

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

“Jnana Sangama”, Belgaum - 590018

A project report on

**“No Code Machine Learning”**

submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

by

1CR17IS106 Vedant Barbhaya 1CR16IS107 Viraj Patel 1CR16IS110 Vishal Kundar

Under the Guidance of

Mrs. Vidya U

Assistant Professor

Department of ISE, CMRIT, Bengaluru

### CMR INSTITUTE OF TECHNOLOGY

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

#132, AECS Layout, IT Park Road, Bengaluru - 560037 2020-21

Certificate

This is to Certified that the project work entitled **“No Code Machine Learning''** carried out by **Mr Vedant Barbhaya (1CR17IS106), Mr Viraj Patel (1CR17IS107) and Mr Vishal Kundar (1CR17IS110)** in partial fulfilment for the award of Bachelor of Engineering in **Information Science & Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2020-21**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name & Signature of Guide

Dr. Sudhakar K N Assistant Professor, Dept. of ISE, CMRIT

Name & Signature of HoD

Dr. M Farida Begam,

HoD Dept. of ISE, CMRIT

Name & Signature of Principal

Dr. Sanjay Jain Principal CMRIT

**External Viva**

Name of the Examiners Institution Signature with Date

1.

2.

# Declaration

We, **Vedant Barbhaya (1CR17IS106), Viraj Patel (1CR17IS107) and Vishal Kundar (1CR17IS110)** bonafide students of **CMR Institute of Technology**, Bangalore, hereby declare that the dissertation entitled, “**No Code Machine Learning”** has been carried out by us under the guidance of **Prof. Vidya U**, Assistant Professor, CMRIT, Bangalore, in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering, of the Visvesvaraya Technological University, Belgaum during the academic year 2020-2021. The work done in this dissertation report is original and it has not been submitted for any other degree in any university.

Vedant Barbhaya

Viraj Patel

Vishal Kundar

# Acknowledgement

The *satisfaction and euphoria that accompany a successful completion* of any task would be incomplete without the mention of people who made it possible. Success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, it is with gratitude that we acknowledge all those whose guidance and encouragement served as beacons of light and crowned our effort with success.

We would like to thank **Dr Sanjay Jain**, Principal, CMRIT, Bangalore, for providing an excellent academic environment in the college and his never-ending support for the B.E program.

We would like to express our gratitude towards **Dr Farida Begam**, Professor and HOD, Department of Information Science & Engineering CMRIT, Bangalore, who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honour to express our sincere gratitude to our internal guide Mrs **Vidya U.**, Assistant Professor, Department of Information Science & Engineering, CMRIT, Bangalore, for their valuable guidance throughout the tenure of this project work.

We would like to thank all the faculty members who have always been very cooperative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during our project.

Vedant B

Viraj Patel

Vishal Kundar

# Abstract

Data is the oil of the 21st century. Essentially if you are running a service/business, every interaction with a customer generates some kind of valuable data.   
  
With the advancements of intelligent systems, it has become possible to extract very valuable information from any given data.   
  
But not every business has the ability or means to develop an in-house team to analyse vast amounts of data generated by them.

No Code ML is trying to create a platform where a user can upload a data set and get personalized predictions with just one click. This eliminates the need to clean or process the data or create a machine learning model.   
  
No code ML helps businesses to analyse their data using ML without actually needing the knowledge of how to code it.   
  
Everything will be handled by our platform and the user will get a personalized dashboard to analyse the outcomes.

