Take-Home Exercise

Estimated Time: 2-3 hours

Our investment banking division needs to improve the efficiency of our IPO (Initial Public Offering) pricing recommendations. You've been asked to build a prototype model that predicts whether a newly public company's stock will perform well (positive returns) in its first quarter post-IPO.

The Challenge

You will work with a dataset of historical IPO information and build a model to predict first-quarter performance. This exercise evaluates your ability to:

- Perform exploratory data analysis
- Handle real-world financial data issues
- Build and evaluate appropriate ML models
- Communicate findings to non-technical stakeholders

Dataset Description

You'll receive a CSV file (`ipo_data.csv`) with the following features:

Company Metrics:

- `company_age`: Years since company founding
- `employees`: Number of employees at IPO
- `revenue_millions`: Annual revenue in millions USD
- `revenue_growth_rate`: YoY revenue growth percentage
- `ebitda_margin`: EBITDA margin percentage
- `industry_sector`: Tech, Healthcare, Finance, Consumer, Industrial, Energy

IPO Characteristics:

- `offer_price`: Initial offering price per share
- `shares_offered_millions`: Number of shares offered
- `underwriter_rank`: Prestige ranking of lead underwriter (1-10)
- `venture_backed`: Binary indicator (0/1)
- `lockup_period_days`: Days until insiders can sell shares

Market Conditions:

- `market_volatility_index`: VIX level at IPO date
- `sector_performance_30d`: Sector index performance prior 30 days (%)
- `ipo_month`: Month of IPO (1-12)

Target Variable:

- `q1_return`: First quarter stock return (%) - **YOUR PREDICTION TARGET**
- Binary classification: 1 if return > 0%, 0 otherwise

Dataset size: ~400 companies, ~20% missing values in some features

Your Tasks

1. Exploratory Data Analysis (30-45 minutes)

- Examine data quality and handle missing values
- Generate visualizations that reveal important insights
- Document any data quality concerns

2. Feature Engineering (20-30 minutes)

- Create at least 1 meaningful new feature
- Explain your rationale
- Handle categorical variables appropriately

3. Model Development (45-60 minutes)

- Build at least 2 different classification models(we suggest: one simple interpretable model + one ensemble)
- Use appropriate train/test split and validation strategy
- Compare model performance using appropriate metrics

4. Business Communication (20-30 minutes)

- Summarize your findings in a brief executive summary (max 1 page)
- What are your model's top 3 most important features?
- What would you recommend as next steps?
- What are the limitations of your approach?

Deliverables

Please submit:

1. **Jupyter Notebook or Python script** with your complete analysis
   - Well-commented code
   - Clear section headings
   - Visualizations embedded
2. **Executive Summary** (PDF or Markdown)
   - Key findings
   - Model performance summary
   - Business recommendations
   - Limitations and next steps
3. **Requirements file** (requirements.txt or environment.yml)

Evaluation Criteria

We will assess your submission on:

Technical Skills (50%)

- Data cleaning and preprocessing approach
- Feature engineering creativity and rationale
- Model selection and implementation
- Proper use of validation techniques
- Code quality and organization

Problem Solving (25%)

- How you handle missing data
- Your approach to class imbalance (if present)
- Feature importance interpretation
- Awareness of overfitting risks

Communication (25%)

- Clarity of executive summary
- Quality of visualizations
- Documentation and comments
- Business-relevant insights

Notes

- **You may use any Python libraries** (scikit-learn, pandas, numpy, xgboost, etc.)
- **Focus on approach over perfection** - we want to see your thinking process
- **Document your assumptions** - there's no single "right answer"
- **Don't over-engineer** - a simple, well-explained solution is better than a complex, poorly documented one
- **Time management matters** - prioritize completing all sections over perfecting one

Bonus Points (Optional - Only if time permits)

- Implement a simple cross-validation strategy
- Address class imbalance if present
- Create an interpretable model explanation for stakeholders
- Discuss ethical considerations in IPO pricing

Mock Dataset Generation Code

Since you need to create test data, here's a sample generator:

python

```python
import pandas as pd
import numpy as np

np.random.seed(42)
n_samples = 400

# Generate synthetic IPO data
data = {
    'company_age': np.random.exponential(8, n_samples),
    'employees': np.random.lognormal(6, 1.5, n_samples),
    'revenue_millions': np.random.lognormal(4, 1.8, n_samples),
    'revenue_growth_rate': np.random.normal(30, 25, n_samples),
    'ebitda_margin': np.random.normal(-5, 15, n_samples),
    'industry_sector': np.random.choice(['Tech', 'Healthcare', 'Finance', 'Consumer', 'Industrial', 'Energy'], n_samples),
    'offer_price': np.random.lognormal(2.5, 0.6, n_samples),
    'shares_offered_millions': np.random.lognormal(2, 1, n_samples),
    'underwriter_rank': np.random.randint(1, 11, n_samples),
    'venture_backed': np.random.binomial(1, 0.6, n_samples),
    'lockup_period_days': np.random.choice([90, 180, 270, 365], n_samples),
    'market_volatility_index': np.random.normal(18, 6, n_samples),
    'sector_performance_30d': np.random.normal(2, 8, n_samples),
    'ipo_month': np.random.randint(1, 13, n_samples),
}

df = pd.DataFrame(data)

# Create target with some signal
score = (
    df['revenue_growth_rate'] * 0.02 +
    df['underwriter_rank'] * 0.15 +
    df['venture_backed'] * 0.3 -
    df['market_volatility_index'] * 0.05 +
    df['sector_performance_30d'] * 0.08 +
    np.random.normal(0, 1, n_samples)
)

df['q1_return'] = (score > score.median()).astype(int)

# Add missing values
missing_mask = np.random.random(df.shape) < 0.1
df = df.mask(missing_mask)

df.to_csv('ipo_data.csv', index=False)
```

Good luck! We look forward to reviewing your approach.