```python
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression

# Function to load and preprocess CSV files
def load_and_preprocess(file_path):
    data = pd.read_csv(file_path, index_col=0)
    data.columns.name = 'Year'  # Set name for the columns
    data.index.name = None       # Set the index name to None
    # Replace unwanted characters and convert to numeric
    new_data = data.replace({',': '', '-': ''}, regex=True).apply(pd.to_numeric, errors='coerce')
    return new_data

# Load the CSV files
bs_data = load_and_preprocess('bs.csv')      # Balance Sheet
is_data = load_and_preprocess('is.csv')      # Income Statement
ratios_data = load_and_preprocess('ratios.csv')  # Ratios

# Now you have bs_data, is_data, and ratios_data preprocessed and ready to use
```

Section 1: CAMELS Analysis - Concept and Significance

The CAMELS analysis is a framework used to evaluate a bank's health based on six critical components:

1. **Capital Adequacy (C):** Evaluates the bank's capital to absorb potential losses.
2. **Asset Quality (A):** Assesses the bank's asset portfolio, especially non-performing assets (NPAs).
3. **Management (M):** Analyzes the efficiency and soundness of management practices.
4. **Earnings (E):** Examines profitability and the stability of income.
5. **Liquidity (L):** Measures the bank's ability to meet short-term obligations.
6. **Sensitivity to Market Risk (S):** Assesses vulnerability to market changes, like interest rate fluctuations.

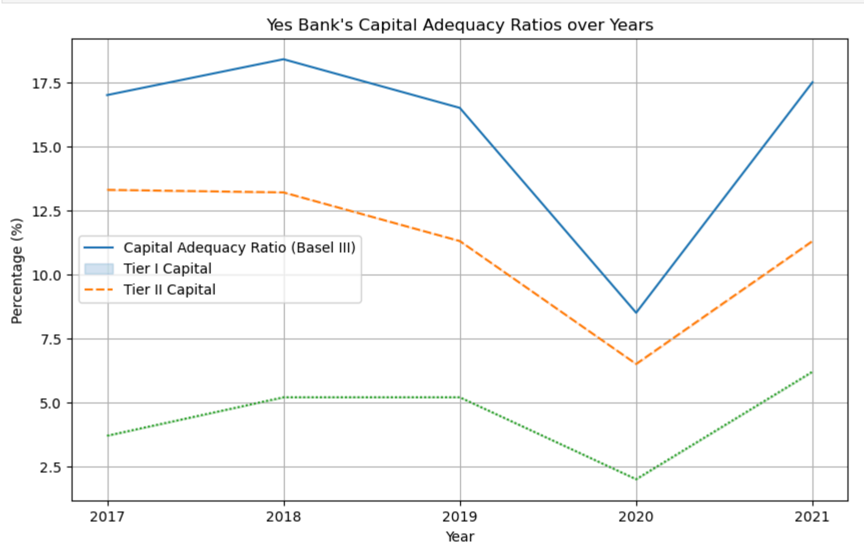
Each factor reflects specific risks or stability indicators, making CAMELS a comprehensive tool for regulatory and financial evaluations.

Section 2: Yes Bank CAMELS Analysis

1. Capital Adequacy

Capital Adequacy is crucial as it reflects the bank's capacity to absorb losses. Let's visualize the trends in Capital Adequacy Ratio (CAR), Tier I Capital, and Tier II Capital over the years to identify any significant drops that might indicate financial vulnerability.

```python
# Plotting Capital Adequacy Ratios (CAR), Tier I, and Tier II over years
plt.figure(figsize=(10, 6))
sns.lineplot(data=ratios_data.loc[['Capital Adequacy Ratio (%) - Basel III', 'Tier I Capital (%)', 'Tier II Capital (%)']].T)
plt.title("Yes Bank's Capital Adequacy Ratios over Years")
plt.xlabel("Year")
plt.ylabel("Percentage (%)")
plt.legend(['Capital Adequacy Ratio (Basel III)', 'Tier I Capital', 'Tier II Capital'])
plt.grid(True)
plt.show()
```

