ABSTRACT

Stock price prediction is one among the complex machine learning problems. It depends on a large number of factors which contribute to changes in the supply and demand. This project presents a time series approach using Linear Regression, for predicting the price of Apple’s stock, and evaluation of the approach for the same. Stock prices are represented as time series data and trading indications which let the user know about when to perform the trading and when to not. We will be using the popular scikit-learn framework to build our model and evaluate it.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapter. No** | **Title** | **Page No.** |
|  | **Abstract** | **i** |
|  | **Table of Contents** | **ii** |
|  | **List of Figures** | **iv** |
|  | **Abbreviation** | **v** |
| **1.** | **INTRODUCTION** | **1-3** |
|  | 1.1 Introduction | 1 |
|  | 1.2 Existing System | 1 |
|  | 1.3 Disadvantages | 2 |
|  | 1.4 Proposed System | 2 |
|  | 1.5 Advantages | 2 |
|  | 1.6 Problem Definition | 3 |
| **2.** | **REQUIREMENT ANALYSIS** | **4-5** |
|  | 2.1 System Requirement | **4** |
|  | 2.2 Software Requirement | 4 |
|  | 2.3 Functional Requirement | 4 |
|  | 2.4 Technologies Used | 5 |
|  | 2.5 Programming Language Used | 5 |
| **3** | **DESIGN ANALYSIS** | **6-9** |
|  | 3.1 Data Flow Diagram: | 6 |
|  | 3.2 System Architecture | 7 |
|  | 3.3 Class Diagram | 8 |
|  | 3.4 Component Diagram | 9 |
| **4** | **ALGORITHM & MODULES** | **10-33** |
|  | 4.1 Algorithm  4.1.1 Linear regression  4.1.2 Regression vs. classification | 10  10  10 |
|  | 4.2 Modules  4.2.1 Numpy  4.2.2 Pandas  4.2.3 Seaborn  4.2.4 Matplotlib  4.2.5 Scikit-learn | 15  15  16  17  24  30 |
| **5** | **IMPLEMENTATION** | **34-35** |
|  | 5.1 Implementation | 34 |
|  | 5.2 Source Code | 34 |
| **6** | **TESTING & VALIDATION** | **36-40** |
|  | 6.1 Testing Methodologies | 36 |
|  | 6.2 Unit Testing | 36 |
|  | 6.3 Integration Testing | 37 |
|  | 6.4 User Acceptance Testing | 38 |
|  | 6.5 Output Testing | 38 |
|  | 6.6 Validation Checking | 38 |
|  | 6.7 Using Live Test Data | 39 |
|  | 6.8 Using Artificial Test Data | 39 |
|  | 6.9 Testing Strategy | 40 |
| **7** | **SCREENSHOTS** | **41-42** |
|  | 7.1 Screenshots  7.1.1 Stock Data  7.1.2 Graph Representation of Stock | 41  41  42 |
| **8** | **CONCLUSION** | **43-44** |
|  | 8.1 Conclusion | 43 |
|  | 8.2 Limitation | 43 |
|  | 8.3 Future Scope | 44 |
| **9** | **REFERENCES** | 45 |

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Title** | **Page No.** |
| 3.1 | Data Flow Diagram | 6 |
| 3.2 | System Architecture | 7 |
| 3.3 | Class Diagram | 8 |
| 3.4 | Component Diagram | 9 |
| 4.1 | Cricket Chirps per Minute | 11 |
| 4.2 | Cricket Chirps Per Minute Temperature in Celsius | 12 |
| 4.3 | Time point | 20 |
| 4.4 | Day Total Bill | 21 |
| 4.5 | level and axes-level functions | 22 |
| 4.6 | Time rate | 23 |
| 7.1 | Stock Data | 41 |
| 7.2 | Graph Representation of Stock | 42 |

ABBREVIATION

|  |  |
| --- | --- |
| GUI | Graphical User Interface |
| PY | Python |
| OPP | Object Oriented Programming |
| DRY | Don't Repeat Yourself |
| PIP | Package Installer for Python |
| MRO | Method Resolution Order |
| PEP | Python Enhancement Proposals |
| BDFL | Benevolent Dictator For Life |
| REPL | Read-Eval-Print Loop |