# Table of Contents

[**Declaration**](#_xss1oeiakl0e) **3**

[**Acknowledgement**](#_hqec3qy2leql) **4**

[**Abstract**](#_3t2p5dhh1jl2) **5**

[**Table of Contents**](#_jz1sfslz8yx3) **6**

[**Chapter 1**](#_ijnu3jgjgxyg) **9**

[Preamble](#_2e450yxovvv) 9

[1.1 Introduction](#_yl150ml50ctb) 9

[1.2 Existing System](#_usiqlm1msho7) 10

[1.3 Drawbacks](#_ndyf5x5yzojx) 11

[1.4 Proposed System](#_ijkx0a4bsmtq) 12

[1.5 Plan of Implementation](#_24s363eky2yj) 13

[1.6 Problem Statement](#_y2ia9v5jqkz7) 15

[1.7 Objective of the project](#_lxbuumgn2l0u) 15

[**Chapter 2**](#_uheappddggok) **16**

[Literature Survey](#_eu6te9bfzurl) 16

[**Chapter 3**](#_y5ae5mjvtaqj) **19**

[Theoretical Background](#_h0e5ajn999yt) 19

[3.1 Machine learning](#_huss3pb7h202) 19

[3.2 Supervised Machine learning](#_z602v7yt92bi) 20

[3.3 Automated Machine Learning (AutoML)](#_cgl0zug2o8tm) 21

[3.4 Scikit Learn](#_yde7fyyb0d2l) 23

[3.5 Pandas](#_u1aieardg1np) 24

[3.6 Flask](#_l4jdqqeg93mo) 24

[**Chapter 4**](#_hamaiv5mmjix) **26**

[SYSTEM REQUIREMENT SPECIFICATION](#_5dmng2uwvc4m) 26

[Functional Requirements](#_ecaen394xd34) 26

[Non Functional Requirements](#_49zs2tyjjonn) 27

[Product Requirements](#_go9la1e59qbp) 27

[Basic Operational Requirements:](#_3jnayhwiezs) 27

[Hardware system configuration](#_3od6rxezv4fz) 28

[Software system configuration](#_ah1e6ljcl7az) 28

[**Chapter 5**](#_q7cpvorqwdb1) **29**

[System Analysis](#_swdm90ncg5s7) 29

[Feasibility Study](#_q42xs4u19tmj) 29

[Economical Feasibility](#_opdityx2kf3w) 29

[Technical Feasibility](#_6vuz3h1kjlm4) 29

[Social Feasibility](#_phllfxbejfct) 29

[Analysis](#_9ld4oy48h2b5) 30

[Technical Analysis](#_a7rq5klw7lqa) 30

[**Chapter 6**](#_5lhxkj6djmxv) **31**

[System Design](#_myca8w5zuo9m) 31

[System Development Methodology](#_mlw5t8g8wvn) 31

[Data Flow Diagram](#_ls97r09ch1r8) 31

[Sequence Diagram](#_wn3911m7c1dd) 33

[System Architecture](#_3mp1zaynzopu) 33

[**Chapter 7**](#_so5z2d4iti9j) **36**

[Data validation](#_tpd4c7ir8zdb) 36

[Data Preprocessing:](#_5ph4r8gz4v97) 44

[Model implementation](#_9hhi4d6su4em) 45

[**Chapter 8**](#_61a394sa2rss) **46**

[**Testing**](#_z9wy0ub7lmpm) **46**

[8.2 STLC (Software Testing Life Cycle):](#_4x17i5ng1p5z) 46

[8.2.1 TEST PLAN](#_vlu3zg4lqgdx) 46

[8.2.2 TEST DEVELOPMENT](#_j8x6r517xds9) 47

[8.2.3 TEST EXECUTION](#_w562bflb65v7) 47

[8.2.4 ANALYZE RESULT:](#_d8mdlu6ut0fd) 47

[8.2.5 DEFECT TRACKING:](#_3660bwykk11h) 47

[8.3 TYPES OF TESTING:](#_5eytzcli15b1) 47

[8.3.1 WHITE BOX TESTING:](#_7fj147pb6cyp) 47

[8.3.2 BLACK BOX TESTING:](#_aq8cxgn13321) 48

[8.3.3 GREY BOX TESTING:](#_qwi198cc30el) 48

[8.4 LEVEL OF TESTING USED IN PROJECT](#_wj5pnilmcbbv) 48

[8.4.1 Unit testing](#_j0avce9stz6l) 48

[8.4.2 Integration testing](#_kps329gmzatq) 48

[8.4.3 System testing](#_84pfl4f0keut) 48

[8.4.4 Functional testing](#_p870f4mxjxoe) 49

[8.4.5 Alpha testing](#_5gsjw7kje2bn) 49

[8.4.6 Beta testing](#_f6zw3kylw84q) 49

[8.5 Quality Assurance](#_leudhyj9bs8w) 49

[**Chapter 9**](#_sdmezmfoq5pj) **50**

[**Screen shots**](#_rppcf735nkaf) **50**

[9.1 Modules](#_ytuxaew87lw8) 52

[**Chapter 10**](#_jr6r4tb1ehg3) **54**

[**Result Performance and Analysis**](#_e2r2784527yl) **54**

[**Chapter 11**](#_o5blw66gq7on) **55**

[**Conclusion**](#_1x5b65yb3tpr) **55**

[**Chapter 12**](#_29b63ulotjls) **56**

[**Bibliography**](#_9yldajxs2vv7) **56**

# Chapter 1

## Preamble

## 1.1 Introduction

With the success of Machine Learning (ML) in recent years, it has started attracting a lot of attention from the research and business communities.

