Using Markov Chains Model and Decision Tree Classifier to Predict the Behavior of Palestinian Stock Market Prices - Case Study :Paltel

By

Areej Nasim Awwad

Supervisor

Prof. Dr. Saed Mallak

This Thesis was submitted in partial fulfillment of the requirements for the Master's Degree of Science in Mathematical Modeling Faculty of Graduate Studies

Palestinian Technical University-Kadoorie

April 2021

استخدام نموذج سلاسل ماركوف ومصنف شجرة القرارللتنبؤ بسلوك أسعار البورصة الفلسطينية
- دراسة حالة:بالتل

إعداد

أريج نسيم عوّاد

المشرف

أ.د.سائد ملاك

قدمت هذه الرسالة استكمالاً لمتطلبات الحصول على درجة الماجستير في

النمذجة الرياضية

كلية الدراسات العليا

جامعة فلسطين التقنية – خضوري

نيسان 2021

جامعة فلسطين التقنية – خضوري

نموذج التفويض

أنا أريج نسيم أحمد عوّاد ، أفوض جامعة فلسطين التقنية خضوري بتزويد نسخ من رسالتي / أطروحتي للمكتبات والمؤسسات أو الهيئات أو الأشخاص عند طلبهم حسب التعليمات النافذة في الجامعة.

التوقيع:

التاريخ:

Palestine Technical University - Kadoorie PTUK

Authorization form

I, Areej Nasim Ahmad Awwad , authorize the PTUK to supply copies of my thesis / dissertation to libraries or establishments or individuals on request, according to the university of PTUK regulations.

Signature:

Date:

الاقرار

انا الموقع ادناه مقدم الرسالة التي تحمل العنوان:

Using Markov Chains Model and Decision Tree Classifier to Predict the Behavior of Palestinian Stock Market Prices - Case Study :Paltel.

أقر بأن ما اشتملت عليه هذه الرسالة انما هو نتاج جهودي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد ، وان هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة علمية أو بحث علمي لدى مؤسسة علمية او بحثية اخرى.

Declaration

I hereby declare that this thesis is the product of my own efforts, except what has been referred to, and this thesis as a whole or any part of t has not been submitted as a requirement for attaining a scientific degree to any other educational or research institution.

Areej Nasim Ahmad Awwad

Signature :……

Date :………….

أريج نسيم أحمد عواد

التوقيع :.............

التاريخ :............

COMMITTEE DECISION

This thesis/dissertation Using Markov Chains Model and Decision Tree Classifier to Predict the Behavior of Palestinian Stock Market Prices - Case study:Paltel was succesfully defended and approved on 24-5-2021.

Examination committee **Signature**

Prof. Dr. Saed Mallak (Supervisor)

Asst. Prof. Dr. Mohammad Assad (External Examiner)

Asst. Prof. Dr. Ata Abu As'ad (Internal Examiner)

DEDICATION

I dedicate this thesis to my wonderful parents, my sisters and brother, my second family, my fiancé and my friends for their love, unfailing support and continuous encouragement throughout my life.

To everyone who inspires me by his/her science.

AKNOWLEDGEMENT

After thanking Allah, who granted me the ability to finish this work .I would like to express my sincere gratitude to my thesis supervisor: Prof. Dr. Saed Mallak for his guidance, understanding, support and sound advice in all aspects of my research work by steering me in the right direction whenever he thought I needed it. I would like to thank Dr. Hadi Khalilia for his scientific support, encouragement, guidance and constructive advice. I appreciate help and support from members of the Mathematical Department at Palestine Technical University – Kadoorie.

Finally, I must express my very profound gratitude to my parents who have always kept me in their prayers ,my sisters and my supportive brother for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis, thanks to my second family who add a distinct flavor to my life through their constant support and encouragement for me. My greatest gratitude to my fiancé for his care and unreserved support which gave me the courage and confidence that I need in my life by his presence in my life and my warm thanks to my friends for tirelessly supporting, motivating and helping me. This accomplishment would not have been possible without them . Thank you.

Contents

Using Markov Chains Model and Decision Tree Classifier to Predict the Behavior of Palestinian Stock Market Prices- Case study :Paltel.

By: Areej Awwad

Supervised by Prof. Dr. Saed Mallak

Abstract

Predicting the behavior of stock market prices has been one of the most poplular topic interest for researchers due to its complex and dynamic nature.In this thesis Markov Chains Model and Decision Tree Classifiers applied to forecast and analyse the behavior of the Palestinian Stock Market prices- case study :Paltel.

For this purpose, 243 days includus the daily opening price, volume and number of deals covering the period from 2nd January 2019 till 31th December 2019 of Paltel company were collected from Palestine Exchange .At the beginning the Markov Chain model was applied depending on the opening price where Markov Chain model is a probability model depend on transition probability matrix and initial state vector. Recognizing three states when the stock decrease "down", when the stock price increase "up" and when stock price unchanged "remain same" .

The transition probability matrix has been observed by monitoring the number of transforming from one state to another. The study showed that negligently of company’s current price index ,steady state probabilities vector of share "up" , "down" and "remain same" performe that in future Paltel stock index increases with probability 0.2840 decrease with probability 0.2660 and the probability for stock index unchanged is derived as 0.4500 . It noted that if the opening value of Paltel stock index is in the state "same" then it could be expected to return to the same state in two days.

We had also applied one of Data Mining techniques , Decision Tree Classifier.To creat the model, the CART methodology used covering real historical data including volume and number of deals of the Paltel company listed in Palestine Stock Exchange .The results showed that the future price remain same if both volume and deals are low-level ,if both volume and deals are high-level, volume is heavy-level with deals is high-level ,volume is high-level with deals is low-level and volume is heavy-level with deals is low-level then the predicted stock behavior is up .Whereas the future price will decrease if volume is average-level and volume is low-level with deal is high-level.

استخدام مصنف شجرة القرار ونموذج سلاسل ماركوف للتنبؤ بسلوك أسعار البورصة الفلسطينية -دراسة حالة: بالتل

إعداد: أريج عوّاد

بإشراف: أ.د.سائد ملاك

الملخص

يعتبر التنبؤ بسلوك أسعار البورصة من أكثر الموضوعات التي تهم الباحثين بسبب طبيعتها المعقدة والديناميكية ، وفي هذه الأطروحة تم تطبيق نموذج سلاسل ماركوف ومصنفات شجرة القرار للتنبؤ وتحليل سلوك أسعار سوق الأوراق المالية الفلسطينية. دراسة حالة (بالتل).

لهذا الغرض ، تم جمع 243 يومًا متضمنًا سعر الافتتاح اليومي والحجم وعدد الصفقات التي تغطي الفترة من 2 كانون الثاني (يناير) 2019 حتى 31 كانون الأول (ديسمبر) 2019 لشركة بالتل من بورصة فلسطين ، وفي البداية طبق نموذج سلسلة ماركوف اعتمادًا على سعر الافتتاح وهو نموذج احتمالي يعتمد على مصفوفة احتمالية الانتقال ومتجه الحالة الأولية .تم التعرف على ثلاث حالات "أسفل" عند انخفاض السهم ، و "أعلى" عند زيادة سعر السهم و "تبقى كما هي" في حالة عدم تغيير سعر السهم . تمت ملاحظة مصفوفة احتمالية الانتقال من خلال مراقبة عدد التحويلات من حالة إلى أخرى. تكشف الدراسة أنه وبغض النظر عن مؤشر الأسعار الحالي للشركة، متجه احتمالات الحالة المستقرة للسهم "لأعلى" و "لأسفل" و "يبقى كما هو"

يؤدي في المستقبل إلى زيادة مؤشر سهم بالتل مع احتمال 0.2840 وانخفاضه مع احتمال 0.2660 واحتمال مؤشر الأسهم يبقى كما هو 0.4500. لوحظ أنه إذا كانت القيمة الافتتاحية لمؤشر سهم بالتل في الحالة (same) ، فمن المتوقع أن تعود إلى نفس الحالة في اليوم الثاني.

لقد طبقنا أيضًا إحدى تقنيات التنقيب في البيانات ، وهي أداة تصنيف شجرة القرار وهي إحدى تقنيات التنقيب عن البيانات. لإنشاء النموذج ، تم استخدام منهجية CART التي تغطي البيانات التاريخية الحقيقية بما في ذلك الحجم وعدد صفقات شركة الاتصالات الفلسطينية المدرجة في بورصة فلسطين .أظهرت النتائج أن السعر المستقبلي سيبقى كما هو إذا كان الحجم وعدد الصفقات منخفضين المستوى ، اما إذا كان كل من الحجم والصفقات عالي المستوى ، الحجم مرتفع المستوى مع الصفقات عالية المستوى ، الحجم مرتفع مع الصفقات منخفضة المستوى والحجم ثقيل المستوى مع الصفقات منخفضة المستوى فإن سلوك السهم المتوقع سيرتفع ، بينما ينخفض السعر المستقبلي إذا كان الحجم متوسط المستوى و الحجم منخفض المستوى مع الصفقات عالية المستوى.

Chapter One

Introduction

The stock market is an area allowing everyone to invest in both national and international economies. Forecasting stock market behavior became an important research area due to its benefits not only for profitable industry but also for investors that allowing them taking a self-confident decision for a good investment in the stock market [12]." The stock market is essentially a non-linear, non- parametric system that is extremely hard to model with any reasonable accuracy" [3]. In the past few decades, many researches have been covered the possibility of employing past information to give meaningful prediction about the future behavior of stock market . In the beginning, assuming that the past information is helpful to predict the future behavior the solution provided by technical and fundamental analysis [11].

In stock market the correct decision taken by the individuals or organizations depends to a large extent on how many information you have about stocks, depending on that it possible to have useful analysis of statistical models that predict the behavior of the stock market .There are various statistical models including Moving average, Regression analysis, Time series, Markov chains, Machine learning and Data mining.

Markov Chain model due to its properties is one of the statistics model that has an important role in prediction. The occurrence of any event in the future relies on the present state, which is one of Markov Chain model property. "The difference between Markov model and other statistics methods (such as time series, etc.) that there is no need to find mutual laws between the factors into the complex predictor". By using Markov Chain model, after forming the initial probability distribution and transition probability matrix it is possible to predict the possibility of state value in a specific period [6].

As well, the stock market could be observed as a particular Data Mining problem. Making strategic business decisions often depend on Big Data . "Big Data refers to the huge amount of structured and unstructured data that overflow the organization. If the overflowed data is used in a proper way it leads to meaningful information. When Big Data compared to traditional databases, it includes a large number of data that requires more processing in real time. It also provides opportunities to discover new values, to understand an in-depth knowledge from hidden values and also provides space to manage those data effectively" [17].

Big Data concern multiple data sources including huge-volume, growing, complex datasets[17].

Big Data are expanding in all domains due to the rabidly development of networking, data storage and data collection capacity [17].

Recently, Data Mining techniques such as Decision Trees have improved in stock area .Data Mining refers to mining information from huge data sets. Its functions include several areas some of them are finding descriptions of concept, correlations and associations, classification, clustering, prediction, trend analysis, outlier and deviation analysis, and similarity analysis. Classification data could be done in many various ways one of these way is Decision Tree classifiers ."It is a graphical representation of all possible outcomes and the paths by which they may be reached ". Decision Trees could be trained by using a suitable learning algorithm. "Following the assumption of technical analysis that patterns exist in price data, it is possible to use data Mining techniques to discover these patterns in an automated manner. Once these patterns have been discovered, future prices behavior can be predicted" [3].

Many articles published to predict stock price behavior in the stock market by using various statistical methods. M. Kumar Bhusal (2017) aimed to apply the Markov Chain model to analyze behavior of Stock of Nepal . The study aimed to analyze the long run behavior of "NEPSE" index , the expected first return time of various states and to observe the expected number of visits to a specific state.

"NEPSE" index of 2741 days from August 15, 2007 to June 18, 2017 has calculated. Three different states noticed for "NEPSE" index includes increase , unchanged and decrease states.

The study explored that regardless of the present status of "NEPSE" index, in the long run the probability that index remains in same state is 0.1707 , increase with probability 0.3855, and decrease with probability 0.4436 [5]. Q. Al Radaideh (2013) attempted to decide the better timing for buying or selling stocks so the investors in the stock market have information about future stock depend on the information from the historical prices. The decision that had been taken depend on decision tree classifier. By building the proposed model, over real historical data of three companies from Amman Stock Exchange (ASE) [3]. Bairagi and Kakaty (2015) attempted to analyze the behavior of stock price of the State Bank of India (SBI), including the time period from 21st March 2011 to 20th March 2015. To analyze the behavior of SBI index the Markov Chain model applied .It was obtained that if the closing value of SBI share was in the state "up" in day one then it could be expected to return to the state "up" at the third day [4]. Adesokan et. al. (2017) studied the behavior of a stock on the Nigerian stock market.

The Markov Chain used to determine expected long and short-run returns, and the result compared to the expected return of the Capital Asset Pricing Model (CAPM).

The study cleared that regardless of present state of returns, the average returns observed after seventeen days.

Also depending on the present state, the average return realized after a maximum of two days. The expected return from the long and short-run expected returns of the Markov chain model computed and compared to the CAPM [2].

