Assignment 1: Risk Analysis & Forecasting

Management Science

Assignment Overview

Due: Start of Lecture 8 Weight: 30% of final grade Expected Time: 4-6 hours Work: Groups

You're a a group of junior analyst at a consulting firm. A client needs help with:

1. Understanding investment risk using simulation
2. Forecasting product demand

Consultants

Who is part of your group?

```
"""
YOUR ANSWER HERE:
"""
```

```
'\nYOUR ANSWER HERE:\n'
```

Setup

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

# Sets a random seed for reproducibility (no need to adjust this for you)
np.random.seed(42)
```

Part A: Investment Risk Simulation (50%)

Scenario

Your client wants to invest €100,000 in a new product launch. They need to understand the risk.

Given Information:

- Best case: €300,000 return (20% probability)
- Most likely: €150,000 return (60% probability)
- Worst case: €50,000 return (20% probability)

Task 1: Simple Monte Carlo Simulation (15%)

```python
# Create a function that runs simulations and randomly picks outcomes
def simulate_investment(n_simulations=100):
    """
    Run Monte Carlo simulation for investment returns.

    Returns:
        array: Simulation results
    """
    # Define possible outcomes and their probabilities
    outcomes = [300000, 150000, 50000]
    probabilities = [0.2, 0.6, 0.2]

    # Hint: use np.random.choice() with probabilities
    # YOUR CODE HERE


    return results

# Run simulation
# YOUR CODE HERE
```

Task 2: Add Uncertainty to Probabilities (15%)

```python
def simulate_with_uncertainty(n_simulations=100):
    """
    Simulate investment with uncertain probabilities.
    - Best case probability: 10% to 30% (uniform)
    - Worst case probability: 10% to 30% (uniform)
    - Most likely probability: remainder to sum to 100%
    """
    outcomes = [300000, 150000, 50000]
    results = []

    return results

# Run simulation with uncertainty
# YOUR CODE HERE
```

Task 3: Analysis & Visualization (20%)

```python
# Create histogram of returns
# YOUR CODE HERE

# Calculate key metrics
# 1. Probability of losing money (return < €100,000)
# 2. 95% Value at Risk (5th percentile)
# 3. Expected profit
# YOUR CODE HERE
```

Business Question: Should the client make this investment? Write 3-4 sentences explaining your recommendation.

```
"""
YOUR ANSWER HERE:
"""
```

```
'\nYOUR ANSWER HERE:\n'
```

Part B: Demand Forecasting (50%)

Scenario

A retail client needs to forecast demand for the next 3 months to plan inventory.

The Data

```python
# 24 months of sales data
sales_data = {
    'month': list(range(1, 25)),
    'sales_units': [
        120, 135, 155, 145,  # Spring Year 1
        165, 180, 195, 175,  # Summer Year 1
        150, 140, 125, 130,  # Fall Year 1
        110, 105, 115, 125,  # Winter Year 1/2
        140, 150, 170, 160,  # Spring Year 2
        175, 190, 210, 185   # Summer Year 2
    ]
}

# Convert to pandas DataFrame for easier handling
df = pd.DataFrame(sales_data)
print(df.head())
print(f"\nTotal months: {len(df)}")
print(f"Average sales: {df['sales_units'].mean():.1f}")
```

```
   month  sales_units
0      1          120
1      2          135
2      3          155
3      4          145
4      5          165

Total months: 24
Average sales: 152.1
```

Task 1: Moving Average Forecast (15%)

```python
def moving_average_forecast(data, window_size=3):
    """
    Calculate moving average forecast for next period.

    Args:
        data: List or array of historical values
        window_size: Number of periods to average

    Returns:
        float: Forecast for next period
    """
    # YOUR CODE HERE
    # Take the average of the last 'window_size' periods

    return forecast

# Test different window sizes and print the results
# YOUR CODE HERE
```

Task 2: Exponential Smoothing (15%)

Understanding Exponential Smoothing:

Exponential smoothing gives more weight to recent observations while still considering historical data. The formula is:

$$F_t = \alpha \times A_{(t-1)} + (1 - \alpha) \times F_{(t-1)}$$

Where:

- F_t = Forecast for period t
- $A_{(t-1)}$ = Actual value in previous period
- $F_{(t-1)}$ = Forecast for previous period
- α = Smoothing parameter between 0 and 1

How it works:

- Start with the first actual value as your initial forecast: $F_1 = A_1$
- For each subsequent period, update the forecast using the formula above
- Higher α (e.g., 0.7) = more weight on recent data (responsive to changes)
- Lower α (e.g., 0.1) = more weight on historical pattern (smooth, stable)

Example with α = 0.3:

- Month 1: Actual = 120, Forecast = 120 (initial)
- Month 2: Actual = 135, Forecast = 0.3 × 120 + 0.7 × 120 = 120
- Month 3: Actual = 155, Forecast = 0.3 × 135 + 0.7 × 120 = 124.5
- And so on…

```python
def exponential_smoothing(data, alpha=0.3):
    """
    Calculate exponential smoothing forecast for next period.
```

```
    Args:
        data: Historical values (list or array)
        alpha: Smoothing parameter (0-1)
                Higher alpha = more weight on recent data

    Returns:
        float: Forecast for next period
    """
    # Initialize forecast with first actual value
    forecast = data[0]

    # YOUR CODE HERE
    # Loop through data starting from index 1
    # For each actual value, update forecast using the formula:
    #   forecast = alpha * actual + (1 - alpha) * forecast

    # Return the final forecast (which predicts the next period)
    return forecast

# Test different alpha values and print the results
# Try alpha = 0.1 (smooth), 0.3 (balanced), 0.7 (responsive)
# YOUR CODE HERE
```

Task 3: Evaluate and Compare Methods (20%)

```
# Split data: first 20 months for training, last 4 for testing
train_data = df['sales_units'].values[:20]
test_data = df['sales_units'].values[20:]

# Generate forecasts for test period
ma_forecasts = []
es_forecasts = []

# For each test month:
# 1. Use training data to forecast
# 2. Add actual value to training for next forecast
# YOUR CODE HERE

# Calculate Mean Absolute Error (MAE)
# YOUR CODE HERE
```

```
# Visualization
# Add lines for:
# - Moving average forecasts
# - Exponential smoothing forecasts
# Mark the train/test split
# YOUR CODE HERE
```

Business Question: Which forecasting method would you recommend and why? How much safety stock should they hold (based on your assumption)? (3-4 sentences)

```
"""
YOUR ANSWER HERE:
"""
```

```
'\nYOUR ANSWER HERE:\n'
```

Submission Checklist

* ☐ All code cells run without errors
* ☐ Functions are commented
* ☐ Visualizations are clear and labeled
* ☐ Business questions are answered
* ☐ Names added to top of notebook

Tips

* Use AI tools to help understand concepts and programm, but make sure you under-stand the code
* Start simple - get something working before optimizing
* Remember: the goal is good solutions, not perfect ones

Bibliography