Machine Learning involves the design and development of pipelines for applications and ML systems.

Building such a pipeline requires a team of human experts: data scientists having statistical and ML knowledge; domain experts with years of experience within a specific domain. Together, these human experts can build a sensible ML pipeline containing data preprocessing, meaningful feature engineering, and fine-tuned models leading to great results.

Every machine learning service, at its core, needs to solve the same fundamental problems: deciding which machine learning algorithm to use on a given dataset, whether and how to preprocess its features, how to generate meaningful features, and how to tune all hyperparameters.

This process is a complex task, performed in an iterative manner with trial and error. Building a good ML pipeline is a laborious process and practitioners often use a suboptimal default ML pipeline.

To solve these issues, a novel idea of automating the entire pipeline of machine learning (ML) has emerged, i.e., automated machine learning (AutoML).

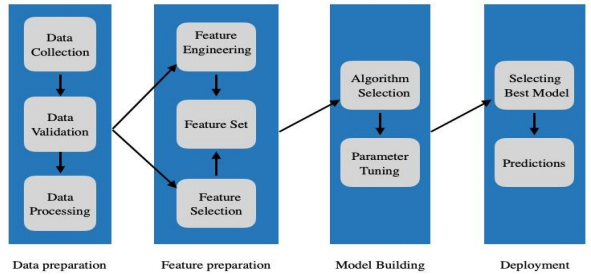
The goal of AutoML is to allow practitioners to build ML applications without much requirement for statistical and ML knowledge.

With the advancements in Cloud and BigData technologies, AutoML has started gaining a lot of attention. A complete AutoML system can dynamically combine various techniques to form an easy-to-use end-to-end ML pipeline system.

In the context of supervised learning, AutoML aims at removing user interaction altogether from all stages of the design and development of supervised learning systems.

As data is being generated at an incredible rate in practically any context and scenario, the number of practitioners available to analyze such data are limited.

AutoML pipeline consists of several processes: data preparation, feature engineering, model generation, and model evaluation. Given below is an illustration of a typical ML pipeline.



## 1.2 Existing System

In the early days of machine learning automation research, researchers focused primarily on hyperparameter optimization.

For example, the most commonly-used form of hyperparameter optimization is grid search, where users apply brute force search to evaluate a predefined range of model parameters to find the model parameters that allows for the best model fit.

More recently, researchers showed that it is possible to discover optimal parameter sets faster than exhaustive grid search by randomly sampling within a predefined grid search, which shows promise for guided search in the hyperparameter space.

Bayesian optimization, in particular, has proven effective for hyperparameter optimization and has even outperformed manual hyperparameter tuning by expert practitioners.

Another focus of AutoML research has been feature construction. One recent example of automated feature construction is the “Data Science Machine,” which automatically constructs features from relational databases via deep feature synthesis.

They demonstrated the crucial role of automated feature construction in machine learning pipelines by entering their Data Science Machine in three machine learning competitions and achieving expert-level performance in all of them.

Thus, we know that automated feature construction can play a vital role in AutoML Systems.

More recently, an AutoML system called auto-sklearn, which uses bayesian optimization to discover the ideal combination of data and feature preprocessors, models, and model hyperparameters to maximize classification accuracy for a particular problem domain.

However, auto-sklearn optimizes over a predefined set of pipelines that only include one data preprocessor, one feature preprocessor, and one model, which precludes auto-sklearn from producing arbitrarily large pipelines that may be important for AutoML.

## 

## 1.3 Drawbacks

AutoML can be viewed as a Combined Algorithm Selection and Hyperparameter optimization (CASH) problem.

Two important problems in AutoML are that (1) no single machine learning method performs best on all datasets and (2) some machine learning methods crucially rely on hyperparameter optimization.

The latter problem has been successfully attacked using Bayesian optimization, which nowadays forms a core component of an AutoML system.

The former problem is intertwined with the latter since the rankings of algorithms depend on whether their hyperparameters are tuned properly.

Another major drawback of current AuntoML platforms is their complexity and cost. A person with little to no experience about data science and machine learning will find it difficult to understand and use the platform.

Also because of how expensive it is to use the platform it discourages small scale industries and individuals.