Stock market has a significant achievement in the rapidly growing economy. The fluctuation in stock market can have a deep influence on individuals and communities. The importance of the study can be summarized by:

1. Predicting Palestinian Stock Market behavior.
2. Determining the effect of the behavior of share prices on the efficiency of the Palestinian Stock Exchange.
3. Providing new investment opportunities for investors by revealing the movement of prices.

In this thesis, Markov Chains model and Decision Tree Classifier applied for the first time, to the best of our knowledge, to forecast and analyze the behavior of the Palestinian Stock Market Prices. The main problem is to:

1. Know the long-term behavior of stocks market prices by applying Markov Chain model.
2. Analyze the historical data available on stock exchange using decision tree as one of the classification methods of Data Mining. To encourage investors when to buy new stocks or to keep them.

This study based on historical daily prices of Paltel company shares, which was collected from the daily list published by the Palestinian Stock Exchange during the period from 2nd January 2019 till 31th December 2019. An important note is that during our collection period Paltel's stock chosen randomly.

In Chapter 2, an introduction and some basic definitions presented with properties of Markov Chain Model.

In Chapter 3, the Markov Chain Model implemented and results reached that encourage investors to make a decision to buy, sell or hold shares.

In Chapter 4, an introduction about Big Data and Data Mining also some basic definitions presented with an explanation of Decision Tree method and its characteristics.

In Chapter 5, through Decision Tree Classifiers, a model built that shows the behavior of stocks, depending on two variables that affect the stock price on a daily basis.

Finally, in Chapter 6, some conclusions and suggestion presented.

Chapter Two

Basics about Markov Chains

2.1. Stochastic Processe

Definition 1 "Let T be a subset of [0, ∞). A family of random variables $\{X_t\}_{\mathrm{t}\in T}$indexed by T is called a stochastic (or random) process. When T =$\{0,1,2,\ldots\}$, $\{X_t\}_{\mathrm{t}\in T}$ said to be a discrete-time process, and when T = [0, ∞), it called a continuous time process. In a common situation, the index t corresponds to discrete units of time"[18].

X_t meant to describe a measurable characteristic. Such that if there is a group of dogs, some with blue eyes and some have different color, the variable X_t could be used to label dogs with blue eyes. Supposedly, if there were three dogs have blue eyes within a given collection, the set X_t could be designated as the set of blue eyes dogs , with each dog labeled as X_1, X_2, or X_3.

X_t = number of blue doges at time t starting from t=0(initial state)

Stochastic processes notable by their state space S which is the range of possible values for the random variables X_t by their index set T and by the dependence relations among the random variables$\{X_t\}_{\mathrm{t}\in T}$.

2.2. Markov Chains

Definition 2 "A Markov process $\{X_t\}_{\mathrm{t}\in T}$ is a stochastic process with the property that, given the value of X_t , the values of X_v for v >t are not influenced by the values of X_z for z < t. In words, the probability of any particular future behavior of the process, when its current state known exactly, is not altered by additional knowledge concerning its past behavior. This property is called the memonlis property (Markov property)"[16].

Definition 3 "A discrete-time Markov Chain is a Markov process whose state space S is a finite or countable set, and whose (time) index set is T= {0, 1, 2, ...}"[16].

So the Markov property in formal terms is :

$$\mathrm{P}\{\ X_{t+1} = j \mid X_0 = i_0 \ \ldots\ X_{t-1} = i_{t-1}, X_t = i\}$$

$$=\mathrm{P}\{X_{t+1} = j \mid X_t = i\}.$$

for all time points t and all states $i_0 , \ldots , i_{t-1} , i, j \epsilon S$.

This means that X_{t+1} depends upon X_t but it does not depend upon $X_{t-1}, \ldots, X_1, X_0$.

It is normally known to label the state space S of a discrete Markov chain by a subset the non-negative integers {0, 1, 2, …,n} .

2.3. Transition Probability and Transition Probability Matrix

In the Markov Chain the transition probability called the transition or jump probability from state i to state j. Then,

$$\mathrm{P}\{X_{t+1} = j \,|X_t = i\} = p_{ij}^{t,t+1}$$

$p_{ij}^{t,t+1}$ indecates the probability that, whenever the chain in state i, transform next (one unit of time later) into state j, and is it known as a one-step transition probability for all i, jϵS. If the transition probabilities defined above are independent of time t, then it called homogenous or stationary transition probability Markov Chain. Thus,

$$\mathrm{P}\{X_{t+1} = j \,|X_t = i \} = \mathrm{P}\{X_1 = j \,|X_0 = i \} = p_{ij}$$

The transition probabilities p_{ij} can be arranged in a square matrix (because X_{t+1} and X_t both take values in the same state space S) form as, P= [p_{ij}], for all i, jϵS

Here, the matrix P indicates the transition probability matrix.

In the transition probability matrix P:

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} \dots \\ p_{10} & p_{11} & p_{12} \dots \\ \vdots & \vdots & \vdots \\ p_{i0} & p_{i1} & p_{i2} \dots \\ \vdots & \vdots & \vdots \end{bmatrix} \quad , \text{i=0,1,2,…}$$

- The rows represent X_t , $\forall$ t .
- The columns represent X_{t+1} , $\forall$ t.

The i-th row of P , for i= 0, 1, 2, 3, . . . under the condition that X_t= i is the probability distribution of the values of x_{t+1}. Such that $p_{ij} = p_{01}$ =P{$X_{t+1} = 1 \mid X_t = 0$ }

P is a finite square matrix if the number of states is finite, where the number of rows equal to the number of states .

$$P = \begin{bmatrix} p_{00} & \cdots & p_{0n} \\ \vdots & \ddots & \vdots \\ p_{n0} & \cdots & p_{nn} \end{bmatrix}$$

It is clear that, the quantities p_{ij} satisfy the conditions:

$$0 \le p_{ij} \le 1$$

$$\sum_j^n p_{ij} = 1$$

In addition, Markov Chains also could have n-step transition defined as the conditional probability that the process will be in state j after n-steps given that it starts in state i at time t.

Definition 4 [16] The n-step transition probabilities are defined as the conditional probability

$$\mathrm{P}\{X_{t+n} = j \mid X_t = i\} = \mathrm{P}\{X_n = j \mid X_0 = i\} = p_{ij}{}^{(n)} \text{ for all } t=0,1,\ldots.$$

Here $p_{ij}{}^{(n)}$ denotes the probability that the process goes from state i to state j in n transitions.

Theorem 2.3 [16] The n-step transition probabilities of a Markov Chain satisfy,

$$p_{ij}{}^{(n)} = \sum_{k=0}^{\infty} P_{ik} P_{kj}{}^{(n-1)}$$

Where we define,

$$P_{ij}{}^{(0)} = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}$$

From the previous theory of matrices, we notice the above relation as the formula for matrix multiplication so that,

$$P^{(n)} = P \times P^{(n-1)}. \text{ n=1,2,3,…}$$

By observing previous formula, we obtain

$$P^{(n)} = P \times P \times \ldots . \times P = P^{n} \text{ … (n factors)}$$

In other words, the n-step transition probabilities $P^{(n)}$, are the entries in the matrix P^n, the n-th power of P.

2.4. Categorizing States of Markov Chains

Definition 5 " A state j is said to be accessible from a state i (denoted by i→j) if ${P_{ij}}^{(n)} > 0$ for some $n \geq 0$ or simply stated, the system can eventually move from state i to state j"[16].

Definition 6 " If state j is accessible from state i and state i is accessible from state j then states i and j are said to communicate with one another (denoted by i↔j) "{16].

The concept of communication is an equivalence relation:

- i ⟷i (reflexivity).
- If i ⟷j , then j ⟷i (symmetry).
- If i ↔ j and j ↔ w, then i ↔ w(transitivity).

If there's two states for example i and j do not communicate, then either

$$P_{ij}{}^{(n)} = 0 \text{ for all } n \geq 1$$

or

$$P_{ji}{}^{(n)} = 0 \text{ for all } n \geq 1$$

or both relations are true.

Since various states can communicate with each other in the same system, the states industrialize classes that are collection of states that only communicate with each other. If there is some way of transforming from state j to state i and there is some way of getting from state i to state j in n steps then states i and j are in the same communication class.

Definition 7 " A state i is recurrent if and only if, after the process starts from state i, the probability of its returning to state i after some finite length of time is one. A non-recurrent state is said to be transient"[16]

Theorem 2.4.1 "A state i is recurrent if and only if $\sum_{n=1}^{\infty} p_{ii}{}^{(n)} = \infty$ equivalently, state i is transient if and only if $\sum_{n=1}^{\infty} p_{ii}{}^{(n)} < \infty$ "[16].

Remark 2.4.1 : In recurrent state $\sum_{n=1}^{\infty} p_{ii}{}^{(n)} = \infty$, if $\lim\limits_{n\to\infty} p_{ii}{}^{(n)} > 0$ then it is positive recurrent [16] .

Remark 2.4.2 : In recurrent state $\sum_{n=1}^{\infty} p_{ii}{}^{(n)} = \infty$, if $\lim\limits_{n\to\infty} p_{ii}{}^{(n)} = 0$ then it is null recurrent [16] .

Note: From the above definitions, transient or recurrent class could apparent when placing Markov Chain states into classes, where each state groping in the same class is either recurrent or transient.

2.5. Irreducible and aperiodic

Definition 9 Irreducible Markov Chain defined for a Markov Chain, which there is only one communication class (all states communicate) [16].

Definition 10 A state j in a Markov Chain called periodic with period d if ${P_{jj}}^{(n)}$ =0 unless n=d,2d,3d,…. If a process can be in state j at times n and n+1, then state j is of the period 1 and it called a periodic [16].

2.6. Properties of Markov Chains in the Lon Run

2.6.1. States Probability Distributions

Consider a discrete Markov Chain.

$$\text{P}\{\ X_{t+1} = j \mid X_0 = i_0\ \ldots\ X_{t-1} = i_{t-1}\,, X = i\}\text{=P}\{X_{t+1} = j\ |X_t = i\}.$$

for all time points t and all states $i_0\,, \ldots\,, i_{t-1}\,, i, j\epsilon S.$

Suppose that we know the initial steady state vector

$$\pi^{(0)} = [\Pr\{X_0 = i\ \}]i\epsilon\ S$$

If we generally define

$$\pi^{(n)} = [\Pr\{X_n = i\ \}]i\epsilon\ S$$

Then we can rewrite the above result in the form of matrix multiplication

$$\pi^{(n)} = \pi^{(n-1)}P = \pi P^{(n)}[4].$$

2.6.2. Steady State Probabilities

The characteristic of a steady state vector of Markov Chain would display after the Markov chain n-step transition probabilities counted, in which the state probabilities do not change with n anymore. This property of Markov Chains states that neglected of the initial state vector of the system , the transition probability from state i to state j settle down to some constant value when the number of transition steps is sufficiently large .

Theorem 2.6.1 [18]

For an Ergodic (i.e., irreducible, aperiodic and positive recurrent) Markov Chain $\lim_{n\to\infty} P_{ij}{}^{(n)}$exists and is independent of the initial state i. That is:

$$\lim_{n\to\infty} p_{ij}{}^{(n)} = \mathrm{T}$$

Where,

$$\pi = (\pi_1, \pi_2, \dots)$$

$$\pi_i > 0 \text{ and } \sum_i \pi_i = 1$$

Where π_i defined the steady state probabilities vector of the Markov Chain. That is $\lim_{n\to\infty} p_{ij}{}^{(n)}$ consists of identical rows, each row is π .

The steady state probability vector term means that the probability of discovering the process in a specific state, say i, after a very big number of transformation tends to the value i, independent of the probability distribution of the initial state vector. Notice that the steady-state probability does not mean that the process settles down into one state.

For a discrete-time Markov Chain which the state space S of it is a finite or countable set, it is enough to be irreducible and aperiodic to be an Ergodic Markov Chain as in this study [7].

2.6.3. State Transition Diagrams

State transition diagram consider as a helpful way to represent the states of Markov Chains when they have a finite number of states. Where each state of a Markov Chain drawn as a node and the conditional probability of transforming from one state to another explained by the connection between the nodes.

For example, the following stochastic process gives two-state Markov Chain for speech activity (on-off source) $\{X_t\}$ for t = 0,1,2,3…

Where:

$$X_t = \begin{cases} 0, & \text{silence (off)} \\ 1, & \text{speech (on)} \end{cases}$$

With

$$P(X_{t+1} = 0 \mid X_t = 0) = 0.6$$

$$P(X_{t+1} = 1 \mid X_t = 1) = 0.7$$

Then the transition probability matrix and state transition diagram for this example is:

$$P=\begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}$$

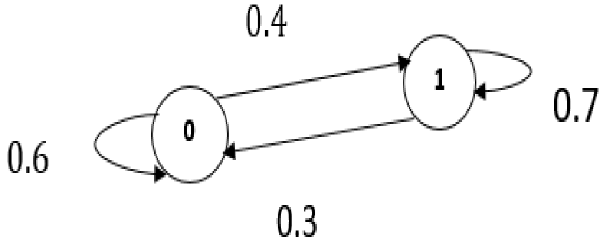

Figure 2.6.3: Transition Digraph.

2.6.4. Expected Return time

The first reaching time n_{ii} $= n_i$ is called the return time to state i starting from the state i. E (n_{ii}) known as the expected return time for the state i.

