ANALYSIS **OF RECESSION USING**

MACHINE LEARNING TECHNIQUES

Submitted in partial fulfillment of the
requirements for the award of

Bachelor of Engineering degree in Computer Science and Engineering

By

KARUNAKARAN.B (Reg.No – 39110517)
TULASI SRINIVAS.K (Reg.No – 39110145)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —A‖ grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KARUNAKARAN.B (39110517) and TULASI SRINIVAS.K (39110145)** who carried out the Project Phase-1 entitled **"ANALYSIS OF RECESSION USING MACHINE LEARNING TECHNIQUES"** under my supervision from June 2022 to November 2022.

Internal Guide

MS.AISWARYA.D

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 20.04.2023

InternalExaminer **ExternalExaminer**

DECLARATION

I, **KARUNAKARAN.B(Reg.No- 39110517),** hereby declare that the Project Phase-1 Report entitled **"ANALSYIS OF RECESSION USING MACHINE LEARNING TECHNIQUES"** done by me under the guidance of **MS.AISWARYA.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 26-04-2023

PLACE: Chennai

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **MS.AISWARYA.D,**for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks toall Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Recession analysis is a well-known tool in hydrological analysis. Its application, however, poses many methodical questions, and throughout the literature numerous solutions have been sought. The quantification of the recession curve involves the selection of an analytical expression, derivation of a characteristic recession and optimization of the recession parameters. A major problem is the high variability encountered in the recession behaviour of individual segments. The segments represent different stages in the outflow process, and a physically based short-term or seasonal influence on the recession rate adds to the problem of deriving a characteristic recession. This project is to analyse the economic recession of a country by applying some machine learning model algorithms. This will let us know about the country development in order to know the needs for that country to the next development. And then we can analyse though a web application.

CHAPTER 1

INTRODUCTION

The COVID-19 has struck the economy of the world. People from many nations and countries, regardless of their economic standings, are suffering due to the catastrophic impact the coronavirus pandemic has had on their economy. Major giant industries and private sectors such as airlines, oil and gas, leisure facilities, and manufacturing have laid off majority of their employees or asked for furlough. The consequences of such devastating economic situation need to be studied and proper remediation actions should be recommended to the authorities such as Federal Reserve or central banks.

As a reasonable approach to deal with analysing the economic impacts of COVID-19, it makes sense to compare this pandemic with previous and similar situations and take the lessons learned there and apply them here. As a comparable case with tons of learned lessons, the Great Recession that hit the financial markets during 2006 - 2009 can be studied and compared with the coronavirus pandemic. Then, the confrontation strategies and remediation the authorities employed during that period can be explored and adapted to minimize the impacts of COVID-19 on economy.

For instance, we can look at the historical data captured by the Federal Reserve, it is also possible to do some other types of analysis using non-structured textual data such as the Federal Reserve statements prepared by the Federal Open Market Committee (FOMC).

Concern and trend analysis is the application of natural language processing (NLP) algorithms on chronological and unstructured textual data. Through concern analysis it is possible to conduct complementary analysis and capture the trends of financial or economical concerns over a given period of time. The building block of concern and then trend analysis is topic modelling (TM) where the candidate topics of a given text are captured automatically using NLP-based topic analysis.

This paper compares Federal Reserve statements for the period between 2005 and 2020 with the goal of capturing the similarities of the Federal Reserve concerns between the Great Recession and the COVID-19 pandemic. To do so, we adapt Latent Drichlet Allocation (LDA) and further use two strategies such as Bag of Words (BoW) and the frequency-based approaches such as the term frequency - inverse document frequency (tf – idf) algorithms in detecting topic

The results of our study show that the consequences and economic impacts of COVID-19 are far deeper damaging than the Great Recession. During the period of COVID-19, the unemployment rate is rocket high (close to 15%) and the interest rate is as low as zero. We present and compare the trend of concerns for these two cases (i.e., the Great Recession and COVID-19) and draw some conclusions. This paper makes the following key contributions:

- We capture the topic and concerns of Federal Reserve statements through NLP-based algorithms.

- The paper compares and contrasts the economic impacts of the Great Recession and COVID-19 using concern analysis.

- A quantitative comparison of the Great Recession and COVID-19 is presented using quantitative data.