## 

## 1.4 Proposed System

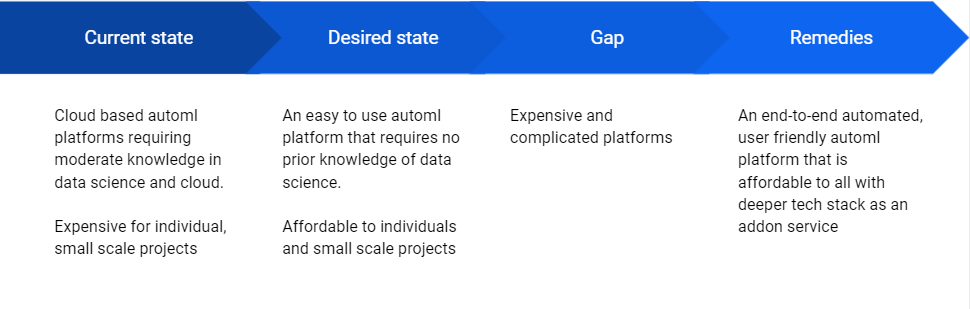
Our project proposes a system of integrated modules. These modules split the problem into simpler parts and form a high level pipeline. The modules are as follows:

* Data validation
* Identification of Problem
* Data preprocessing
* Model Building
* Advanced preprocessing
* Visualization and Results

With these independent modules our system is able to tackle the CASH problem more effectively.

We incorporate a simple to use UI that requires no prior experience to use the platform. The user will be able to reap the benefits of AutoML by simply uploading a file.   
  
Using optimizations and restrictions the platform is affordable to individuals and small scale businesses.

The figure below encompasses the gap between existing systems and what we aim to achieve.



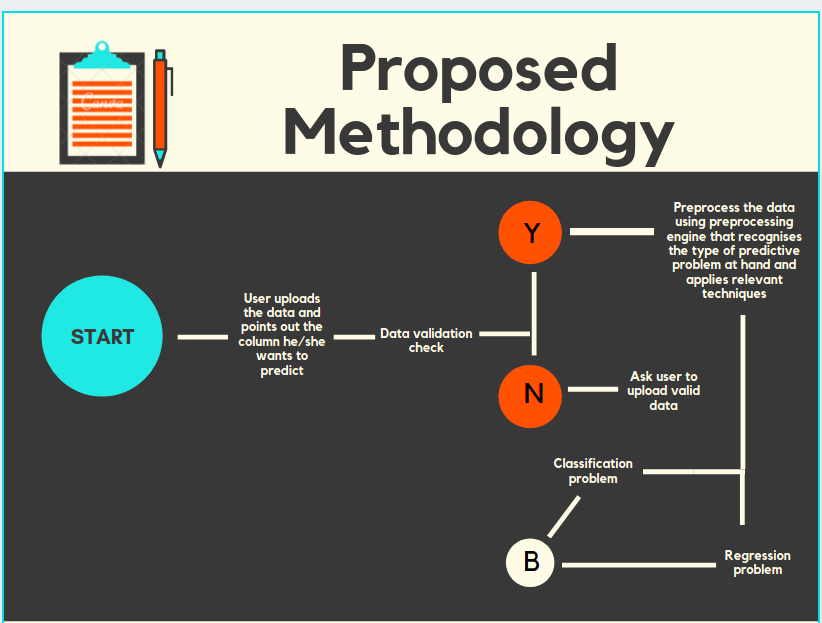
## 1.5 Plan of Implementation

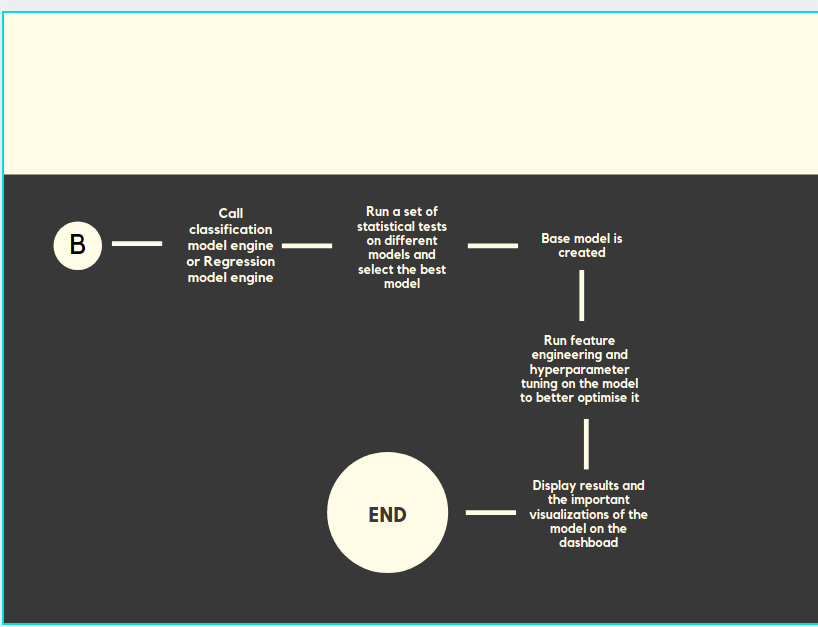
Once the user uploads the data file, our system proceeds to the data validation process. Here the filetype, number of features, data points and their type is checked to ensure its integrity. We set restrictions here that allow us to run the model at low cost and low runtime.