For an Ergodic Markov Chain the expected return time to state i is E $(n_{ii}) = \frac{1}{\pi_i}$.

Where π_i defined the steady state probabilities vector [4].

2.7. Summary

This chapter presented an introduction about Markov Chain model in detail where Markov chain model could applied for not only stock market, but also it could be extended to other applications. Methods to estimate the parameters discussed in detail so that this model could be applied by anyone who would be interested to reproduce the results presented in the next chapters.

Chapter Three

Methodology of Markov Chain Model

3.1. Data Source

This study based on historical daily opening prices of Paltel company shares collected from the Palestinian Stock Exchange during the period from 2nd January 2019 till 31th December 2019 giving a total of 243 days of prices that used for this study as shown in Figure 3.1.1.

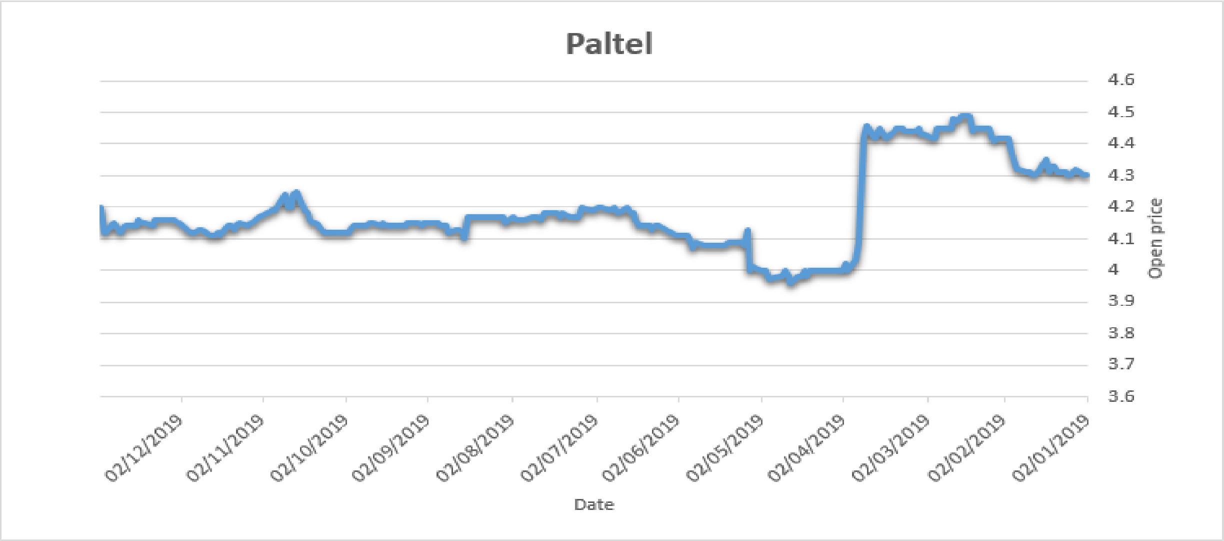

Figure 3.1.1: Plot of stock price (Date, Open price).

3.2. Procedural Overview

Once the opening stock prices found, we applied Markov Chains model to the data set started with examining and reviewing the data, from the data it is obvious that the following day opening price of stock is either remains same, increase or decrease. These three cases observed as the three states of a transition probability matrix of a Markov Chain. If the opening price of t^{th}day is greater than the previous day i.e., $(t-1)^{th}$ day then it labeled as increase (U), if the opening price of t^{th} day is less than $(t-1)^{th}$day then it labeled as down (D) and if the price is same for t^{th} day and $(t-1)^{th}$day then it labeled as remains same (S).

The change in the data from the previous day calculated to convert the data into discrete and finite state so that we could apply Markov Chain.

For example, there is decreasing in 8th January, where the opening price is 4.30, since the opening price in 6th January is 4.32.

The transition matrix of the Markov Chain model involved three states only, the states are the opportunity that a stock index increases (+1) , the index decreases (-1) or the index remains the same (0) .

We stated the three states as follows:

D (corresponding State 3) = open price decreases.

U (corresponding State 2) = open price increases.

S (corresponding State 1) = open price remains the same.

Each share labeled with its corresponding state, the number of transforming for each state to the next state counted. For example, if in 3rd January stock price belonged to state (1) and 6th January price belonged to state (2), then a count put to the transition from n1 to n2 (n12). Every such transition from state counted and observed .Therefore, each entry n_{ij} in the table belong to how many transformation had been from state i to state j.

When all the previous information recorded, and then a one-step transition matrix was ready to build. Each entry of the matrix is supposed to be the probability of the data points transitioning from one state to another, i.e. transforming of stock price index from state increase (D) to state down (U) is denoted by (DU), with the states corresponding to the suitable rows and columns as shown in Table 3.2.1.

Table 3.2.1: Number of transition of Paltel index

State	Same	Up	Down
Same	n_{11}	n_{12}	n_{13}
Up	n_{21}	n_{22}	n_{23}
Down	n_{31}	n_{32}	n_{33}

Where n_{ij} (i, j = 1, 2, 3) performs the number of times transition is move from state i to state j.

$$p_{ij} = \frac{n_{ij}}{\sum_j n_{ij}}$$

Where p_{ij} performs the probability of transforming from state i to state j.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

After the one-step transition matrix prepared, for each state the initial steady state vector probabilities found then we could analyze the long run behavior. This provides where future prices may be.

3.3. Procedure with Data

For better recognizing of the process of applying Markov Chains the behavior of stock index determination, this section explained the procedure with a corresponding data set to better understand our work.

Table 3.3.1: Opening price of Paltel Stock Exchange from January to December 2019

Date	Opening Price	Change from previous day
02/01/2019	4.3	
03/01/2019	4.3	0
06/01/2019	4.32	1
08/01/2019	4.3	-1
09/01/2019	4.3	0
10/01/2019	4.31	1
13/01/2019	4.31	0
14/01/2019	4.33	1
15/01/2019	4.33	0
16/01/2019	4.31	-1
17/01/2019	4.35	1
20/01/2019	4.31	-1
21/01/2019	4.3	-1
22/01/2019	4.3	0
23/01/2019	4.31	1
24/01/2019	4.31	0

27/01/2019	4.32	1
28/01/2019	4.32	0
29/01/2019	4.35	1
30/01/2019	4.39	1
31/01/2019	4.42	1
03/02/2019	4.42	0
04/02/2019	4.42	0
05/02/2019	4.41	-1
06/02/2019	4.42	1
07/02/2019	4.45	1
10/02/2019	4.45	0
11/02/2019	4.45	0
12/02/2019	4.45	0
13/02/2019	4.44	-1
14/02/2019	4.49	1
17/02/2019	4.49	0
18/02/2019	4.48	-1
19/02/2019	4.47	-1
20/02/2019	4.48	1
21/02/2019	4.45	-1
24/02/2019	4.45	0
25/02/2019	4.45	0
26/02/2019	4.45	0
27/02/2019	4.42	-1
28/02/2019	4.42	0

03/03/2019	4.43	1
04/03/2019	4.43	0
05/03/2019	4.45	1
06/03/2019	4.44	-1
07/03/2019	4.44	0
10/03/2019	4.44	0
11/03/2019	4.45	1
12/03/2019	4.45	0
13/03/2019	4.45	0
14/03/2019	4.44	-1
17/03/2019	4.42	-1
18/03/2019	4.43	1
19/03/2019	4.45	1
20/03/2019	4.44	-1
21/03/2019	4.42	-1
24/03/2019	4.46	1
25/03/2019	4.42	-1
27/03/2019	4.09	-1
28/03/2019	4.03	-1
31/03/2019	4	-1
01/04/2019	4.02	1
02/04/2019	4	-1
04/04/2019	4	0
07/04/2019	4	0
08/04/2019	4	0

09/04/2019	4	0
10/04/2019	4	0
11/04/2019	4	0
14/04/2019	4	0
15/04/2019	3.98	-1
16/04/2019	4	1
17/04/2019	3.98	-1
18/04/2019	3.98	0
21/04/2019	3.96	-1
22/04/2019	3.98	1
23/04/2019	4	1
24/04/2019	3.98	-1
25/04/2019	3.98	0
29/04/2019	3.97	-1
30/04/2019	4	1
02/05/2019	4	0
05/05/2019	4.01	1
06/05/2019	4	-1
07/05/2019	4.13	1
08/05/2019	4.08	-1
09/05/2019	4.09	1
12/05/2019	4.09	0
13/05/2019	4.09	0
14/05/2019	4.09	0
15/05/2019	4.08	-1

16/05/2019	4.08	0
19/05/2019	4.08	0
20/05/2019	4.08	0
21/05/2019	4.08	0
22/05/2019	4.08	0
23/05/2019	4.08	0
26/05/2019	4.09	1
27/05/2019	4.07	-1
28/05/2019	4.09	1
29/05/2019	4.11	1
30/05/2019	4.11	0
02/06/2019	4.11	0
09/06/2019	4.14	1
10/06/2019	4.14	0
11/06/2019	4.13	-1
12/06/2019	4.14	1
13/06/2019	4.14	0
16/06/2019	4.14	0
17/06/2019	4.16	1
18/06/2019	4.18	1
19/06/2019	4.18	0
20/06/2019	4.2	1
23/06/2019	4.18	-1
24/06/2019	4.18	0
25/06/2019	4.2	1

26/06/2019	4.19	-1
27/06/2019	4.19	0
30/06/2019	4.2	1
01/07/2019	4.2	0
02/07/2019	4.19	-1
03/07/2019	4.19	0
04/07/2019	4.19	0
07/07/2019	4.2	1
08/07/2019	4.17	-1
09/07/2019	4.17	0
10/07/2019	4.17	0
11/07/2019	4.17	0
14/07/2019	4.18	1
15/07/2019	4.17	-1
16/07/2019	4.18	1
17/07/2019	4.18	0
18/07/2019	4.18	0
21/07/2019	4.18	0
22/07/2019	4.16	-1
23/07/2019	4.17	1
24/07/2019	4.17	0
25/07/2019	4.17	0
28/07/2019	4.16	-1
29/07/2019	4.16	0
30/07/2019	4.16	0

31/07/2019	4.16	0
01/08/2019	4.17	1
04/08/2019	4.15	-1
05/08/2019	4.17	1
06/08/2019	4.17	0
07/08/2019	4.17	0
08/08/2019	4.17	0
18/08/2019	4.17	0
19/08/2019	4.1	-1
20/08/2019	4.12	1
21/08/2019	4.13	1
22/08/2019	4.13	0
25/08/2019	4.12	-1
26/08/2019	4.14	1
27/08/2019	4.14	0
28/08/2019	4.14	0
29/08/2019	4.15	1
01/09/2019	4.15	0
02/09/2019	4.15	0
03/09/2019	4.15	0
04/09/2019	4.14	-1
05/09/2019	4.15	1
08/09/2019	4.15	0
09/09/2019	4.15	0
10/09/2019	4.14	-1

11/09/2019	4.14	0
12/09/2019	4.14	0
15/09/2019	4.14	0
16/09/2019	4.14	0
17/09/2019	4.14	0
18/09/2019	4.15	1
19/09/2019	4.14	-1
22/09/2019	4.15	0
23/09/2019	4.15	0
24/09/2019	4.14	-1
25/09/2019	4.14	0
26/09/2019	4.14	0
29/09/2019	4.14	0
30/09/2019	4.13	-1
01/10/2019	4.12	-1
02/10/2019	4.12	0
03/10/2019	4.12	0
06/10/2019	4.12	0
07/10/2019	4.12	0
08/10/2019	4.12	0
09/10/2019	4.12	0
10/10/2019	4.12	0
13/10/2019	4.15	1
14/10/2019	4.15	0
15/10/2019	4.16	1

16/10/2019	4.18	1
17/10/2019	4.19	1
20/10/2019	4.25	1
21/10/2019	4.24	-1
22/10/2019	4.2	-1
23/10/2019	4.2	0
24/10/2019	4.24	1
27/10/2019	4.2	-1
28/10/2019	4.19	-1
29/10/2019	4.19	0
30/10/2019	4.18	-1
31/10/2019	4.18	-1
03/11/2019	4.17	-1
04/11/2019	4.16	-1
05/11/2019	4.15	0
06/11/2019	4.15	0
07/11/2019	4.14	-1
10/11/2019	4.15	1
11/11/2019	4.14	-1
12/11/2019	4.13	-1
13/11/2019	4.14	1
14/11/2019	4.14	0
17/11/2019	4.11	-1
18/11/2019	4.12	1
19/11/2019	4.11	-1

20/11/2019	4.11	0
21/11/2019	4.11	0
24/11/2019	4.13	1
25/11/2019	4.13	0
26/11/2019	4.12	-1
27/11/2019	4.12	0
28/11/2019	4.12	0
01/12/2019	4.14	1
02/12/2019	4.15	1
03/12/2019	4.15	0
04/12/2019	4.16	1
05/12/2019	4.16	0
08/12/2019	4.16	0
09/12/2019	4.16	0
11/12/2019	4.16	0
12/12/2019	4.14	-1
15/12/2019	4.15	1
16/12/2019	4.15	0
17/12/2019	4.16	1
18/12/2019	4.14	-1
19/12/2019	4.14	0
22/12/2019	4.14	0
23/12/2019	4.13	-1
24/12/2019	4.12	-1
26/12/2019	4.15	1

29/12/2019	4.12	-1
30/12/2019	4.12	0
31/12/2019	4.2	1

Once the state labeled, the transition of the data from one state to the next state calculated by using Microsoft excel as shown in table 3.3.2. The Paltel index of 243 trading days shows that it was unchanged 112 days, increase 66 days and decrease 65 days as shown in Table 3.3.3. The Paltel index recorded in the first trading day was in unchanged state and there is no information about the transition state of Paltel index in the previous day, due to this reason the total numbers of unchanged states taken as 111 and had a total of 242 trading day.