This is one of the examples for the great recession of a country. Thus, some country having this kind of great recession. This may cause to unemployment to the whole people who living inside the country. To aware of this kind of problems among the respective country, and take an action in order to the improvement of the development of a country, there is some needs to analysis the recession of a country in terms of all kind of recessions from the scratch and is to analyse the consequences occurred by through this recession. This will help us to improve the country and reduce the recession.

PYTHON:

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.
Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions: Python 2 and Python 3. Both are quite different.

Beginning with Python programming:

1) Finding an Interpreter:

Script Begins

```
print("Geeks Quiz")

# Scripts Ends
```

Before we start Python programming, we need to have an interpreter to interpret and run our programs. There are certain online interpreters that can be used to run Python programs without installing an interpreter.

Windows: There are many interpreters available freely to run Python scripts like IDLE (Integrated Development Environment) that comes bundled with the Python software downloaded from **http://python.org/**.

Linux: Python comes preinstalled with popular Linux distros such as Ubuntu and Fedora. To check which version of Python you're running, type "python" in the terminal emulator. The interpreter should start and print the version number.

macOS: Generally, Python 2.7 comes bundled with macOS. You'll have to manually install Python 3 from **http://python.org/**.

2) Writing our first program:

Just type in the following code after you start the interpreter.

```
# Script Begins

print("Geeks Quiz")

# Scripts Ends
```

Output:

```
Geeks Quiz
```

Let's analyse the script line by line.

Line 1: [# Script Begins] In Python, comments begin with a #. This statement is ignored by the interpreter and serves as documentation for our code.

Line 2: [print (“Geeks Quiz”)] To print something on the console, print () function is used. This function also adds a newline after our message is printed (unlike in C). Note that in Python 2, “print” is not a function but a keyword and therefore can be used without parentheses. However, in Python 3, it is a function and must be invoked with parentheses.

Line 3: [# Script Ends] This is just another comment like in Line 1.

Python designed by Guido van Rossum at CWI has become a widely used general-purpose, high-level programming language.

Prerequisites:

Knowledge of any programming language can be a plus.

Reason for increasing popularity

1. Emphasis on **code readability, shorter codes**, ease of writing
2. Programmers can express logical concepts in **fewer lines** of code in comparison to languages such as C++ or Java.
3. Python supports **multiple** programming paradigms, like object-oriented, imperative and functional programming or procedural.
4. There exist inbuilt functions for almost all of the frequently used concepts.
5. Philosophy is “Simplicity is the best”.

LANGUAGE FEATURES

- **Interpreted**
    - There are no separate compilation and execution steps like C and C++.
    - Directly run the program from the source code.
    - Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.

    - No need to worry about linking and loading with libraries, etc.
- **Platform Independent**
    - Python programs can be developed and executed on multiple operating system platforms.
    - Python can be used on Linux, Windows, Macintosh, Solaris and many more.
- **Free and Open Source;** Redistributable
- **High-level Language**
    - In Python, no need to take care about low-level details such as managing the memory used by the program.
- **Simple**
    - Closer to English language; Easy to Learn
    - More emphasis on the solution to the problem rather than the syntax
- **Embeddable**
    - Python can be used within C/C++ program to give scripting capabilities for the program's users.
- **Robust**:
    - Exceptional handling features
    - Memory management techniques in built
- **Rich Library Support**
    - The Python Standard Library is very vast.
    - Known as the **"batteries included"** philosophy of Python ;It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.

- Besides the standard library, there are various other high-quality libraries such as the Python Imaging Library which is an amazingly simple image manipulation library.

Python vs JAVA

Python	Java
Dynamically Typed • No need to declare anything. An assignment statement binds a name to an object, and the object can be of any type. • No type casting is required when using container objects	**Statically Typed** • All variable names (along with their types) must be explicitly declared. Attempting to assign an object of the wrong type to a variable name triggers a type exception. • Type casting is required when using container objects.
Concise Express much in limited words	**Verbose** Contains more words
Compact	**Less Compact**
Uses Indentation for structuring code	**Uses braces for structuring code**

The classical **Hello World program** illustrating the **relative verbosity** of a Java Program and Python Program

Java Code

```
public class HelloWorld
{
  public static void main (String[] args)
```

```
    {
        System.out.println("Hello,  world!");
    }
}
```

