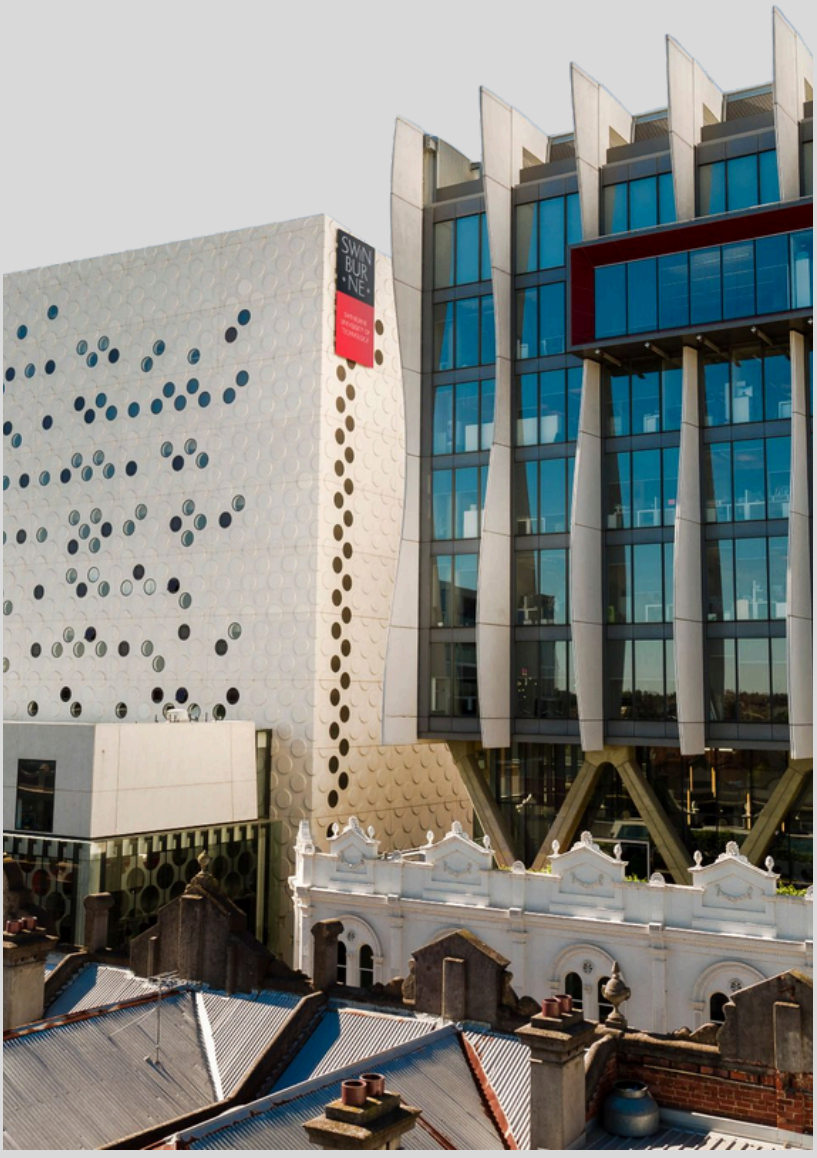
Swinburne University of Technology

TASK B.0 REPORT

COS30018
Intelligent Systems



Instructor:
Mr Aiden Nguyen

Group 4:
Le Hoang Triet Thong
104171146

Table of Contents

Report Task B0

1. Core Concepts Demonstrated:

Data Source Integration (yfinance library): The script successfully utilizes the yfinance library, a popular and powerful open-source tool that provides a simple interface to download historical and near real-time market data from Yahoo! Finance. This is a critical skill, as it automates the data collection process.

Targeting a Specific Asset: The line yf.Ticker("CBA.AX") demonstrates how to target a specific financial instrument. In this case, it's the Commonwealth Bank of Australia (CBA) on the Australian Securities Exchange (AX). This shows the importance of using the correct ticker symbol for the desired asset and exchange.

Fetching Historical Data: The use of ticker.history(period="10y") shows how to retrieve a substantial amount of historical data. The data is returned as a pandas DataFrame, a standard data structure in Python for data analysis, which includes columns like Open, High, Low, Close, and Volume for each trading day.

Data Persistence: By using data.to_csv("cba_stock_data.csv"), the script demonstrates how to save the downloaded data into a local CSV file. This is a crucial step for efficiency and reproducibility, as it allows you to run analyses multiple times without having to re-download the data from the internet.

Accessing Current Information: The use of ticker.fast_info['last_price'] shows how to quickly fetch the most recent trading price, which is essential for applications needing up-to-date information.

2. Overall Lesson:

This script perfectly encapsulates the **data acquisition phase** of a financial analysis workflow. Before any prediction or modeling can occur, one must have a reliable, automated way to gather clean, well-structured historical and current data. This code provides a robust and straightforward template for that exact purpose.