Table 3.3.2: Sample of our historical Opening price.

Date	Opening Price	Change from previous day	Change from previous day	Transforming for each state to the next state
02/01/2019	4.3			
03/01/2019	4.3	0	1	
06/01/2019	4.32	1	2	n12
08/01/2019	4.3	-1	3	n23
09/01/2019	4.3	0	1	n31
10/01/2019	4.31	1	2	n12
13/01/2019	4.31	0	1	n21
14/01/2019	4.33	1	2	n12

Table 3.3.3: Number of transition of Paltel index

	Paltel index remains same (S)	Increase in Paltel index (U)	Decrease in Paltel index (D)
Paltel index remains same (S)	56	30	25
Increase in Paltel index (U)	29	13	24
Decrease in Paltel index (D)	24	26	15

1) The transition probability matrix of Paltel index using the above information can be constructed as:

$$P_{paltel} = \begin{bmatrix} \frac{56}{111} & \frac{30}{111} & \frac{25}{111} \\ \frac{29}{66} & \frac{13}{66} & \frac{24}{66} \\ \frac{24}{65} & \frac{26}{65} & \frac{15}{65} \end{bmatrix}$$

$$P_{paltel} = \begin{bmatrix} 0.5045 & 0.2703 & 0.2252 \\ 0.4394 & 0.1970 & 0.3636 \\ 0.3692 & 0.4 & 0.2308 \end{bmatrix}$$

As shown the transition probability matrix of Paltel index is irreducible and aperiodic .For a discrete-time Markov Chain it is enough to be irreducible and aperiodic to be an Ergodic Markov Chain.

2) The transition diagraph of the transition probabilities of Paltel index is shown below:

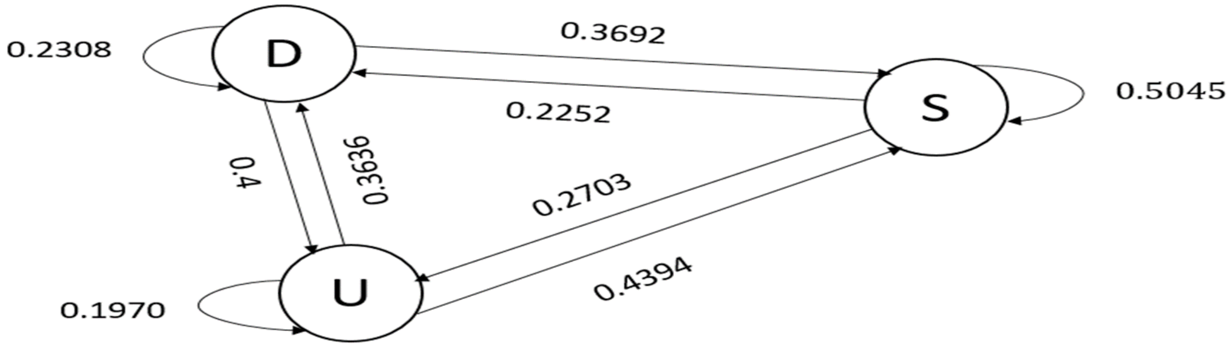

Figure 3.3.1: Transition Digraph of Paltel index

3) Determination of the initial state vector:

The Paltel index during the study period represents three various states "increase" U, "remains same" and "decrease" D. The probability of occurrence of these three different states obtained from the initial state vector.

The initial state vector denoted by $\pi_i$and given by:

$$\pi_i = (\pi_1 \quad \pi_2 \quad \pi_3)$$

where π_1 , $\pi_2 \quad and \ \pi_3$ provide the probability that Paltel index remains same, increase and decrease respectively .So the initial state vector for Paltel share is:

$$\pi_1 = \frac{112}{243} = 0.4609$$

$$\pi_2 = \frac{66}{243} = 0.2716$$

$$\pi_3 = \frac{65}{243} = 0.2675$$

4) Calculating state probabilities for forecasting the next day open price:

By implementation the initial state vector and transition probability matrix, it is easy and possible to discover the state probabilities of various opening day in future.

$$\pi^{(n)} = \pi^{(n-1)}P$$

The state probabilities of opening price of index for Paltel for 244th day will be:

$$\pi^{(1)} = \pi^{(0)}P = [0.4609\ \ 0.2716\ \ 0.2675] \times \begin{bmatrix} 0.5045 & 0.2703 & 0.2252 \\ 0.4394 & 0.1970 & 0.3636 \\ 0.3692 & 0.4 & 0.2308 \end{bmatrix}$$

= [0.4506 0.2851 0.2643]

Which indicates:

- The probability that the opening stock price in the 244th day will remain same is 0.4506.
- The probability that the opening stock price in the 244th day will increase is 0.2851.

- The probability that the opening stock price in the 244th day will reduce is 0.2643.

The state probabilities of opening price of index for Paltel for 245th day will be:

$$\pi^{(2)} = \pi^{(1)}P = [0.4506\ 0.2851\ 0.2643] \times \begin{bmatrix} 0.5045 & 0.2703 & 0.2252 \\ 0.4394 & 0.1970 & 0.3636 \\ 0.3692 & 0.4 & 0.2308 \end{bmatrix}$$

= [0.4502 0.2837 0.2661]

Which indicates:

- The probability that the opening stock price in the 245th day will remain same is 0.4502.
- The probability that the opening stock price in the 245th day will increase is 0.2837.
- The probability that the opening stock price in the 245th day will reduce is 0.2661.

5) Long Run Behavior of Paltel Index:

The forecasting of long run behavior of Paltel index is very meaningful for investors. The long run behavior of Paltel index observed by determining the higher order transition probability matrix of Paltel index by using Matlab as given below:

$$p^{(2)}{}_{paltel}= \begin{bmatrix} 0.4564 & 0.2797 & 0.2639 \\ 0.4525 & 0.3030 & 0.2545 \\ 0.4472 & 0.2709 & 0.2819 \end{bmatrix}$$

$$p^{(3)}{}_{paltel}= \begin{bmatrix} 0.4506 & 0.2840 & 0.2654 \\ 0.4503 & 0.2811 & 0.2686 \\ 0.4487 & 0.2870 & 0.2643 \end{bmatrix}$$

$$p^{(4)}{}_{paltel}= \begin{bmatrix} 0.4501 & 0.2839 & 0.2660 \\ 0.4599 & 0.2845 & 0.2656 \\ 0.4501 & 0.2835 & 0.2664 \end{bmatrix}$$

$$p^{(5)}{}_{paltel}= \begin{bmatrix} 0.4500 & 0.2840 & 0.2660 \\ 0.4500 & 0.2839 & 0.2661 \\ 0.4500 & 0.2840 & 0.2659 \end{bmatrix}$$

$$p^{(6)}{}_{paltel}= \begin{bmatrix} 0.4500 & 0.2840 & 0.2660 \\ 0.4500 & 0.2840 & 0.2660 \\ 0.4500 & 0.2840 & 0.2660 \end{bmatrix}$$

The higher order transition probability matrix of Paltel index computed above shows that after the 6th trading days, the transition probability matrix tends to the steady state. After that, the transition probability matrix remains same for the consecutive trading days subsequent. This steady state transition probability matrix of Paltel index reveals the following information:

- The probability that in near future the Paltel index remain same regardless of its initial states increase, remains same or decrease is 0.4500.
- The probability that in near future the Paltel index increase regardless of its initial states increase, remains same or decrease is 0.2840.
- The probability that in near future the Paltel index decrease irrespective of its initial states increase, remains same or decrease is 0.2660.

6) Expected Return Time

Another side of stock price behavior is the expected return time to a specific state . For Paltel index:

- Expected return time to the state (1) S, beginning from same state is

E (n_{11}) = 1/0.4500=2.2

- Expected return time to the state (2) U, beginning from same state is

E (n_{22}) = 1/0.2840=3.521

- Expected return time to the state (3) D, beginning from same state is

E (n_{33}) = 1/0.2660=3.759

So it determined from the previous calculation that the Markov Chain should visit the state (1) "Same" in two days, state (2) "Up" on average in three days and the state (3) "Down" on average in four days.

3.4.1.Summary and Result

The Markov Chain model helpful to analyze the behavior of stock index of Paltel. Since Markov Chain model predict and forecast the behavior of stock index in a potential method. The transition probability matrix and initial state vectors express a description of the following day stock price for Paltel stock index and the n-step transition probability matric determine the long-term behavior of stock index price.

Steady state probability express that in future the Paltel stock price increases with probability 0.2840, with probability 0.2660 decrease and the probabilities for stock price index remain same showed as 0.4500. In addition, the Markov Chain should visit state (D) , on average, in four days, state (U)) , on average , in three days and the state remain same in two days. These may help the future investors and shareholders for the company. Since results for share price index remain same, up and down performed in probabilities that have some significance economic.

Chapter Four

Preprocessing Data and Modeling

4.1. Introduction about Big Data and Data Mining

"The term Big Data refers to the evolution and use of technologies that provide the right user at the right time with the right information from a mass of data that has been growing exponentially for a long time in our society". The main goal that represent a challenge for the researchers is to deal with the increasingly volumes of data and the difficulty of dealing with rapidly complex and interconnected data [14].

Big Data has three-dimensions which are huge datasets (volume), various structured, semi-structured, and unstructured (variety) data, and arriving faster (velocity) than before. These are the 3V [14]. As shown in Figure 4.1.1

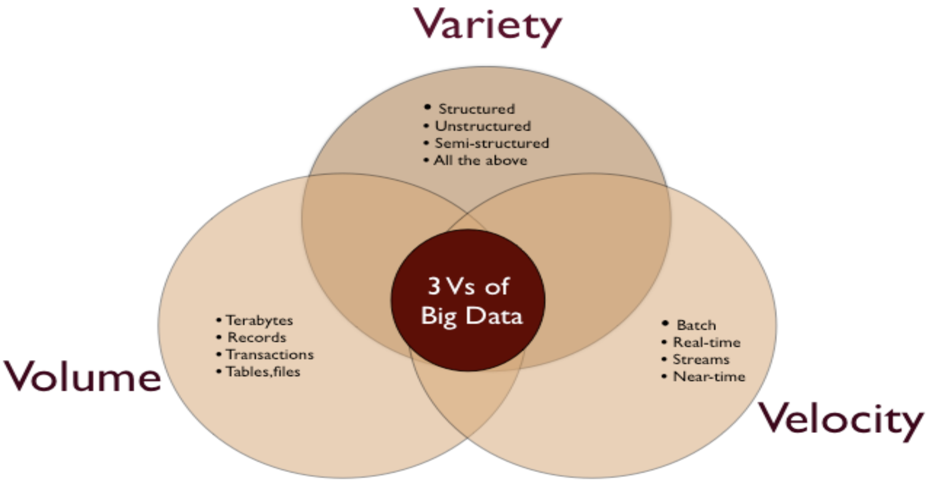

Figure 4.1.1: The 3Vs of Big Data

Commonly, Big Data express to a gathering of large volumes of data, with the help of Data Mining these data extract some helpful information [17]. "Data mining is the process of analyzing data from different perspectives and summarizing it into useful information, information that can be used to increase revenue, cuts costs, or both" [13]. The relationship of Data Mining as shown in Figure 4.1.2 with Big Data is that Big Data gives many relationships whereas Data Mining gives lots of information [17].

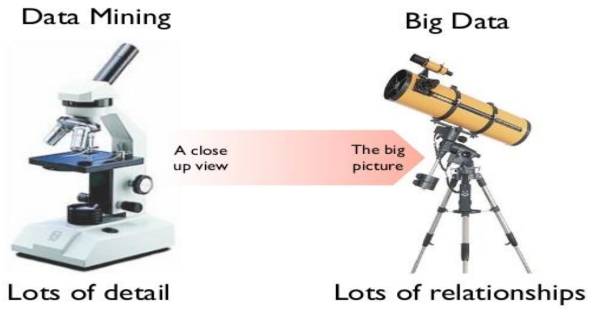

Figure 4.1.2: Big Data with Data Mining

4.2. Preprocessing the Data

Preprocessing the Data describe preparing data in approperte and helpable format, so that extraction process of information can be applied [10].

By applying the data preprocessing techniques as show in Figure 4.2.1 its possible to develop the quality of the data, by improving the efficiency and accuracy of the subsequent mining process [1].