Python Code

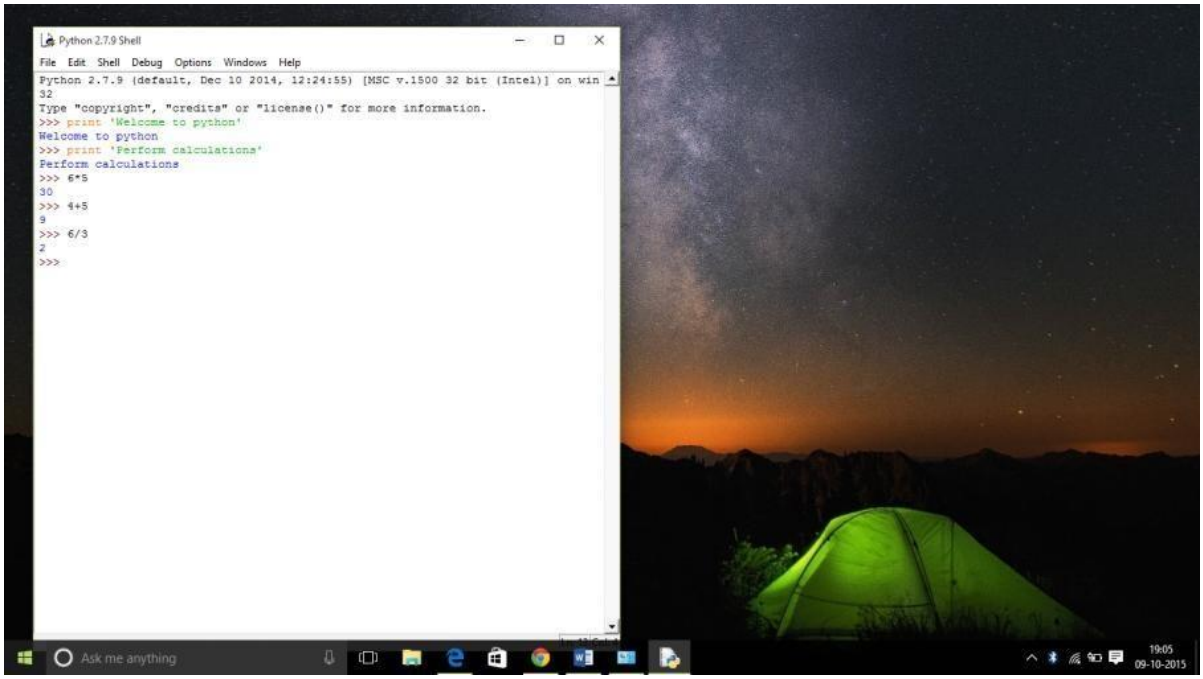
```
print("Hello, world!")
```

Similarity with Java

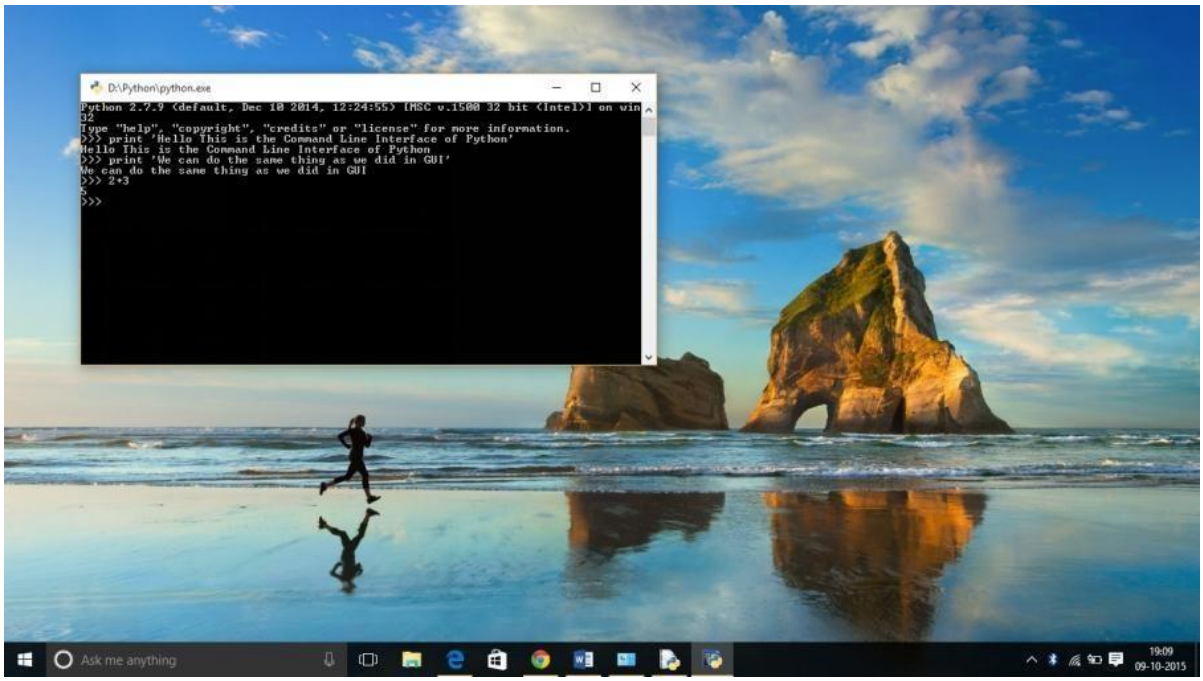
- Require some form of runtime on your system (JVM/Python runtime)
- Can probably be compiled to executables without the runtime (this is situational, none of them are designed to work this way)