After validation, we identify the type of machine learning problem. Our system handles supervised learning problems which includes Regression and Classification. The type of problem is identified based on the dependent feature having continuous or discrete values.

Based on the problem type the pipeline is branched out into two sub-branches. Since the preprocessing techniques, models and visualizations of Regression and Classification are different, having separate branches for each makes sense.

A basic workflow is shown in the following figures.





The Regression and Classification sub-branch each have their specific techniques for each module, but at a high level perform the same functions. After data validation, basic data preprocessing techniques are applied such as imputing missing values and encoding categorical features.

The next step includes fitting a series of models, these are further subjected to advanced techniques such as hyperparameter tuning, feature selection etc.. to obtain the best out of each model.

The best model is selected using appropriate metrics. Visualizations and results are then displayed to the user, these include details about the features itself, their correlation with the dependent feature and the model results.

The user also has the option to test the model with new data either through file upload or by inputting individual data points.

## 1.6 Problem Statement

Data is the oil of the 21st century. Essentially if you are running a service/business, every interaction with a customer generates some kind of valuable data.

With the advancements of intelligent systems, it has become possible to extract very valuable information from any given data.

But not every business has the ability or means to develop an in-house team to analyze vast amounts of data generated by them.

No code ML helps businesses to analyse their data using ML without actually needing the knowledge of how to code it. Our system will give the user a personalised dashboard to analyse their outcomes.

## 

## 1.7 Objective of the project

An AutoML platform that allows individuals and small businesses to reap the benefits of data science and supervised machine learning to further enhance their business processes at low expense.

No code ML helps businesses to analyse their data using ML without actually needing the knowledge of how to code it. Our system will give the user a personalised dashboard to analyse their outcomes.

We also aim to inspire other ML enthusiasts to develop similar projects to help people and businesses expand at affordable prices.

# Chapter 2

## Literature Survey

The following papers are being referred to in this project.

| **SL.NO** | **Title** | **Authors** |
| --- | --- | --- |
| 1 | Identifying and Harnessing the Building Blocks of Machine Learning Pipelines for Sensible Initialization of a Data Science Automation Tool | Randal S. Olson and Jason H. Moore |
| 2 | Efficient and Robust Automated Machine Learning | Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, Frank Hutter |
| 3 | FLAML: A Fast And Lightweight AutoML Library | Chi Wang, Qingyun Wu, Markus Weimer, Erkang Zhu |

**2.1 Identifying and Harnessing the Building Blocks of Machine Learning Pipelines for Sensible Initialization of a Data Science Automation Tool**

**Goal**: A Tree-based Pipeline Optimization Tool (TPOT) that automatically designs and optimizes machine learning pipelines for a given problem domain, without any need for human intervention.

TPOT optimizes machine learning pipelines using a version of genetic programming (GP), a well-known evolutionary computation technique for automatically constructing computer programs.

Combining GP with Pareto optimization enables TPOT to automatically construct high-accuracy and compact pipelines that consistently outperform basic machine learning analyses.

**Observations:** TPOT-SI saw significant improvement in performance on only one benchmark out of 160. But even with this it did not significantly degrade performance on any of the other benchmarks.

**2.2 Efficient and Robust Automated Machine Learning**

**Goal:** The goal is to follow and extend the AutoML approach first introduced by AUTO-WEKA. At its core, this approach combines a highly parametric machine learning framework F with a Bayesian optimization method for instantiating F well for a given dataset.

The contribution of this paper is to extend this AutoML approach in various ways that considerably improve its efficiency and robustness, based on principles that apply to a wide range of machine learning frameworks.