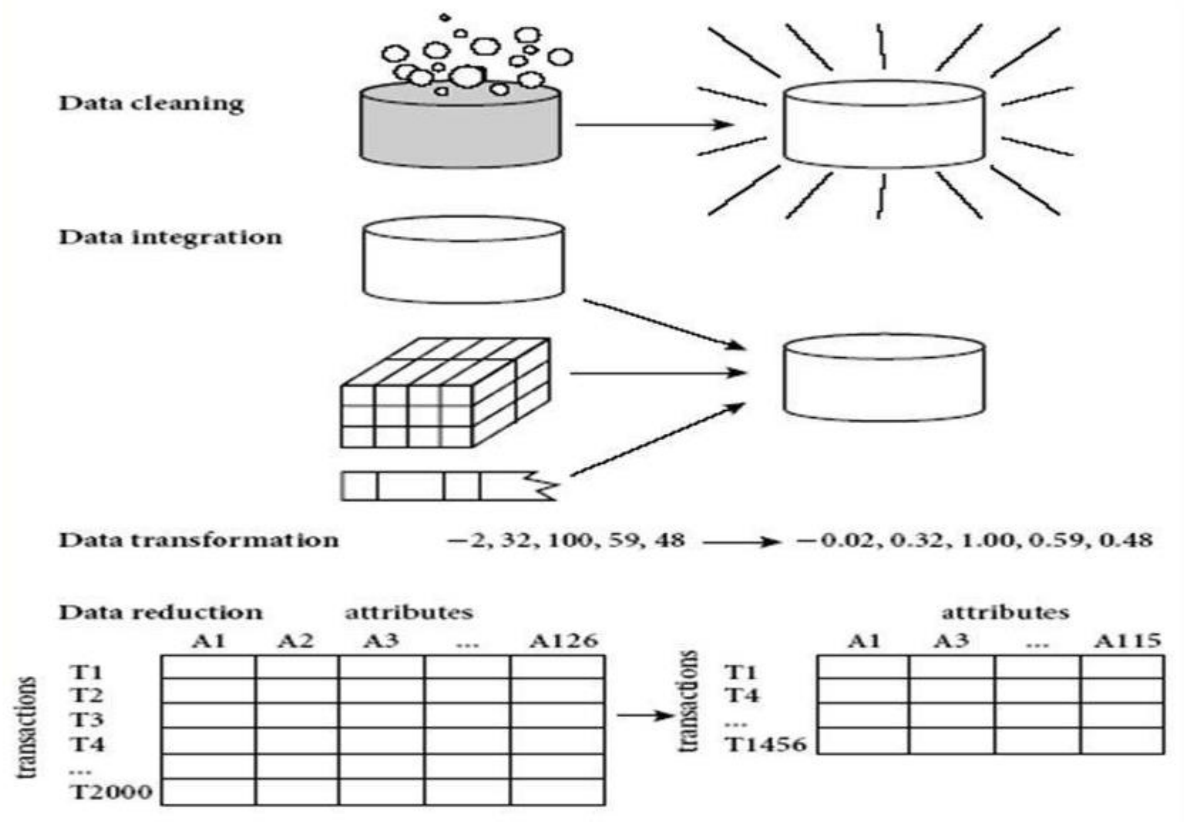

Figure 4.2.1: Forms of data preprocessing.

The techniques summarized as follows [1, 10]:

- **Data cleaning**: taking away outliers, discovering missing values, smoothing noisy data and solving inconsistencies.
- **Data reduction**: dealing with the huge volume of data to reduce the amount of data by taking away repeated observations and applying instance selection .

For example, the attribute male or female and student could represent as male student or female student. This can be helpful if the study area doesn’t interest us and the research talk about how many men and, or women are students .

- **Data transformation**: transforming text and graphical data to a format that dealing, normalization or scaling the data. In addition to aggregation, generalization the data. For example, study include smog around the global world , we have data about wind speeds such that the data mixed so we have three figures: meter/s, miles /s, and km/ h. We have to transform these three various data to the same scale .
- **Data integration**: correcting various in coding schemes due to the combining of different sources of data. For example, the persona called by his last name say (Mansour) also in the database he could be recognize by the first name as Ali and A. in the third.

4.3. Data Mining

"Data Mining is a technique for discovering interesting patterns as well as descriptive, understandable models from large data" [17].

Data Mining process have many steps [13] as shown in Figure 4.3.1 include:

- Data Cleaning.
- Data Integration.
- Data Selection.
- Data Transformation.
- Data Mining.
- Pattern Evaluation.
- Knowledge Presentation.

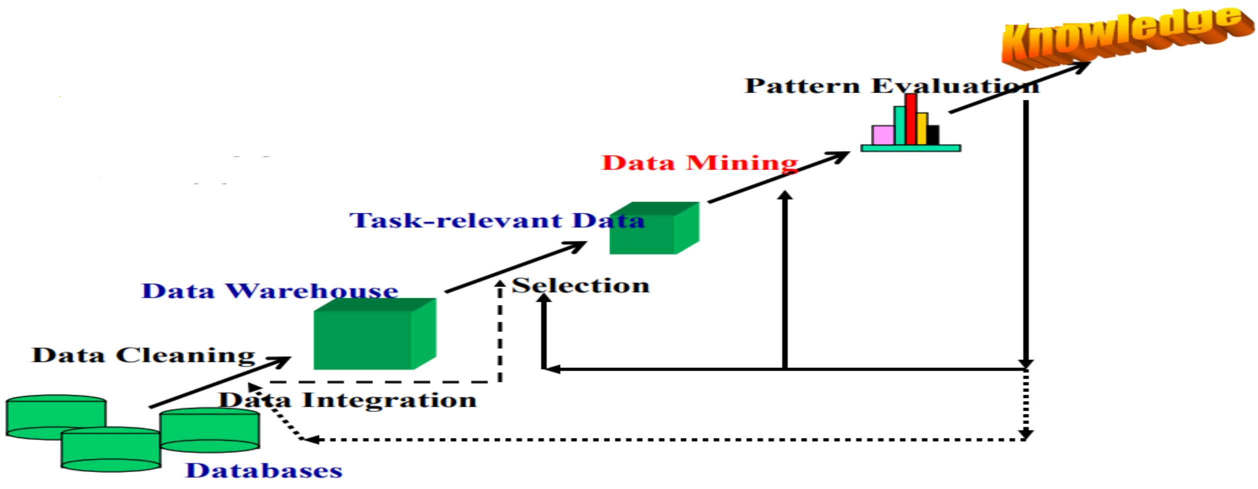

Figure 4.3.1: Data Mining process.

Whereas Data Mining is a collection process of discovering patterns in large data, which include ways at the intersection of artificial intelligence, machine learning, statistics, and database systems [7].

Having data from many heterogeneous places that support analytical reporting and decision making, data cleaning, data integration and data consolidations are required, which could be done by Data Warehouse.

4.4. Data Warehouse

"A data warehouse is a collection of data that supports decision-making processes "[8]. The features of a data warehouse can represent as [13]:

- **Subject Oriented**: it gives information for subject moreover of the given operations.

- **Integrated**: integrated data made for data warehouse from heterogeneous data sources. Which is useful in dealing with the data.

- **Time Variant**: The data collected for data warehouse forming with a specific period. The data in data warehouse gives information from the historical dataset.

- **Non-volatile**: means when new data added to it the previous data there's no deleted or changed .

4.5. Classification and Prediction Models

According to the Data Mining approaches, the techniques of Data Mining are varying. There are a huge number of useful techniques for mining and other data process where these techniques include Association, Clustering, Regression, and Classification [9]. As shown in Figure 4.5.1

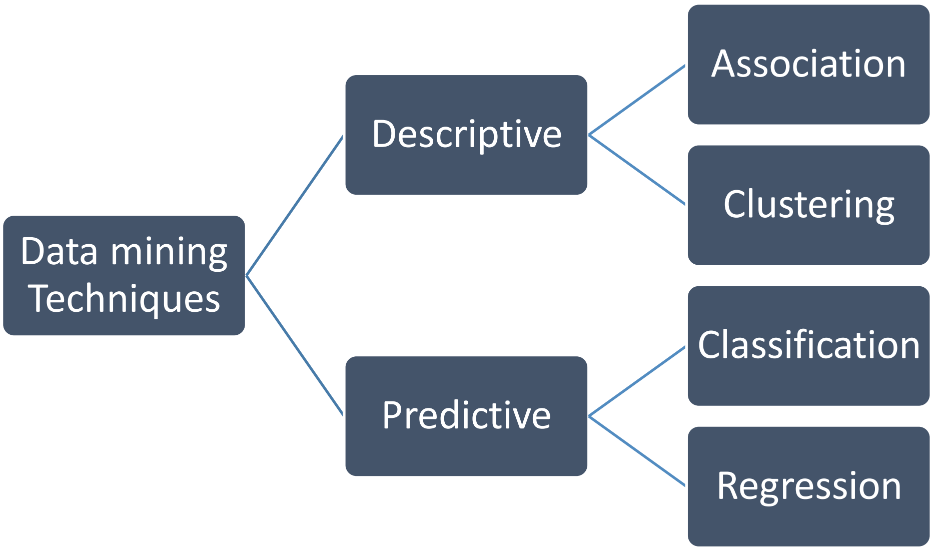

Figure 4.5.1: Data Mining techniques

The predictive analysis known as the part of data analysis to know unknown values of prediction target variable [9].

"Classification is predicting a categorical (discrete) variable and the organization of data into categories could be easy to use and more efficient"[10].

"It aims to accelerate getting data and retrieve it as well as predict a particular effect based on given information "[10].One of the popular classifier used for classification algorithms is Decision Trees.

Where Decision Tree based on the values of input variables express the data into finite number of classes .It is most suitable for categorical data [10].

4.6. Decision Tree Induction

"Decision Tree (DT) induction is the learning of Decision Trees, which is a simple flowchart that selects target variables (class labels) of a dataset using the values of one or more input variables (attribute)" [10]. In Decision Tree the process begin at the root node of the Decision tree and then repeated the progresses until it reaches the leaf node with same class labels.

Homogenous subsets record with the same class label but it is very hard to achieve pure homogenous subsets while dealing with real data. Since there always will be some mixing .Therefore, while building the decision tree, the aim at each node is to select split conditions that best divide the dataset into homogenous subsets [15].

Important concepts related to Decision Trees [22]:

1. **Root Node:** it express all population or sample where more dividing gets into two or more homogeneous sets.

2. **Internal Node:** it called the internal node when a subsection node splits into more subsection nodes.

3. **Leaf (Terminal Node):** it is a node that does not split anymore called a Leaf or a Terminal node.

4. **Branch:** A subsection of the entire tree called branch .

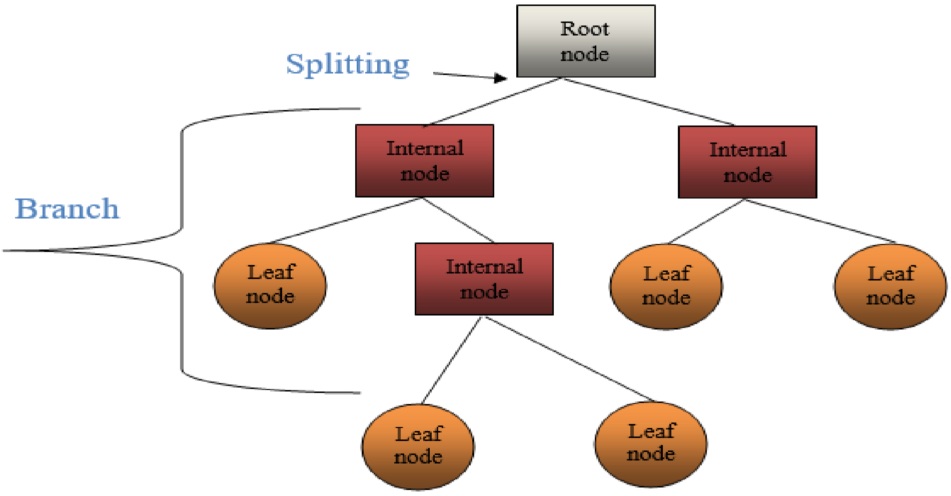

Figure 4.6.1: Decision Tree structure.

4.7. Algorithms

Different algorithms are using in Decision Tree Classifier to make the decision of splitting a node into two or more subsection nodes. However the algorithm selection is depend on the type of target variables. Some of these algorithms that used in Decision Trees:

- **ID3** a decision tree algorithm improved by J. Ross Quinlan → (Iterative Dichotomiser) [10].
- **C4.5** a decision tree algorithm represent later by B. Hunt, J. Marin, and P. T. Stone. Quinlan → (successor of ID3) [10].
- **CART** a decision tree algorithm improved in 1984 by L. Breiman, J. Friedman, R. Olshen, and C. Stone → (Classification and Regression Trees) [10].

Note: Classification and Regression Trees (CART) is the most common technique in the statistical area. In the areas of statistics, CART use Decision Tree to gain acceptance to make binary split on inputs to get the purpose that we put [9].

4.8. Decision Trees Construction

The process for tree algorithm is [19]:

In the beginning, we represent the entire population or sample in a root node by choosing the best attribute using attribute selection measures to split the dataset.

Then make that attribute a decision node and split the dataset into smaller subsection node known internal node.

By repeating this process frequently .Starts tree building for each child until one of the condition happen:

All the nodes belong to the same attribute value.

No more remaining attributes.

No more instances.

4.9. Attribute Selection Measures

Many indices proposed to measure the impurity value of a split condition included Information gain , Gini index, and Gain ratio. In this thesis, we applied Gini index.