LOOK and FEEL of the Python

GUI



Command Line interface



Software's making use of Python

Python has been successfully embedded in a number of software products as a scripting language.

1. GNU Debugger uses Python as a **pretty printer** to show complex structures such as C++ containers.
2. Python has also been used in artificial intelligence
3. Python is often used for **natural language processing** tasks.

Current Applications of Python

1. A number of Linux distributions use installers written in Python example in Ubuntu we have the **Ubiquity**
2. Python has seen extensive use in the **information security industry**, including in exploit development.
3. Raspberry Pi– single board computer uses Python as its principal user-programming language.
4. Python is now being used **Game Development** areas also.

Pros:

1. Ease of use
2. Multi-paradigm Approach

Cons:

1. Slow speed of execution compared to C, C++
2. Absence from mobile computing and browsers
3. For the C, C++ programmers switching to python can be irritating as the language requires proper indentation of code. Certain variable names commonly used like sum are functions in python. So, C, C++ programmers have to look out for these.

Industrial Importance

Most of the companies are now looking for candidates who know about Python Programming. Those having the knowledge of python may have more chances of impressing the interviewing panel. So I would suggest that beginners should start learning python and excel in it.

Python is a high-level, interpreted, and general-purpose dynamic programming language that focuses on code readability. It has fewer steps when compared to Java and C. It was founded in 1991 by developer Guido Van Rossum. Python ranks among the most popular and fastest-growing languages in the world. Python is a powerful, flexible, and easy-to-use language. In addition, the community is very active there. It is used in many organizations as it supports multiple programming paradigms. It also performs automatic memory management.

Advantages:

1. Presence of third-party modules
2. Extensive support libraries (NumPy for numerical calculations, Pandas for data analytics etc)
3. Open source and community development
4. Versatile, Easy to read, learn and write
5. User-friendly data structures

6. High-level language
7. Dynamically typed language (No need to mention data type based on the value assigned, it takes data type)
8. Object-oriented language
9. Portable and Interactive
10. Ideal for prototypes – provide more functionality with less coding
11. Highly Efficient (Python's clean object-oriented design provides enhanced process control, and the language is equipped with excellent text processing and integration capabilities, as well as its own unit testing framework, which makes it more efficient.)
12. (IoT)Internet of Things Opportunities
13. Interpreted Language
14. Portable across Operating systems

Applications:

1. GUI based desktop applications
2. Graphic design, image processing applications, Games, and Scientific/ computational Applications
3. Web frameworks and applications
4. Enterprise and Business applications
5. Operating Systems
6. Education
7. Database Access
8. Language Development
9. Prototyping
10. Software Development

Organizations using Python:

1. Google (Components of Google spider and Search Engine)
2. Yahoo (Maps)
3. YouTube
4. Mozilla
5. Dropbox
6. Microsoft
7. Cisco
8. Spotify
9. Quora

CHAPTER 2

2.1

LITERATURE REVIEWS:

1. **AUTHOR NAME:** Alessi, L., & Detken, C. (2018).

   TITLE: Identifying excessive credit growth and leverage. *Journal of Financial Stability*, *35*, 215-225.

   DESCRIPTION: Unsustainable credit developments lead to the build-up of systemic risks to financial stability. While this is an accepted truth, how to assess whether risks are getting out of hand remains a challenge. To

identify excessive credit growth and aggregate leverage we propose an early warning system, which aims at predicting banking crises. In particular, we use a modern classification tree ensemble technique, the "Random Forest", and include (global) credit as well as real estate variables as predictors.