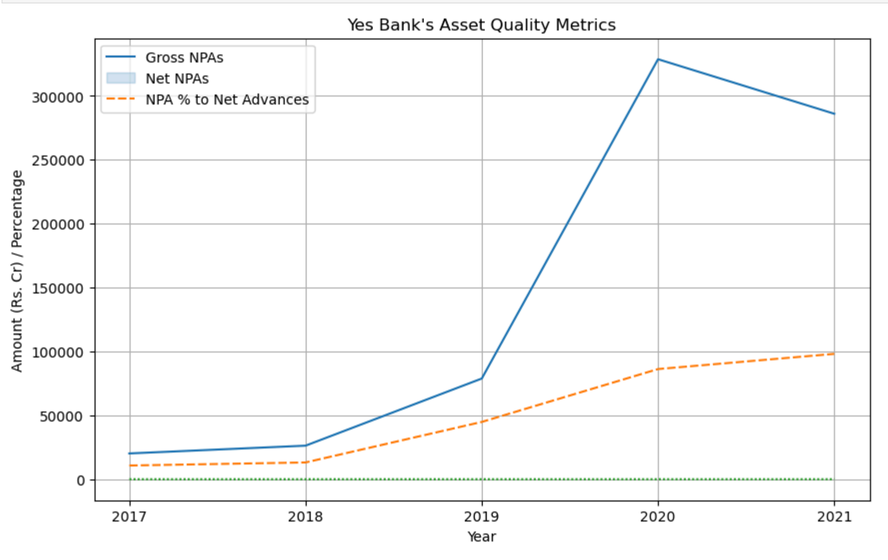


Interpretation: Look for significant declines in the CAR and Tier I ratios. For instance, a drop in 2020 could suggest that Yes Bank's capital buffer became insufficient during that period.

2. Asset Quality

Asset quality is assessed by examining trends in Gross NPAs, Net NPAs, and the percentage of NPAs to Net Advances. Let's track the increase in these figures to see how asset quality has deteriorated.

```python
# Plotting Gross NPAs, Net NPAs, and % of NPAs to Net Advances
plt.figure(figsize=(10, 6))
sns.lineplot(data=ratios_data.loc[['Gross Non-Performing Assets (Rs. Cr)', 'Net Non Performing Assets (Rs. Cr)', '% of Net Non-Performing Assets to Net Advance']].
plt.title("Yes Bank's Asset Quality Metrics")
plt.xlabel("Year")
plt.ylabel("Amount (Rs. Cr) / Percentage")
plt.legend(['Gross NPAs', 'Net NPAs', 'NPA % to Net Advances'])
plt.grid(True)
plt.show()
```

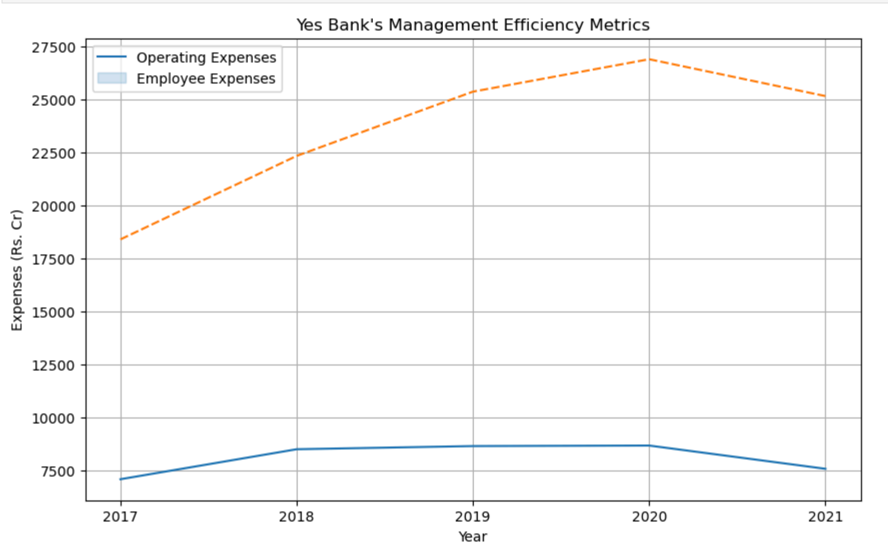


Interpretation: Track the sharp increase in NPAs, particularly if Gross NPAs began rising sharply after 2018 or 2019. This rise would indicate growing asset quality issues.