- **Information Gain [10]:**

"ID3 uses information gain as its attribute selection measure, which measures the impurity of the input set". The attribute with the highest information gain is chosen as the splitting attribute

$$Info(D) = -\sum_{i=1}^{m} p_{i\,\log_2} p_i$$

Where, m is the distinct classes and $p_i$is the probability that an arbitrary in D (dataset) belongs to class Ci .

Then the expected information requirement for each attribute :

$$Info_A(D) = \sum_{j=1}^{V} \frac{|D_j|}{|D|} \times Info(D_j)$$

Where,

- D represent dataset .
- v is the number of discrete values in attribute A.
- $\frac{|D_j|}{D}$ acts as the weight of the j-th partition.
- $Info_A(D)$ is the expected information required to classify a tuple from D based on the partitioning by A.

Information gain is defined as the difference between the original information requirement and the new requirement. That is,

$$Gain(A) = Info(D) - Info_A(D)$$

- **Gain Ratio[10]:**

"Information gain biased for the attribute with many outcomes. It means it prefers the attribute with a large number of distinct values". C4.5, a successor of ID3, uses an extension to information gain known as gain ratio it applies a kind of normalization to information gain using a “split information” value defined analogously with Info(D) as

$$SplitInfo_A(D) = -\sum_{j=1}^{V} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

Where,

- $\frac{|D_j|}{D}$ represent the weight of the j-th partition.
- v is the number of discrete values in attribute A.

The gain ratio defined as:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

The attribute with the highest gain ratio choose as the splitting attribute.

- **<u>Gini index [10]:</u>**

"The Gini index used in CART, determines the purity of a specific class after splitting along a particular attribute. The best split increases the purity of the sets resulting from the split". If D is a dataset with m different class labels, Gini defined as:

$$Gini(D) \ = 1 - \sum_{i=1}^{m} p_i^2$$

Where p_i is frequency of class i in D. If the Dataset split on attribute A into two subsection D1 and D2 have sizes N1 and N2 (N=sample size), GINI computes as:

$$Gini_A(\mathrm{D}) = \frac{N_1}{N}\mathrm{gini}\ (D_1) + \ \frac{N_2}{N}\,\mathrm{gini}(D_2)$$

Reduction in impurity calculated as:

$$\Delta Gini(A) = Gini(D) - Gini_A(\mathrm{D})$$

"There is no significant difference in the performance of models using GINI index and Information gain "[15].

4.10. Summary

Many researcher beside company's and governments have used various algorithms of Data Mining to predict and analyze trends with particular aims. In this study, we have presented a summary of data mining development, techniques, Predictive analysis and focus on the algorithm of the decision tree in addition to analyse the behavior of the Palestinian Stock Market Prices.

Chapter Five

Methodology of Decision Tree

5.1. Introduction

As recognized previously the objective of this thesis is to forecast and analyse the behavior of the Palestinian Stock Market Prices:case stdudy (Paltel) by applying two models which are:

1. Decission Tree Model.

2. Markov model

5.2. Dataset Description

Palestine Stock Exchange contains the historical prices of many companies listed in the exchange. As the volume of such data is very large and not easy to deal with , the decision took to choose one company listed in the exchange. The selection of this company was at random during our collection time period ,the daily collected data contained 2 attributes: 1) volume. 2) no. of deals.

The classification goal is to forecast and analyse the behavior of the Palestinian Stock Market Prices:case stdudy(Paltel).The data on share prices of the study were collected from the daily list published by the Palestinian Stock Exchange during the period from 2nd January 2019 till 31th December 2019 which gives a total of 243 days of values that were used in this study as shown in Figure 5.2.1 & Figure 5.2.2 . Details of the Dataset are given in Table 5.2.1

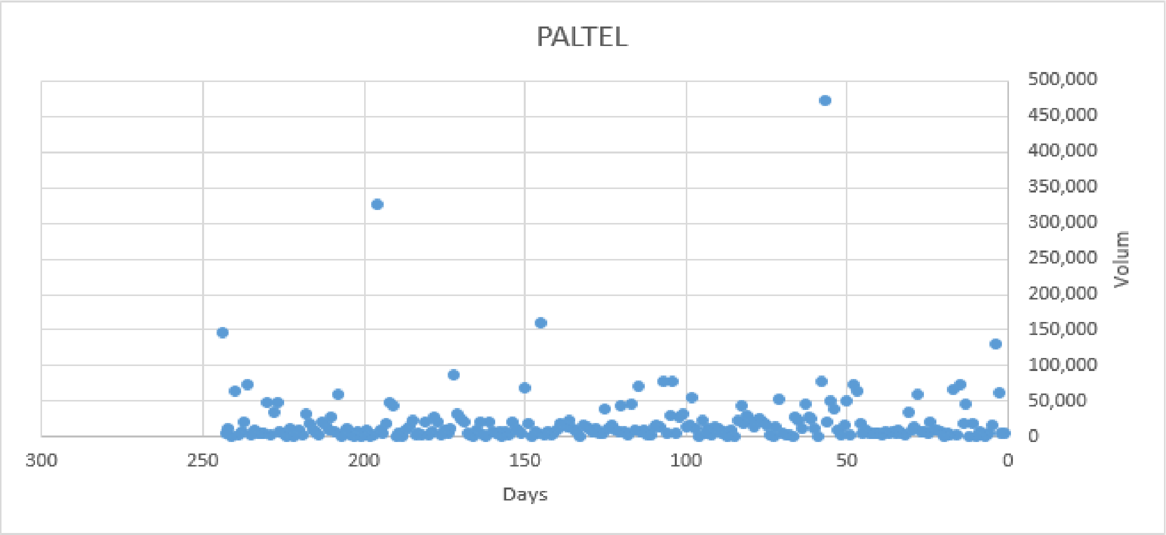

Figure 5.2.1: Plot of volume of Paltel index.

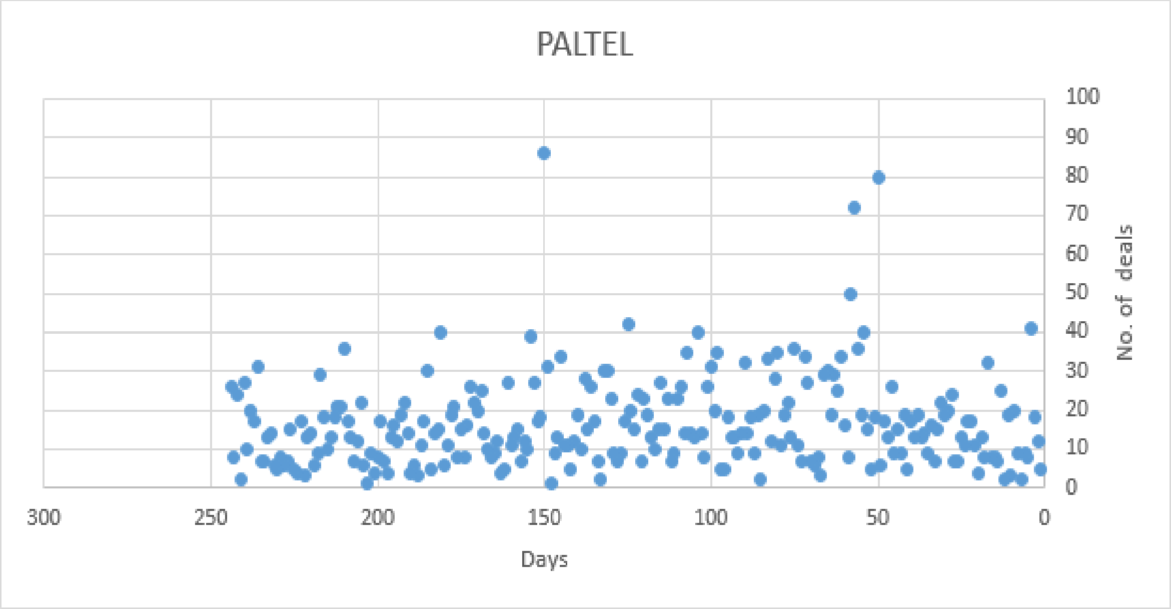

Figure 5.2.2: Plot of No. of deals of Paltel index.

The target variable in Decision Tree Classifiers is the" behavior" of the Stock Prices , which might:

- ○ remain the same (unchanged) = same
- ○ decrease = down
- ○ increase = up
- ○

Whereas, we use two attribute:

- Volume

Defined all number of stocks that sold and bought during time period. Where it give some information beside the price of the stock. Many scenarios can happen when prices are moving and volume can give you an idea about the next move of market .

Here are some of those scenarios:

- Prices rise on heavy volume
- Prices fall on heavy volume
- Prices rise on average volume
- Prices fall on average volume
- Prices rise in low volume
- Prices fall on low volume
- Prices remain unchanged on high volume
- Prices remain unchanged on low volume

Every scenarios consider if you are aiming to analyze the stock market.[20].

- No. of deals

The number of deals shows the intensity of a price move. If a price move is associated with a large number of deals then the move is strong.

If the move is associated with a small number of trades then the price move is weak and may not hold.

Low levels of trades are characteristic of the indecisive expectations that are present in consolidation periods and market bottoms. A higher number of trades will occur at market tops when there is a strong consensus that prices will continue to rise

. It is also very common for the number of trades to rise at the beginning of new trends [21].

Table 5.2.1: Discerption of the Dataset

Variable	Descriptive	Possible value
Volume	All number of stocks that sold and bought during time period	average-level, heavy-level, high-level, low-level
No. of deals	It is the number of buying and selling operations that take place on the stock exchange, which is announced and are not limited to a quantity or a minimum, so anyone can buy or sell any amount of shares he pleases	high-level, low-level
Behavior	Behavior of share	Same ,up, down

5.3. Preparing the data

When the data had been collected, all the values of the attributes selected were numeric values. So firstly Data transformation had been applied by transforming data so as all the values became discrete. The transforming of the numeric value of each attribute to discrete values depended on the highest and lowest value of each attributes. After calculating the highest and lowest value, four possible level for volume have been formed where as two value for the number of deals.

Depending on who we setting the values of both attributes previously the numeric values of the volume attributes were replaced by average-level, heavy-level, high-level, low-level and by high-level, low-level instead of the numeric values of the deals attributes were replaced.

Three cases have been taken as the three discrete value of the behavior of stock index. If the opening price of t^{th}day is greater than the previous day i.e., $(t-1)^{th}$ day then it is described as up. If the opening price of t^{th} day is less than $(t-1)^{th}$day then it is taken as down and if the price is same fort^{th} day and $(t-1)^{th}$day then it is taken as same. Table 5.3.1shows sample after choosing 2 attributes and after transforming them to discrete values.

Table 5.3.1: Sample of our historical data after selecting attributes and after transforming.

volume	deals	Behavior
low-level	low_level	Same
low-level	low_level	Up
high-level	low_level	down
low-level	low_level	same
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	down

Class label
(target
variable)

Attribute (variable)

5.4. Decision Tree Classifier Building in Python-language

Now the following step is to built the classification model by the decision tree method. The decision tree is a very useful way since it is relatively fast.

The gini index used to split attributes and to create the Decision Tree where according to the gini index each attribute have been located . The attribute that has the lowest gini index was the volume (low-level). This attribute is considered as the root node of the decision tree. We have build the complete decision tree by repeated the process for the remaining attributes to create the next level of the tree.

✚ Data Description:

behavior = The target variable,

volume values: average-level, heavy-level, high-level, low-level

deals values: high_level, low_level

behavior values: down, same ,up

Step 1: Load Python Packages

```
    import pandas as pd # pandas is a Python library that provides fast, flexible, and
expressive data structures. the two primary
# data structures of pandas, Series (one-dimensional) and DataFrame (two-
dimensional).pandas is well suited for inserting
# and deleting columns from DataFrame, for easy handling of missing data
import pip
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.metrics import accuracy_score
import xlrd # library to read data from execl file, if you don't import you will get the
following error:
# ImportError: Missing optional dependency 'xlrd'. Install xlrd >= 1.0.0 for Excel
support Use pip or conda to install xlrd.
# Load Libraries to Visualize Decision Tree
from IPython.display import Image
from sklearn import tree
import pydotplus
```

Step 2: Pre-Process the data

```
    candidates = pd.read_excel (r'D:\stock1.xlsx')
df = pd.DataFrame(candidates,columns= ['volume', 'deals','behaviour'])
print(df)
```

Step 3: Subset The Data: (our new dataset should only have the variables that we will be using to build the model)

```
X = df[['volume', 'deals']]

y = df['behavior']

print(' ')

# Convert categorical variable into dummy/indicator variables or (binary vairbles)
essentialy 1's and 0's

#one_hot_data = pd.get_dummies(df[['volume','deals']])

one_hot_data = pd.get_dummies(X)
```

Step 4: Build a Decision Tree Classifier

```
     # the default criterion in DecisionTreeClassifier is "gini", and we can use another
criterion as in DecisionTreeClassifier(criterion=’gini’,....)

     clf = DecisionTreeClassifier()

# Train Decision Tree Classifer

clf_train =clf.fit(one_hot_data, df['behavior'])

#print(clf_train)
```