2. **AUTHOR NAME:** Alessi, L., Antunes, A., Babecký, J., Baltussen, S., Behn, M., Bonfim, D., Bush, O., Detken, C., Frost, J., Guimaraes, R. and Havranek, T., 2015.

   TITLE: Comparing different early warning systems: Results from a horse race competition among members of the macro-prudential research network.

   DESCRIPTION: Over the recent decades researchers in academia and central banks have developed early warning systems (EWS) designed to warn policy makers of potential future economic and financial crises. These EWS are based on diverse approaches and empirical models. In this paper we compare the performance of nine distinct models for predicting banking crises resulting from the work of the Macroprudential Research Network (MaRs) initiated by the European System of Central Banks. In order to ensure comparability, all models use the same database of crises created by MaRs and comparable sets of potential early warning indicators. We evaluate the models' relative usefulness by comparing the ratios of false alarms and missed crises and discuss implications for practical use and future research. We find that multivariate models, in their many appearances, have great potential added value over simple signalling models. One of the main policy recommendations coming from this

exercise is that policy makers can benefit from taking a broad methodological approach when they develop models to set macro-prudential instruments.

3. **AUTHOR NAME:** Athey, S., & Luca, M. (2019).

   TITLE: Economists (and economics) in tech companies. *Journal of Economic Perspectives*, *33*(1), 209-30.

   DESCRIPTION: As technology platforms have created new markets and new ways of acquiring information, economists have come to play an increasingly central role in tech companies-tackling problems such as platform design, strategy, pricing, and policy. Over the past five years, hundreds of PhD economists have accepted positions in the technology sector. In this paper, we explore the skills that PhD economists apply in tech companies, the companies that hire them, the types of problems that economists are currently working on, and the areas of academic research that have emerged in relation to these problems.

4. **AUTHOR NAME:** Biau, G., 2012.

   TITLE: Analysis of a random forests model. Journal of Machine Learning Research, 13(Apr), pp.1063-1095.

   DESCRIPTION: Random forests are a scheme proposed by Leo Breiman in the 2000's for building a predictor ensemble with a set of decision trees that grow in randomly selected subspaces of data. Despite growing interest

and practical use, there has been little exploration of the statistical properties of random forests, and little is known about the mathematical forces driving the algorithm. In this paper, we offer an in-depth analysis of a random forests model suggested by Breiman (2004), which is very close to the original algorithm. We show in particular that the procedure is consistent and adapts to sparsity, in the sense that its rate of convergence depends only on the number of strong features and not on how many noise variables are present.

5. **AUTHOR NAME:** Biau, G., Devroye, L. and Lugosi, G., 2008.

TITLE: Consistency of random forests and other averaging classifiers. Journal of Machine Learning Research, 9(Sep), pp.2015-2033.

DESCRIPTION: In the last years of his life, Leo Breiman promoted random forests for use in classification. He suggested using averaging as a means of obtaining good discrimination rules. The base classifiers used for averaging are simple and randomized, often based on random samples from the data. He left a few questions unanswered regarding the consistency of such rules. In this paper, we give a number of theorems that establish the universal consistency of averaging rules. We also show that some popular classifiers, including one suggested by Breiman, are not universally consistent. Keywords: random forests, classification trees, consistency, bagging.

2.2

EXISTING SYSTEM

- In the existing system, the researcher proposed a model that implies on the work that is analyzing economic recession of the respective country. The datasets are collected and implanted a machine learning algorithm named Random Forest.
- By train the model with the datasets, we can analyze the economic recession of the counties by testing the model with the test data. Finally we may get the exact recession on the required periods through the feature extracted for the classification.

DISADVANTAGES:

- Accuracy of the prediction is very low
- Performance of the model is very low
- Improper results obtained

CHAPTER 3

SYSTEM STUDY

- **3.1. FEASIBILITY STUDY**

  The feasibility of the project is analyzed in this phase and business proposal is put forth N with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- i. Economical Feasibility
- ii. Technical Feasibility
- iii. Social Feasibility

- **3.1.1. Economic Feasibility**

- This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

- **3.1.2. Technical Feasibility**

- This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available

technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

- **3.1.3. Social Feasibility**

- The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel

threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the is the final user of the system.

3.2

REQUIREMENT ANALYSIS

Requirements are a feature of a system or description of something that the system is capable of doing in order to fulfil the system's purpose. It provides the appropriate mechanism for understanding what the customer wants, analysing the needs assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are translated into an operational system.