3. Management Efficiency

For management efficiency, we can examine the trends in Operating Expenses, Payments to Employees, and Other Expenses to gauge how well Yes Bank managed its operational costs.

```python
# Plotting Operating Expenses and Employee Expenses over years
plt.figure(figsize=(10, 6))
sns.lineplot(data=is_data.loc[['Operating Expenses & Administrative Expenses', 'Payments to/Provisions for Employees']].T)
plt.title("Yes Bank's Management Efficiency Metrics")
plt.xlabel("Year")
plt.ylabel("Expenses (Rs. Cr)")
plt.legend(['Operating Expenses', 'Employee Expenses'])
plt.grid(True)
plt.show()
```

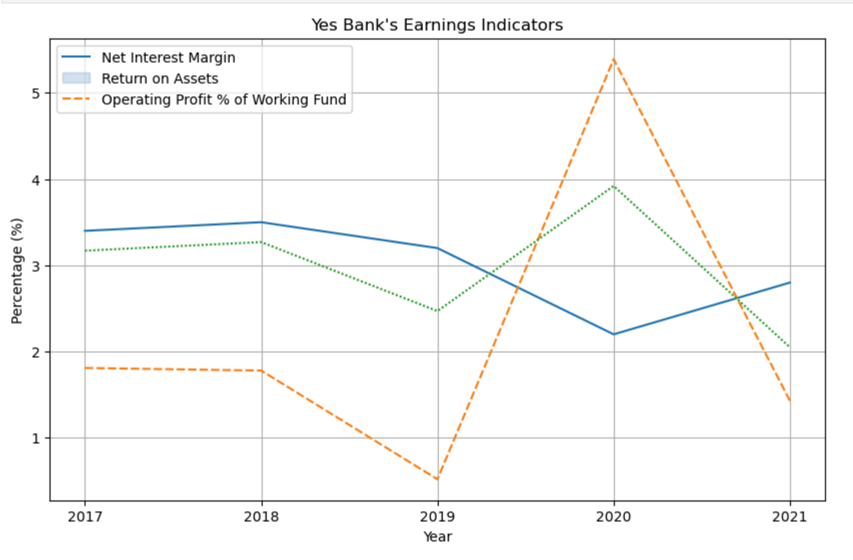


Interpretation: Steady or well-managed expenses reflect efficient management, while sudden increases may point to inefficiencies.

4. Earnings

For earnings, we'll review Net Interest Margin (NIM), Return on Assets (ROA), and Operating Profit as a percentage of the Average Working Fund to understand profitability trends.

```python
# Plotting Net Interest Margin, ROA, and Operating Profit % Average Working Fund
plt.figure(figsize=(10, 6))
sns.lineplot(data=ratios_data.loc[['Net Interest Margin(%)', 'Return on Assets (%)', 'Operating Profit % Average Working Fund']].T)
plt.title("Yes Bank's Earnings Indicators")
plt.xlabel("Year")
plt.ylabel("Percentage (%)")
plt.legend(['Net Interest Margin', 'Return on Assets', 'Operating Profit % of Working Fund'])
plt.grid(True)
plt.show()
```