Step 5: Predictions for new data using Decision Tree

```
     # important note: after converting the categorial data into binary using dummy
function the data structure is:

# volume_average-level, volume_heavy-level, volume_high-level, volume_low-level,
deals_high_level, deals_low_level
```

```
# Note: get_dummies fun in pycharm sorts the new (released) attributes by alphabetical
order.
# in dummy  fun: each attribute is converted into number of attributes based on the
number of groups of values.
print(' ')
# predict if it will paly or not for the following case:
# volume=low-level,deals=low_level

prediction1 = clf.predict([[0,0,0,1,0,1]])   # hhhh
print ('Predicted Result1 using Decision Tree Classifier: ', prediction1)
print(' ')
print(list(one_hot_data.columns.values))
print(' ')
print(' ')
```

Step 6: the draw Decision tree classifier

```
     # Visualize Decision Tree: first : Create DOT data
#feature_names: should be after dummy fun is used.
# Export/Print a decision tree in DOT format.    DOT format: format as rules if
then … try to print next command
 dot data = tree.export_graphviz(clf_train, out_file=None,
feature_names=list(one_hot_data.columns.values),
class_names=['down', 'same','up'], rounded=True, filled=True)
#Gini decides which attribute/feature should be placed at the root node, which features
will act as internal nodes or leaf nodes

#Create Graph from DOT data
graph = pydotplus.graph_from_dot_data(dot_data)
```

```
# Show graph

Image(graph.create_png())

# if dot_data in ( graph = pydotplus.graph_from_dot_data(dot_data) )   is dummied the

function: graph.write_png()  will be released using GVEdit framework

graph.write_png("StocksTree.png")
```

Output

```
volume        deals     behavior

0 low-level  low_level      same

1 low-level  low_level        up

2 high-level  low_level      down

3 low-level  low_level      same

4 low-level  low_level        up

        ..        ...       ...      ...

238 low-level  low_level      down

239 low-level  low_level        up

240 low-level  low_level      down

241low-level  low_level      same
```

```
242high-level  low_level        up

[243 rows x 3 columns]

Predicted Result1 using Decision Tree Classifier:  ['same']

['volume_average-level', 'volume_heavy-level', 'volume_high-level', 'volume_low-level',
'deals_high_level', 'deals_low_level']

Process finished with exit code.
```

Decision Tree Graph:

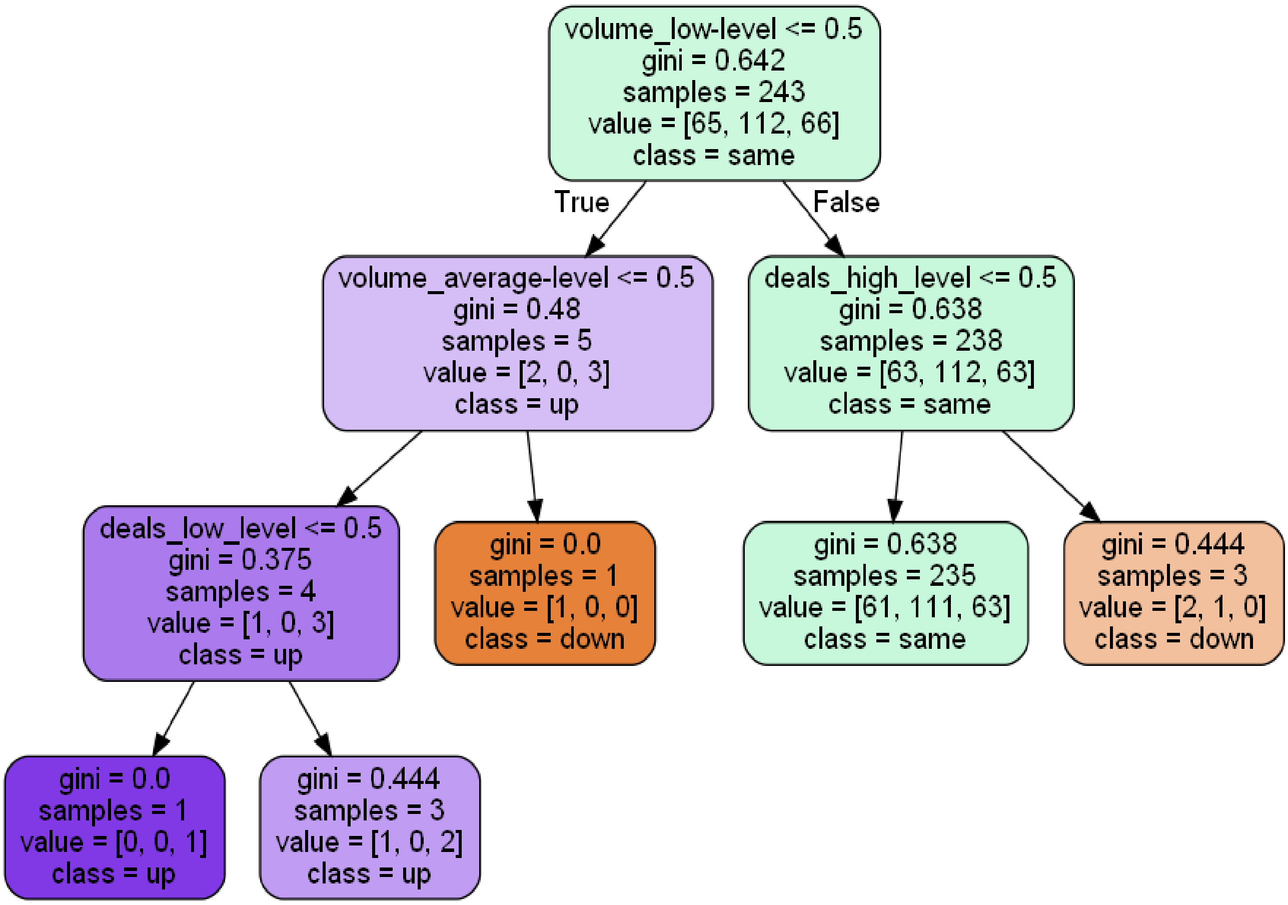

Figure 5.3.1: Decision Tree

Descriptive:

- IF volume (average-level) ,then the predicted stock behavior is down.
- IF volume (high-level) & deals (high-level) , then the predicted stock behavior is up.
- IF volume (heavy-level) & deals (high-level) , then the predicted stock behavior is up.
- IF volume (high-level) & deals (low-level) , then the predicted stock behavior is up.
- IF volume (heavy-level) & deals (low-level), then the predicted stock behavior is up.
- IF volume (low-level) & deals(low-level) , then the predicted stock behavior is same.
- IF volume (low-level) & deals(high-level) , then the predicted stock behavior is down.

Applying Decision Tree model to predict future behavior of the Paltel stock index behavior

The CART algorithm is turned recursively on the non-leaf branches till all data is classified as shown in Figure 5.3.1. Which leaded to prediction the behavior of the Paltel index as following:

Table 5.3.1: Sample of the Paltel stock index prediction.

Volume	deals	Behavior	prediction
low-level	low_level	Same	Same
low-level	low_level	Up	Same
high-level	low_level	Down	up
low-level	High-level	Down	down
low-level	low_level	Up	Same
average-level	low_level	Down	Down
low-level	low_level	Up	Same
low-level	low_level	Same	Same
low-level	low_level	Down	Same

5.5. Summary

In this thesis, we presented a summary of using Decision Tree classifier on the historical prices of the stocks to built decisions that encourage investors to keep or sell the stocks.The results for the proposed model showed that :

1. The future price will remain same if both volume and deals are low-level.
2. If both volume and deals are high-level, volume is heavy-level with deals is high-level ,volume is high-level with deals is low-level and volume is heavy-level with deals is low-level then the predicted stock behavior is up .
3. The future price will decrease if volume is average-level and volume is low-level with deals is high-level.

Chapter Six

Conclusion

The prediction of the behavior of stock market is very complicated because many factors like regional and global economic conditions, socio-political conditions, poor-corporate governance, varying policies of the government, psychological factors of investors etc. have crucial role behind the performance of the market. Due to such complexity it is much better to make investment decisions on the basis of forecast results obtained using Markov chain model and Decision Tree Classifiers as well as giving prime considerations to the factors mentioned above.

In this thesis Markov Chain model had been applied to predict the behavior of Paltel index depending only on the opening price. The predicted results expressed in terms of probability of certain state of Paltel index in the future. The model did not provide the forecasting results in an absolute state. The transition probability matrix and initial state vector used to calculate the probability of Paltel index being in different states in the following days. The steady state probabilities vector observed from the n-th step transition probability matrix. The result of steady state probability matrix showed that the chance of Paltel index will remain same in the near future is 0.4500. The probability that the index will increase in near future is 0.2840 and the index will decrease in the near future with probability 0.2660. The result for expected return time to a certain state starting from the same state showed that the Paltel index will be in increasing state, on average, after three days when it was initially in increasing state, the chain for Paltel index will be in remain same state after two days when it was in remain same state. Initially the chain for Paltel index will be in decreasing state , on average, after four days when it was in decreasing state .We also used the Decision tree Classifier on the historical prices of the Paltel stocks depending on the volume and number of deals to help in giving decision that encourage the investors to keep , sell or buy new stocks .The result showed that the future price will remain same if both volume and deals are low-level .If both volume and deals are high-level, volume is heavy-level with deals is high-level ,volume is high-level with deals is low-level and volume is heavy-level with deals is low-level then the predicted stock behavior is up and the future price will decrease if volume is average-level and volume is low-level with deals is high-level .So Decision Tree can be useful for the investors in making the correct decision depend on the analysis of the historical prices of stocks index to gain any predictive data from that historical data.

This study represented how the Markov model is suitable for the data and the ability of Markov model to forecast index share because of it is random walk, so each state could be linked straight by every other state in the transition matrix, and having good results.

Markov Chain could be useful in prediction more with other factors for decision making. The current study was a case in one company Paltel. In addition, the study performed depends on first order Markov Chain having only three possible state.

It also showed that the knowledge obtained by Decision Tree Classifiers could be used to give a deeper understanding of stock market behavior, however the results for model were not perfect because many factors including but not limited to political events, general economic conditions, and investors' expectations influence stock market.

Therefore this thesis proposes that more studies can be conducted on various companies listed in Palestine, and to deal with factors like regional and global economic conditions, socio-political conditions, poor-corporate governance, varying policies of the government, psychological factors of investors etc. , .

it may be useful to conduct studies with fuzzy states ,using higher order Markov Chain and create Decision Tree depending on more than two attribute, to have better understanding about the behavior of the stock market.

References

[1] E. Acuña Publisher . Preprocessing in Data Mining .International Encyclopedia of Statistical Science Springer-Verlag Berlin Heidelberg . January 2011.

[2] I. Adesokan, P. Ngare and A. Kilishi . Analyzing Expected Returns of a Stock Using The Markov Chain Model and the Capital Asset Pricing Model .Applied Mathematical Sciences, Vol. 11, 2017, no. 56, 2777 – 2788.

[3] Q. Al Radaideh, Adel Abu Assaf and Eman Alnagi. Predicting Stock Prices using Data Mining Techniques .The International Arab Conference on Information Technology (ACIT'2013)

[4] A. Bairagi and S. Kakaty . Analysis of stock market price behaviour: A Markov

Chain approach . 10, pp. 7061-7066 .October 2015.

[5] M. Bhusal . Application of Markov Chain Model in the Stock Market Trend Analysis of Nepal . International Journal of Scientific & Engineering Research Volume 8, Issue 10, ISSN 2229-5518. October-2017

[6] A. Fitriyanto and T. E. Lestari . Application of Markov Chain to stock trend: A study of PT HM Sampoerna, tbk ,3rd Annual Applied Science and Engineering Conference (AASEC 2018).

[7] S. Frederick , Hillier and Gerald J. Lieberman. Introduction to Operations Research, Eighth Edition .New York: McGraw Hill. 2005

[8] Golfarelli and Rizz . Data Warehouse Design: Modern Principles and Methodologies , page 6 . April 2009.

[9] Radhwan H. A. Alsagheer , Abbas F. H. Alharan and Ali S. A. Al-Haboobi. Popular Decision Tree Algorithms of Data Mining Techniques: A Review .International Journal of Computer Science and Mobile Computing , Vol. 6, Issue. 6 . June 2017.

[10] Han J., Kamber M. and Pie J. . Data Mining :Concepts and Techniques Second Edition , pages 48-50,292-303 . 2006.

[11] Butler K.C., and Malaikah, S. . Efficiency and inefficiency in thinly traded stock markets: Kuwait and Saudi Arabia. Journal of Banking & Finance, 16(1): 197.2010

[12] Adam M. .Financial Markets: The Recent Experience of a Developing Economy. Savings and Development, 33(1): 27-40.2009.

[13] R. Pandey, Lalit Mohan, Sanjeev Bisht and Janmejay Pant .Data Mining and Data Warehouse .International Journal on Emerging Technologies ,Special Issue NCETST-2017) 8(1): 155-157(2017).2017.

[14] Y. Riahi, Sara Riahi. Big Data and Big Data Analytics: Concepts. Types and Technologies .International Journal of Research and Engineering , Vol. 5 No. 9 .September-October 2018.

[15] S. Tangirala. Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree ClassifierAlgorithm. International Journal of Advanced Computer Science and Applications,Vol. 11, No. 2.2020.

[16] H. Taylor and S. Karlin, An Introduction to Stochastic Modeling, 3rd edition,pages 5, 95-98,200-250, Academic Pres .1998.