3.2.1 PYTHON:

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language.

For example, x=10. Here, x can be anything such as String, int, etc.

Python is an interpreted, object-oriented programming language similar to PERL, that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users.

Features in Python

There are many features in Python, some of which are discussed below

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted Language

3.2.2 ANACONDA

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It

will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g., the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

Opensource packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands.

Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

3.2.3 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

Jupyter Notebook can connect to many kernels to allow programming in different languages. By default, Jupyter Notebook ships with the IPython kernel.

As of the 2.3 release (October 2014), there are currently 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell.

The Notebook interface was added to IPython in the 0.12 release (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. According to The Atlantic, Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

3.2.4 RESOURCE REQUIREMENTS:

SOFTWARE REQUIREMENTS:

Operating System	Windows 7or later
Simulation Tool	Anaconda (Jupyter notebook)
Documentation	Ms – Office

HARDWARE REQUIREMENTS:

CPU type	I5 and above

Ram size	4GB
Hard disk capacity	80 GB
Keyboard type	Internet keyboard
Monitor type	15 Inch colour monitor
CD -drive type	52xmax

CHAPTER 4

PROPOSED SYSTEM

- In the proposed system, to create a web application that we use it as analyzer for the purpose of economic recession between the

countries which has chosen by us. Here we collect the datasets of country's information as CSV (Comma Separated Value) file. Then it will be preprocessed by the preprocessing method. Then we separate the datasets as training and test datasets.

- After this, we implement some machine algorithms namely Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbor(KNN), Naïve Bayes(NB) and Decision Tree(DT). We used many of these algorithms for getting the best algorithm based on the accuracy metrics between all the algorithms. Thus, we train the training datasets with all algorithms, we get trained model. Then we get the results of the recession, by testing the test data through the web application. Web application will be a user input window to give the information of respective country and we can get the results of recession in period wise.

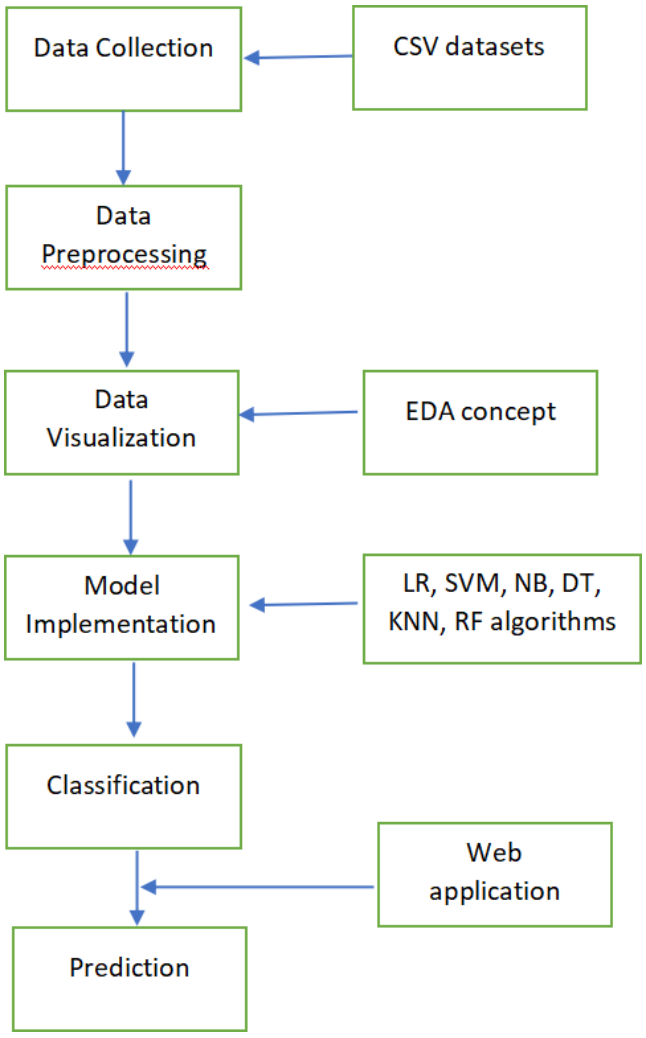
ADVANTAGES:

- Accuracy of prediction is high

- Application is user friendly
- More effective than the existing system

4.2

SYSTEM ARCHITECTURE:



4.3

SYSTEM MODULES:

- MODULE 1: Data Collection
- MODULE 2: Data Preprocessing
- MODULE 3: Data Visualization
- MODULE 4: Model Implementation
- MODULE 5: Classification and Prediction

MODULE 1: Data Collection:

- In this module, we collect the vast datasets from the online which has relates to our project needs of information with the required features
- To get the datasets, a website named Kaggle is one of the most useful website for collecting the datasets

MODULE 2: Data Preprocessing:

- In this module, we preprocessing our whole datasets through our codes only

- Preprocessing is the process of removing null values, replacing the some improper values in the datasets and arranging into perfect format with respect to our current projects.

