth opportunities.