Sales Prediction Project - Python Code

Below is the complete Python code used for the Sales Prediction project.

```
# Sales Prediction Project using Python

import os
import zipfile
import pandas as pd
import numpy as np
from pathlib import Path
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import joblib

ARCHIVE_PATH = "/mnt/data/archive (20).zip"
WORKDIR = "/mnt/data/sales_project_workdir"
os.makedirs(WORKDIR, exist_ok=True)

# 1. Unzip and choose largest CSV/XLSX
with zipfile.ZipFile(ARCHIVE_PATH, 'r') as z:
    z.extractall(WORKDIR)
files = list(Path(WORKDIR).rglob("*"))
tabular = [f for f in files if f.suffix.lower() in ['.csv', '.xlsx', '.xls']]
tabular = sorted(tabular, key=lambda p: p.stat().st_size, reverse=True)
data_file = tabular[0]

# 2. Load dataset
if data_file.suffix.lower() == '.csv':
    df = pd.read_csv(data_file, low_memory=False)
else:
    df = pd.read_excel(data_file, engine='openpyxl')

# 3. Detect sales column
lower_cols = {c.lower(): c for c in df.columns}
sales_col = None
for s in ['sales','revenue','amount','sales_amount','salesvalue','sales_value']:
    if s in lower_cols:
        sales_col = lower_cols[s]
        break
if sales_col is None:
    numeric = df.select_dtypes(include=[np.number]).columns.tolist()
    sales_col = df[numeric].var().sort_values(ascending=False).index[0]

# 4. Detect ad-related columns
ad_candidates = [c for c in df.columns if any(x in c.lower() for x in ['tv','radio','newspaper','ad','advert'])]

# 5. Build modeling table
use_cols = [c for c in [sales_col] + ad_candidates if c in df.columns]
DF = df[use_cols].copy()
```

```
# 6. Fill missing values
for c in DF.columns:
    if DF[c].dtype.kind in 'biufc':
        DF[c] = DF[c].fillna(DF[c].median())
    else:
        DF[c] = DF[c].fillna("unknown")

# 7. Encode categoricals
DF_encoded = pd.get_dummies(DF, drop_first=True)

# 8. Train/test split
X = DF_encoded.drop(columns=[sales_col])
y = DF_encoded[sales_col].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 9. Models
models = {
    "LinearRegression": LinearRegression(),
    "RandomForest": RandomForestRegressor(n_estimators=200, random_state=42, n_jobs=-1),
    "GradientBoosting": GradientBoostingRegressor(n_estimators=200, random_state=42)
}

results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    mse = mean_squared_error(y_test, preds)
    rmse = mse**0.5
    mae = mean_absolute_error(y_test, preds)
    r2 = r2_score(y_test, preds)
    results[name] = {"model": model, "mse": mse, "rmse": rmse, "mae": mae, "r2": r2}
    joblib.dump(model, os.path.join(WORKDIR, f"{name}.joblib"))

# 10. Save feature importances, scenario simulation, results summary
# (Code from notebook continues for EDA, scenario testing, and forecasts.)
```