First, following successful previous work for low dimensional optimization problems, they reason across datasets to identify instantiations of machine learning frameworks that perform well on a new dataset and warmstart Bayesian optimization with them.

Second, automatically construct ensembles of the models considered by Bayesian optimization.

Third, carefully design a highly parameterized machine learning framework from high-performing classifiers and preprocessors implemented in the popular machine learning framework scikit-learn.

Finally, perform an extensive empirical analysis using a diverse collection of datasets to demonstrate that the resulting AUTO-SKLEARN system outperforms previous state-of-the-art AutoML methods.

**Observations:** The new AutoML system AUTO-SKLEARN performs favorably against the previous state of the art in AutoML, and that meta-learning and ensemble improvements for AutoML yield further efficiency and robustness.   
  
There are certain shortcomings, it does not tackle regression or semi-supervised problems. Most importantly, though, the focus on scikit-learn implied a focus on small to medium-sized datasets only.

**2.3 FLAML: A Fast And Lightweight AutoML Library**

**Goal:** A study of the problem of using low computational cost to automate the choices of learners and hyperparameters for an ad-hoc training dataset and error metric, by conducting trials of different configurations on the given training data.

Build a fast and lightweight library FLAML which optimizes for low computational resources in finding accurate models. FLAML integrates several simple but effective search strategies into an adaptive system.

FLAML leverages the structure of the search space to choose a search order optimized for both cost and error. It iteratively decides the learner, hyperparameter, sample size and resampling strategy while leveraging their compound impact on both cost and error as the search proceeds.

To satisfy these properties, they integrate several non-traditional search strategies judiciously because commonly employed strategies do not sufficiently exploit the analyzed relations of the multiple factors.

FLAML is designed for robustly adapting to an ad-hoc dataset out of the box, without relying on expensive preparation such as meta-learning.

**Observations:** It significantly outperforms top-ranked AutoML libraries on a large open source AutoML benchmark under equal, or sometimes orders of magnitude smaller budget constraints.