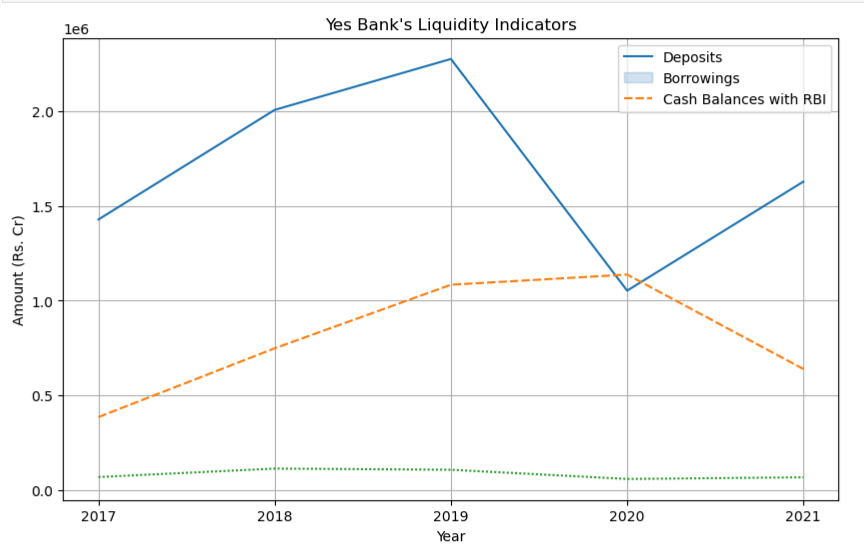


Interpretation: Dips in NIM and ROA, especially post-2019, could indicate profitability pressures, signaling distress.

5. Liquidity

We'll examine liquidity by analyzing trends in Deposits, Borrowings, and Cash Balances.

```python
# Plotting Deposits, Borrowings, and Cash Balances
plt.figure(figsize=(10, 6))
sns.lineplot(data=bs_data.loc[['Deposits', 'Borrowings', 'Cash & Balances with RBI']].T)
plt.title("Yes Bank's Liquidity Indicators")
plt.xlabel("Year")
plt.ylabel("Amount (Rs. Cr)")
plt.legend(['Deposits', 'Borrowings', 'Cash Balances with RBI'])
plt.grid(True)
plt.show()
```

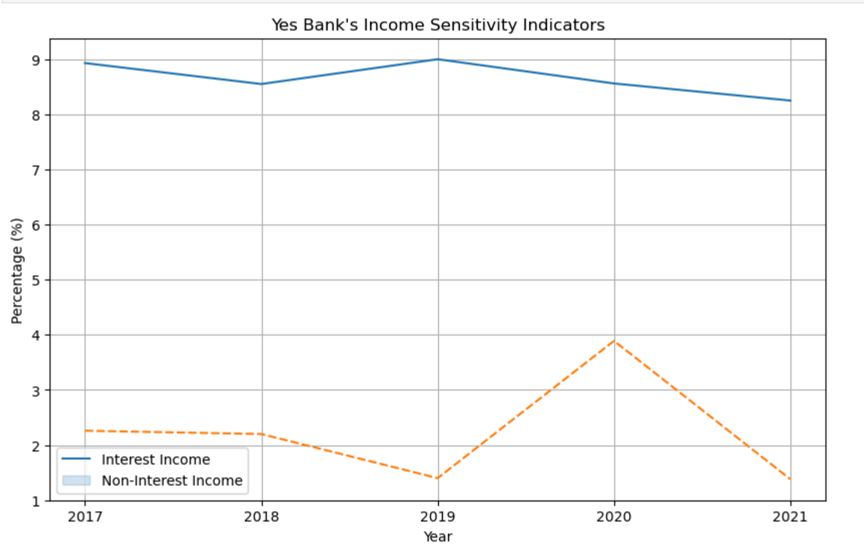


Interpretation: Declining deposits or heavy borrowings, particularly after 2019, might indicate liquidity stress.

6. Sensitivity to Market Risk

Sensitivity is often reflected in metrics such as interest income vs. non-interest income. A sudden increase in non-interest income (often from volatile sources) might suggest riskier revenue strategies.

```python
# Plotting Interest Income and Non-Interest Income as % of Average Working Fund
plt.figure(figsize=(10, 6))
sns.lineplot(data=ratios_data.loc[['Interest Income % Average Working Fund', 'Non Interest Income % Average Working Fund']].T)
plt.title("Yes Bank's Income Sensitivity Indicators")
plt.xlabel("Year")
plt.ylabel("Percentage (%)")
plt.legend(['Interest Income', 'Non-Interest Income'])
plt.grid(True)
plt.show()
```



Interpretation: Look for higher reliance on non-interest income, which could indicate sensitivity to market risks.

