|  |
| --- |
| A–LEVEL COMPUTER SCIENCE PROJECT |
|  |
| October 20  OCR A LEVEL COMPUTER SCIENCE NEA  Authored by: SAMIR OUARBYA |



# Contents

aNALYSIS

* Problem Identification
* Stakeholders
* Why it is suited to a computational solution

design

DEVELOPMENT AND TESTING

EVALUATION

FINAL CODE

SECTION 1 – ANALYSIS

pROBLEM IDENTIFICATION

Like stocks, the market for rare shoes fluctuates depending on multiple factors. In the stock market, investors recognize patterns in the relationship between time and stock price. For example, a certain stock may experience a short time rise every day at around 9am, trading 'robots' recognize these patterns and use AI to develop a winning strategy. Not unlike this, I am experimenting with trying to predict market prices using data from a shoe market website.

STAkeholders

To my knowledge, this has not been done before, and I think that it could be useful to try and analyze trends in sales to identify if a certain shoe will spike in price or not. Factors that need to be considered to solve this problem are things such as: the volatility past sales, the volume of sales in the last 24 hours etc. All of which will be scraped from the StockX API.  In a business setting, this would be useful as vendors can identify which shoes will have a popularity spike, such that they can purchase stock before it becomes more expensive.

The demographic for this software that would be vendors of these sort of products, so the software will need to be developed in a business-friendly manner, for example an easy to use UI, easy to read graphs, information and pictures of the shoe.

The casual consumer/ collector of rare shoes is also a stakeholder as they can identify the right time to purchase a shoe to add to their collection.

The shoe re-selling market is an ever-growing industry, which was recently estimated to be a $2 billion market, experts also estimate it to triple in size over the next few years, projecting it to reach more than $6 billion by 2025.

The ability to predict the market/ popularity of a shoe to some degree of accuracy is applicable in a business and a casual setting.

why it is suited to a computational solution

Analyzing sales trends is something that can be achieved manually, like real stocks, however developing a solution such as the one I have described is far more efficient via an application due to the fact that it should be able to make predictions to a higher level of accuracy than a human, it also has the ability to monitor multiple shoes at once which a human may not be able to do.

COMPUTATIONAL METHODS THE SOLUTION LENDS ITSELF TO:

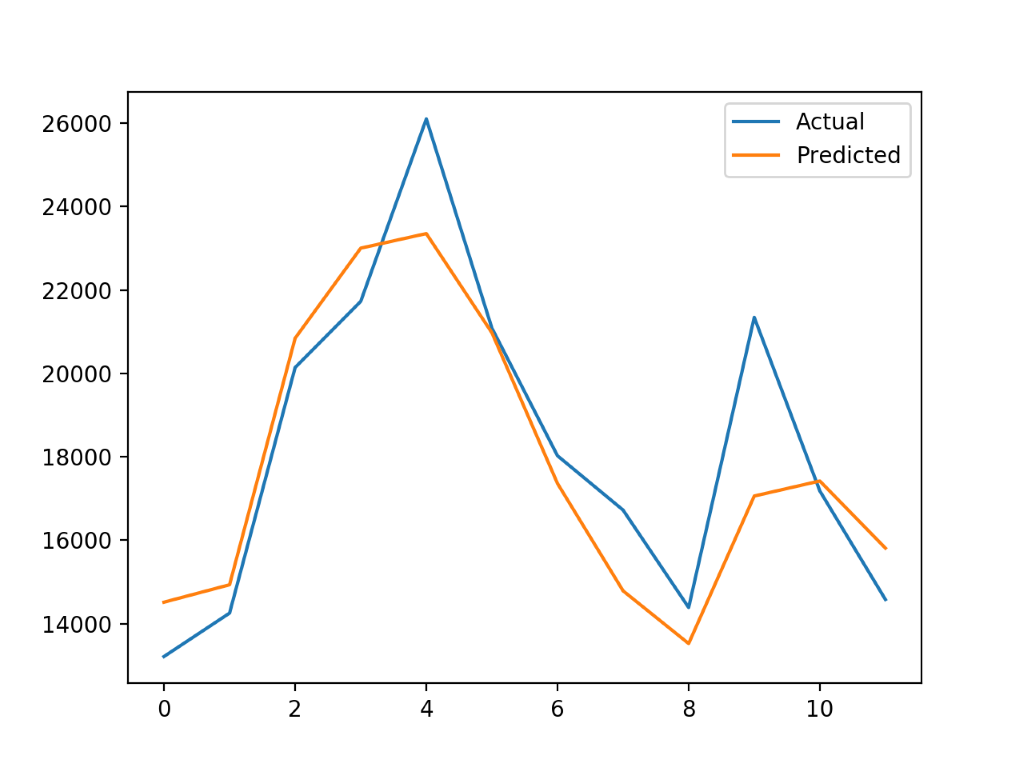
### PROBLEM RECOGNISION

The problem at hand is finding a way to supply a machine learning algorithm with accurate real-time and past sales data such that it can model future sales to a good degree of accuracy. This also should be done in a way that is easily visualized so that it is easily accessible to the user.

### PROBLEM DECOMPOSITION

The problem should be decomposed into a set of smaller steps, this is a very broad description of these steps:

1. Dynamically scrape data from the StockX API and store adequately in a database
2. Serve this scraped data to the machine learning algorithm to generate predicted data.
3. Serve scraped and generated data to the frontend to provide a visualization and an easy to use application.

If done correctly, the app should be able to display a graph plotting real time sales with predicted sales in a time-series graph for a particular product like so:  


### DIVIDE AND CONQUER

I will be using a divide and conquer approach to developing the app to ensure my code is modular and to minimize developing time.

### ABSTRACTION

The StockX API provides a large amount of data such as aesthetic data which will not necessarily be needed to achieve the solution, abstraction will need to be used to ensure that only relevant data is being fed to the algorithm.

Not all of the data scraped will need to be displayed in the frontend so collected data will need to be abstracted to suit the needs of the user.