# Chapter 3

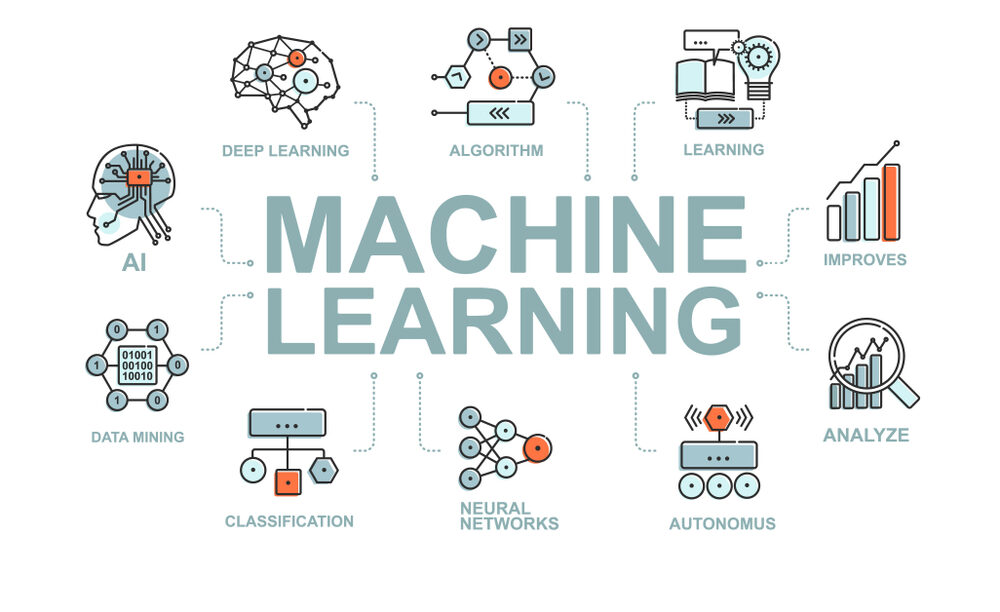
## Theoretical Background

## 3.1 Machine learning

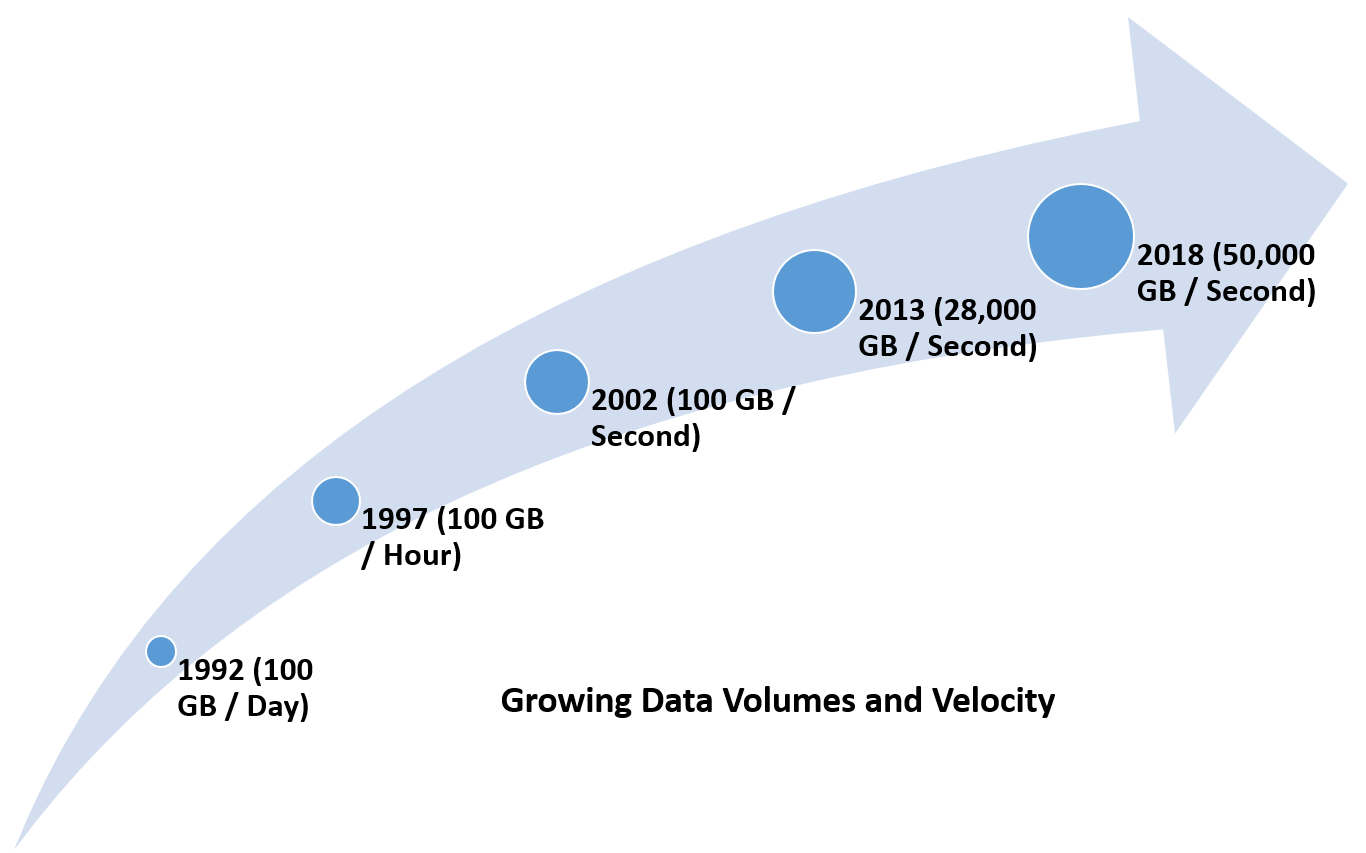
Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal to no human intervention.

Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data.

The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results.



Resurging interest in machine learning is due to the same factors that have made data mining and Bayesian analysis more popular than ever. Things like growing volumes and varieties of available data, computational processing that is cheaper and more powerful, and affordable data storage.



Most industries working with large amounts of data have recognized the value of machine learning technology. By gleaning insights from this data – often in real time – organizations are able to work more efficiently or gain an advantage over competitors.

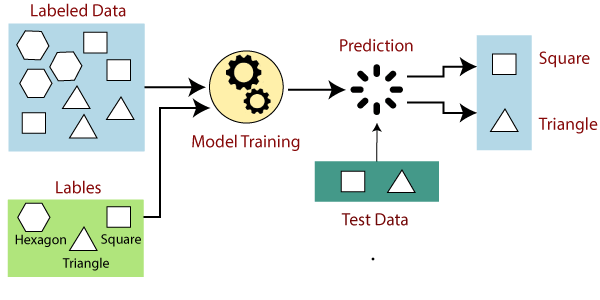
Two of the most widely adopted machine learning methods are supervised learning and unsupervised learning.