Section 3: Financial Distress and Contributing Factors

Financial distress in banking occurs when a bank struggles to meet its financial obligations due to poor asset quality, low capital reserves, or insufficient liquidity. Common causes include:

- **High Levels of NPAs:** Non-performing assets reduce earning assets and increase risk.
- **Low Profit Margins:** Declining profits reduce the bank's ability to absorb losses.
- **Insufficient Capital Reserves:** Low capital adequacy undermines the ability to cover potential losses.

Section 4: Significance of the PFD Model

The Probability of Financial Distress (PFD) model quantifies the likelihood that a bank may experience financial distress. The PFD model considers factors like capital adequacy, asset quality, and liquidity to estimate the risk of default. This model helps regulators and investors assess financial stability and implement risk management strategies.

Section 5: Estimating Yes Bank's Financial Distress with the PFD Model (Detailed)

To build a more thorough PFD model, we'll use logistic regression with relevant ratios to estimate Yes Bank's financial distress likelihood. A higher predicted probability suggests increased distress.

Step 1: Feature Selection and Data Preparation

For the PFD model, we can use the Capital Adequacy Ratio, Net NPAs, ROA, and other ratios that indicate distress. To enhance model accuracy, we'll also add new features derived from trends, such as year-over-year change in NPAs.

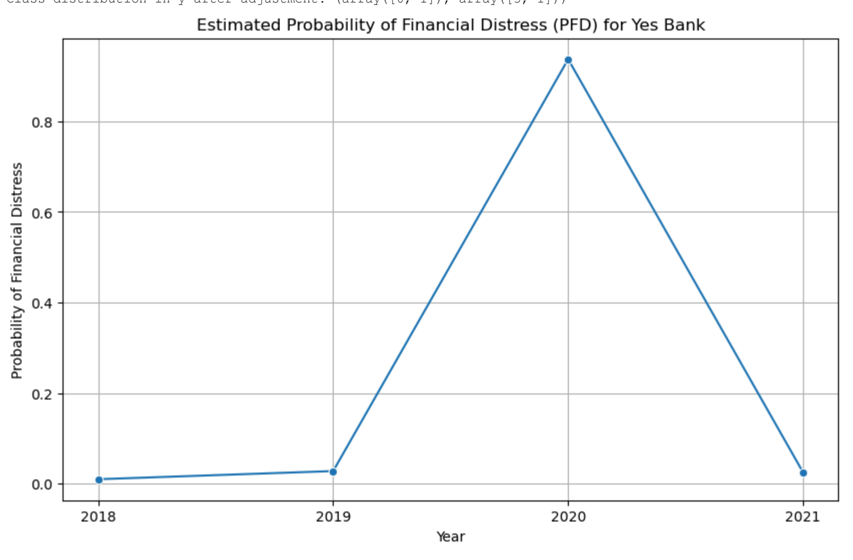
```python
# Add new features: Year-over-year change in Gross and Net NPAs
ratios_data.loc['Gross NPAs YoY Change'] = ratios_data.loc['Gross Non-Performing Assets (Rs. Cr)'].pct_change()
ratios_data.loc['Net NPAs YoY Change'] = ratios_data.loc['Net Non Performing Assets (Rs. Cr)'].pct_change()

# Selecting features for PFD model
X = ratios_data.loc[['Capital Adequacy Ratio (%) - Basel III', '% of Net Non-Performing Assets to Net Advance',
                     'Return on Assets (%)', 'Gross NPAs YoY Change', 'Net NPAs YoY Change']].T.dropna()

# Define target variable based on Return on Assets (assuming distress when ROA is negative)
y = np.where(X['Return on Assets (%)'] < 0, 1, 0)  # Example threshold

# Fill missing values with zero for simplicity in derived features
X = X.fillna(0)
```