MODULE 3: Data Visualizing:

- Data visualization is the process of visualize the numerated outputs such as in the graph type or any other types.
- One of the data visualizing method EDA concept is used in this project. It is used to plotting the output in the graph type.

MODULE 4: Model Implementation:

- In this module, we implementing the machine learning algorithms for the purpose of classification. Some of the machine learning algorithms are namely

1. Logistic Regression
2. Random Forest
3. Support vector Machine
4. K-Nearest Neighbor
5. Naïve Bayes
6. Decision Tree

MODULE 5: Classification and Prediction:

- In this module, we classify the recessions on basis of the features which is extracted from the datasets.
- Finally, we predict the recession of respective country in the period wisely like after 6 months, 12 months, 24 months

4.4

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub – assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1. TYPES OF TESTS

6.1.1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the

application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3. FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised

Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4. SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.1.5. WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.1.6. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.1.7. UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

CHAPTER 5

CONCLUSION

The evidence suggests quite clearly that public sector debt played

no causal role in generating the Great Recession. Omitting this variable from the explanatory set in every country makes no difference to the predictive power of the model. In contrast, the ratios of household debt to GDP and non-financial corporate sector debt to GDP does appear to have played a significant role. As Ormerod and Mounfield (2000) show, using modern signal processing techniques, the time series GDP growth data is dominated by noise rather than by signal. So there is almost certainly a quite restrictive upper bound on the degree of accuracy of prediction which can be achieved. However, machine learning techniques do seem to have considerable promise in extending useful forecasting horizons and providing better information to policy makers over such horizons. The future works are implies on the basis of reducing the recession of every county and develop with respect to growth in order to reduce the poverty, unemployment as well as economics.

CHAPTER 6

REFERENCES

- Alessi, L. and Detken, C., 2011. Quasi real time early warning indicators for costly asset price boom/bust cycles: A role for global liquidity. European Journal of Political Economy, 27(3), pp.520-

- Alessi, L., Antunes, A., Babecký, J., Baltussen, S., Behn, M., Bonfim, D., Bush, O., Detken, C., Frost, J., Guimaraes, R. and Havranek, T., 2015. Comparing different early warning systems: Results from a horse race competition among members of the macro-prudential research network.

- Athey, S. and Luca, M., 2019. Economists (and Economics) in Tech Companies. Journal of Economic Perspectives, 33(1), pp.209-30

- Biau, G., 2012. Analysis of a random forests model. Journal of Machine Learning Research, 13(Apr), pp.1063-1095.

- Biau, G., Devroye, L. and Lugosi, G., 2008. Consistency of random forests and other averaging classifiers. Journal of Machine Learning Research, 9(Sep), pp.2015-2033.

- Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32 Breiman, L., 2002. Manual on setting up, using, and understanding random forests v3. 1. Statistics Department University of California Berkeley, CA, USA Fernández-Delgado, M.,

- Cernadas, E., Barro, S. and Amorim, D., 2014. Do we need

hundreds of classifiers to solve real world classification problems. J. Mach. Learn. Res, 15(1), pp.3133- 3181.

- Fildes, R. and Stekler, H., 2002. The state of macroeconomic forecasting. Journal of Macroeconomics, 24(4), pp.435-468

- Friedman, M. and Schwartz, A., 1963. A monetary history of the United States Mullainathan, S. and Spiess, J., 2017. Machine learning: an applied econometric approach. Journal of Economic Perspectives, 31(2), pp.87-106.

- Ormerod, P. and Mounfield, C., 2000. Random matrix theory and the failure of macroeconomic forecasts. Physica A: Statistical Mechanics and its Applications, 280(3), pp.497-504.

- Ribeiro, M.T., Singh, S. and Guestrin, C., 2016, August. Why should I trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144). ACM.