[17] Shobana V. , Maheshwari S. andSavithri M..Study on Big data with Data Mining ,Vol. 4, Issue 4, April 2015.

[18] G. Žitković . Introduction to Stochastic Processes- Lecture Notes ,page 26 .December 2010.

[19] https://www.datacamp.com/community/tutorials/decision-tree-classification-python. December 28th, 2018.last visit April 5th 2021

[20] https://www.warriortrading.com/stockvolume/?fbclid=IwAR2DbztT_8HRAl-w6LaObHbYavWwFeqScPZQPHDCF0ExjOs4hGLRoLTL5n0. last visit March 2th 2021.

[21] https://www.sharechart.com.au/Education/TechnicalIndicators/Number%20of%20Trades.htm?fbclid=IwAR0uzC9CUDa jk3Vywa6p9YMOyVVDGs261qpH1cKvOnpLR2h7JBzsEA_pGQ. last visit March 4th 2021

[22] https://www.kdnuggets.com/2020/02/decision-tree-intuition.html .last visit April 5th 2021

Appendices

1. The Preprocessed Data

State	Frequency
state 1(S)	112
state 2 (U)	66
state 3 (D)	65
Transition Count	**Frequency**
n_11	56
n_12	30
n_13	25
n_21	29
n_22	13
n_23	24
n_31	24
n_32	26
n_33	15

Table 1: Historical data before selecting attributes and before transformation.

Opening price	Volume	Number of deals
4.3	4,508	5
4.3	4,296	12
4.32	62,943	18
4.3	131,436	41
4.3	17,220	8
4.31	6,019	9
4.31	520	2
4.33	2,880	9
4.33	6,920	20
4.31	120	3
4.35	18,780	19
4.31	572	2
4.3	45,928	25
4.3	18,960	7
4.31	74,493	8
4.31	3,900	8
4.32	67,797	32
4.32	2,752	8
4.35	6,089	13
4.39	690	4
4.42	7,151	11
4.42	10,319	17

4.42	9,555	17
4.41	20,197	11
4.42	5,741	13
4.45	7,489	7
4.45	7,243	7
4.45	58,944	24
4.45	14,034	20
4.44	10,073	19
4.49	34,960	22
4.49	3,700	15
4.48	5,306	7
4.47	9,867	16
4.48	4,450	9
4.45	7,192	14
4.45	4,175	13
4.45	8,084	19
4.45	3,155	13
4.42	5,556	17
4.42	4,442	5
4.43	4,978	19
4.43	6,071	9
4.45	9,914	15
4.44	4,200	9
4.44	18,959	26
4.44	64,622	13

4.45	74,423	17
4.45	2,029	6
4.45	51,157	80
4.44	15,515	18
4.42	2,020	5
4.43	9,143	15
4.45	38,906	40
4.44	51,709	19
4.42	21,640	36
4.46	473,276	72
4.42	78,552	50
4.09	850	8
4.03	12,615	16
4	24,955	34
4.02	28,845	25
4	46,597	29
4	11,254	19
4	23,129	30
4	27,068	29
4	497	3
4	2,629	8
4	3,226	6
4	4,550	7
3.98	53,727	27
4	14,402	34

3.98	884	7
3.98	3,500	11
3.96	16,506	36
3.98	21,928	13
4	25,055	22
3.98	23,276	19
3.98	13,255	11
3.97	23,312	35
4	30,919	28
4	18,370	12
4.01	43,771	33
4	23,892	20
4.13	1,500	2
4.08	10,275	19
4.09	1,261	9
4.09	7,597	18
4.09	4,243	14
4.09	11,818	32
4.08	15,186	14
4.08	2,776	9
4.08	11,209	13
4.08	4,233	13
4.08	24,238	18
4.08	895	5
4.08	11,020	5

4.09	56,203	35
4.07	16,337	20
4.09	15,018	31
4.11	31,943	26
4.11	27,586	8
4.11	5,488	14
4.14	78,430	40
4.14	30,959	13
4.13	5,907	14
4.14	77,196	35
4.14	15,240	14
4.14	16,613	26
4.16	14,589	23
4.18	3,460	9
4.18	2,217	7
4.2	8,826	23
4.18	6,834	15
4.18	70,677	27
4.2	9,999	15
4.19	45,459	10
4.19	3,397	13
4.2	7,493	19
4.2	43,803	23
4.19	7,741	7
4.19	8,908	24

4.19	17,227	15
4.2	11,066	20
4.17	39,922	42
4.17	5,749	17
4.17	5,497	9
4.17	12,475	7
4.18	6,600	9
4.17	12,280	23
4.18	13,508	30
4.18	16,535	30
4.18	320	2
4.18	6,995	7
4.16	11,908	17
4.17	22,358	26
4.17	14,062	15
4.17	16,969	28
4.16	19,768	10
4.16	9,425	19
4.16	4,606	12
4.16	1,962	5
4.17	6,567	11
4.15	3,797	11
4.17	159,712	34
4.17	4,119	13
4.17	8,202	9

4.17	1,000	1
4.17	19,186	31
4.1	68,570	86
4.12	5,594	18
4.13	10,833	17
4.13	11,401	27
4.12	20,618	39
4.14	3,850	10
4.14	6,700	12
4.14	1,450	7
4.15	7,728	15
4.15	4,473	13
4.15	6,995	11
4.15	21,794	27
4.14	1,160	5
4.15	3,050	4
4.15	21,580	12
4.15	8,783	9
4.14	820	8
4.14	2,444	10
4.14	5,766	14
4.14	20,710	25
4.14	26,318	20
4.14	32,624	22
4.15	87,146	26

4.14	11,519	16
4.15	4,562	8
4.15	9,241	15
4.14	1,831	8
4.14	21,599	21
4.14	29,071	19
4.14	7,950	11
4.13	2,806	6
4.12	20,404	40
4.12	3,057	15
4.12	5,166	14
4.12	2,295	5
4.12	23,767	30
4.12	14,109	17
4.12	10,889	11
4.12	500	3
4.15	5,504	6
4.15	1,658	4
4.16	44,609	14
4.18	48,512	22
4.19	18,032	19
4.25	4,574	12
4.24	10,066	16
4.2	327,114	13
4.2	2,705	4

4.24	1,291	7
4.2	8,960	17
4.19	2,376	8
4.19	674	4
4.18	6,920	9
4.18	94	1
4.17	2,105	6
4.16	12,825	22
4.15	6,380	12
4.15	1,565	7
4.14	60,279	13
4.15	7,345	17
4.14	28,048	36
4.13	8,770	21
4.14	17,267	21
4.14	21,683	18
4.11	2,741	13
4.12	6,940	10
4.11	10,924	18
4.11	19,369	29
4.11	32,051	9
4.13	2,744	6
4.13	9,509	14
4.12	7,195	13
4.12	880	3

4.12	10,978	17
4.14	640	4
4.15	4,614	5
4.15	8,642	15
4.16	48,696	7
4.16	34,375	6
4.16	2,115	8
4.16	48,100	5
4.16	6,100	6
4.14	5,086	14
4.15	6,174	13
4.15	9,643	7
4.16	1,845	7
4.14	74,190	31
4.14	20,719	17
4.14	6,469	20
4.13	3,666	10
4.12	63,960	27
4.15	240	2
4.12	11,020	24
4.12	4,300	8
4.2	147,220	26

Table 2: Historical data after selecting attributes and after transformation.

Volume	Deals	Behavior
low-level	low-level	Same
low-level	low_level	Up
high-level	low_level	Down
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Down
low-level	low_level	Down
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Up
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Same

low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Down
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Down
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Down
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Up
low-level	low_level	Down

low-level	low_level	Same
low-level	high_level	Same
low-level	low_level	Down
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Up
low-level	low_level	Down
low-level	low_level	Down
heavy-level	high_level	Up
low-level	high_level	Down
low-level	low_level	Down
low-level	low_level	Down
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Down
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Same
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Down

low-level	low_level	Same
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	Up
low-level	low_level	Down
low-level	low_level	Same
low-level	low_level	Down
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	Up
low-level	low_level	down
low-level	low_level	Up
low-level	low_level	down
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	Up
low-level	low_level	down

low-level	low_level	Up
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	down
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	Up
low-level	low_level	Up
low-level	low_level	same
low-level	low_level	Up
low-level	low_level	down
low-level	low_level	same
low-level	low_level	up
low-level	low_level	down
low-level	low_level	same
low-level	low_level	up
low-level	low_level	same
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up

low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up
low-level	low_level	down
low-level	low_level	up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	up
low-level	low_level	same
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up
low-level	low_level	down
high-level	low_level	up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	high_level	down

low-level	low_level	up
low-level	low_level	up
low-level	low_level	same
low-level	low_level	down
low-level	low_level	up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same

low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up
low-level	low_level	same
low-level	low_level	up
low-level	low_level	up
low-level	low_level	up
low-level	low_level	up
low-level	low_level	down
average-level	low_level	down
low-level	low_level	same
low-level	low_level	up
low-level	low_level	down
low-level	low_level	down
low-level	low_level	same

low-level	low_level	down
low-level	low_level	down
low-level	low_level	down
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	up
low-level	low_level	down
low-level	low_level	down
low-level	low_level	up
low-level	low_level	same
low-level	low_level	down
low-level	low_level	up
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up
low-level	low_level	same
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	up
low-level	low_level	up
low-level	low_level	same

low-level	low_level	up
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	up
low-level	low_level	up
low-level	low_level	down
low-level	low_level	same
low-level	low_level	same
low-level	low_level	same
low-level	low_level	down
low-level	low_level	down
low-level	low_level	up
low-level	low_level	down
low-level	low_level	same
high-level	low_level	up

2. Python code

```
    Data Description:
behavior = The target variable,

volume values: average-level, heavy-level, high-level, low-level
deals values: high_level, low_level
behavior values: down, same,up
see the link :
https://medium.com/@randerson112358/python-decision-tree-classifier-example-d73bc3aeca6
'''
# Step 1: Load Python Packages
import pandas as pd # pandas is a Python library that provides fast, flexible, and
expressive data structures. the two primary
# data structures of pandas, Series (one-dimensional) and DataFrame (two-dimensional).
pandas is well suited for inserting
# and deleting columns from DataFrame, for easy handling of missing data
import pip
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.metrics import accuracy_score
import xlrd # library to read data from execl file, if you don't import you will get the
following error:
# ImportError: Missing optional dependency 'xlrd'. Install xlrd >= 1.0.0 for Excel
support Use pip or conda to install xlrd.
# Load Libraries to Visualize Decision Tree
from IPython.display import Image
from sklearn import tree
```

```
import pydotplus
# Step 2: Pre-Process The Data
candidates = pd.read_excel (r'D:\stock1.xlsx')
df = pd.DataFrame(candidates,columns= ['volume', 'deals','behavior'])
print(df)

# Step 3: Subset The Data: (Our new dataset should only have the variables that we will
be using to build the model)
X = df[['volume', 'deals']]
y = df['behavior']

print(' ')
# Convert categorical variable into dummy/indicator variables or (binary vairbles)
essentialy 1's and 0's
one_hot_data = pd.get_dummies(X)
#print the new dummy data
#print(one_hot_data)

#Step 4: Build A DecisionTreeClassifier  (Create Decision Tree classifer object)
# the default criterion in DecisionTreeClassifier is "gini", and we can use another
criterion as in DecisionTreeClassifier(criterion=’gini’,....) clf =
DecisionTreeClassifier()
# Train Decision Tree Classifer
clf_train =clf.fit(one_hot_data, df['behavior'])
#print(clf_train)

# Step 5: Predictions for new data using Decision Tree
# important note: after converting the categorial data into binary using dummy function
the data structure is:
```

```
# volume_average-level, volume_heavy-level, volume_high-level, volume_low-level,
deals_high_level, deals_low_level
# Note: get_dummies fun in pycharm sorts the new (released) attributes by alphabetical
order.
# in dummy  fun: each attribute is converted into number of attributes based on the
number of groups of values.
print(' ')
# predict if it will paly or not for the following case:
# volume=low-level,deals=low_level

prediction1 = clf.predict([[0,0,0,1,0,1]])   # hhhh
print ('Predicted Result1 using Decision Tree Classifier: ', prediction1)

print(' ')
#prediction2 = clf.predict([[0,0,1,0,1,0,1,0,1,0]])   # hhhh
#print ('Predicted Result2 using Decision Tree Classifier: ', prediction2)  # hhhh
print(list(one_hot_data.columns.values))
print(' ')
print(' ')



# step 6: the draw tree #################### Decision tree classifier
# Visualize Decision Tree: first : Create DOT data
#feature_names: should be after dummy fun is used.
# Export/Print a decision tree in DOT format.    DOT format: format as rules if then ...
try to print next command

dot_data = tree.export_graphviz(clf_train, out_file=None,
feature_names=list(one_hot_data.columns.values),
class_names=['down', 'same','up'], rounded=True, filled=True)
#Gini decides which attribute/feature should be placed at the root node, which features
will act as internal nodes or leaf nodes

#Create Graph from DOT data
graph = pydotplus.graph_from_dot_data(dot_data)

# Show graph
Image(graph.create_png())

# Create PNG
# if dot_data in ( graph = pydotplus.graph_from_dot_data(dot_data) )   is dummied the
function: graph.write_png()  will be released using GVEdit framework
graph.write_png("StocksTree.png")
```