Step 2: Model Training and Probability Prediction

Using logistic regression, we'll train the model on these selected features and plot the distress probability over time.

```python
# Check the distribution of classes in y
print("Class distribution in y:", np.unique(y, return_counts=True))

# If all are 0, try modifying the condition for y
# Example: Assume distress if either ROA is negative or Capital Adequacy Ratio is below a threshold (e.g., 10%)
y = np.where((X['Return on Assets (%)'] < 0) | (X['Capital Adequacy Ratio (%) - Basel III'] < 10), 1, 0)

# Re-check the distribution of classes in y
print("Class distribution in y after adjustment:", np.unique(y, return_counts=True))

# Fit the model with the adjusted y
model = LogisticRegression()
model.fit(X, y)
pfd_prob = model.predict_proba(X)[:, 1]  # Probability of Financial Distress

# Plotting the Probability of Financial Distress over Years
plt.figure(figsize=(10, 6))
sns.lineplot(x=X.index, y=pfd_prob, marker='o')
plt.title("Estimated Probability of Financial Distress (PFD) for Yes Bank")
plt.xlabel("Year")
plt.ylabel("Probability of Financial Distress")
plt.grid(True)
plt.show()
```

```
Class distribution in y: (array([0]), array([4]))
Class distribution in y after adjustment: (array([0, 1]), array([3, 1]))
```

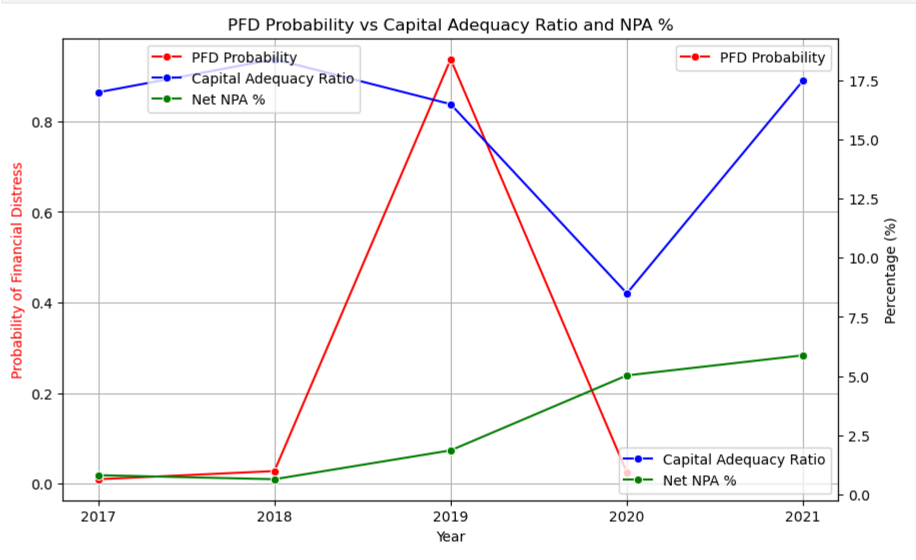


Interpretation: Observe if the PFD probability significantly increases around 2019-2020, which may correspond to distress.

Step 3: Further Trend Analysis with Key Ratios

For added insights, let's plot more key ratios used in the PFD model alongside distress probability.

```python
# Plotting PFD Probability with Capital Adequacy Ratio and NPAs
fig, ax1 = plt.subplots(figsize=(10, 6))

# Plotting Probability of Financial Distress
sns.lineplot(x=X.index, y=pfd_prob, marker='o', ax=ax1, label='PFD Probability', color='red')
ax1.set_ylabel("Probability of Financial Distress", color='red')
ax1.set_xlabel("Year")
ax1.grid(True)

# Plotting Capital Adequacy Ratio and % of Net NPAs on secondary axis
ax2 = ax1.twinx()
sns.lineplot(data=ratios_data.loc['Capital Adequacy Ratio (%) - Basel III'], marker='o', ax=ax2, label='Capital Adequacy Ratio', color='blue')
sns.lineplot(data=ratios_data.loc['% of Net Non-Performing Assets to Net Advance'], marker='o', ax=ax2, label='Net NPA %', color='green')
ax2.set_ylabel("Percentage (%)", color='black')

fig.legend(loc="upper left", bbox_to_anchor=(0.1,1), bbox_transform=ax1.transAxes)
plt.title("PFD Probability vs Capital Adequacy Ratio and NPA %")
plt.show()
```



Interpretation: Look for correlations between high distress probability, low CAR, and high NPA %, indicating where Yes Bank faced critical financial distress.

Conclusion

This detailed analysis gives a multi-dimensional view of Yes Bank's financial health, helping identify trends and pinpoint distress periods.