## 

## 3.2 Supervised Machine learning

Supervised learning is the type of machine learning in which machines are trained using well labelled training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data, and then it predicts the output.



Supervised learning can be further divided into two types of problems:

* Regression
* Classification

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.

Classification algorithms are used when the output variable is categorical, which means there are two or more classes such as Yes-No, Male-Female, True-false, etc.

Supervised learning models can be a valuable solution for eliminating manual classification work and for making future predictions based on labeled data. However, formatting your machine learning algorithms requires human knowledge and expertise to avoid overfitting data models.

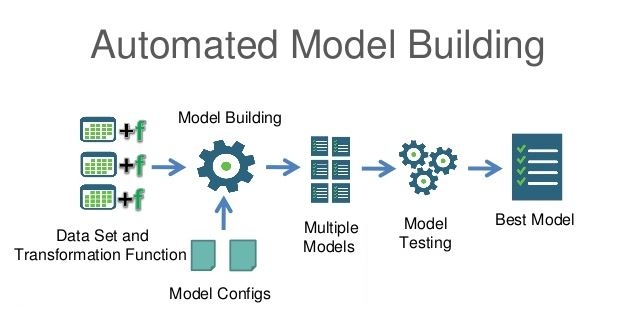
## 3.3 Automated Machine Learning (AutoML)

Automated machine learning (AutoML) represents a fundamental shift in the way organizations of all sizes approach machine learning and data science. Applying traditional machine learning methods to real-world business problems is time-consuming, resource-intensive, and challenging. It requires experts in several disciplines, including data scientists.

Automated machine learning changes that, making it easier to build and use machine learning models in the real world by running systematic processes on raw data and selecting models that pull the most relevant information from the data.

In manual machine learning there are countless opportunities for human error and bias, which degrades model accuracy and devalues the insights you might get from the model.  
  
AutoML covers the complete pipeline from the raw dataset to the deployable machine learning model. AutoML was proposed as an artificial intelligence-based solution to the ever-growing challenge of applying machine learning.  
  
The high degree of automation in AutoML allows non-experts to make use of machine learning models and techniques without requiring them to become experts in machine learning.

Automating the process of applying machine learning end-to-end additionally offers the advantages of producing simpler solutions, faster creation of those solutions, and models that often outperform hand-designed models.



## 

## 

## 

## 

## 

## 3.4 Scikit Learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

* **NumPy:** Base n-dimensional array package
* **SciPy:** Fundamental library for scientific computing
* **Matplotlib:** Comprehensive 2D/3D plotting
* **IPython:** Enhanced interactive console
* **Sympy:** Symbolic mathematics
* **Pandas:** Data structures and analysis

The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as easy of use, code quality, collaboration, documentation and performance.

The library is focused on modeling data. It is not focused on loading, manipulating and summarizing data. For these features, refer to Pandas.

Scikit-learn is largely written in Python, and uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance.

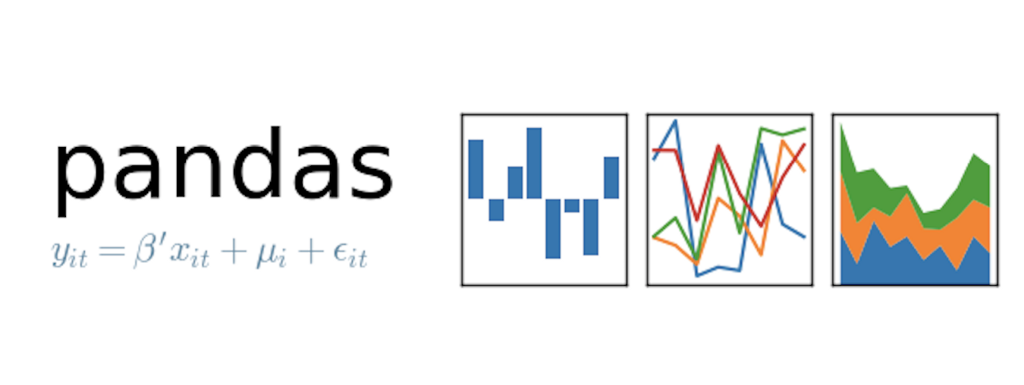


## 3.5 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

It is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel.

Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.  
  
It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. It works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution.



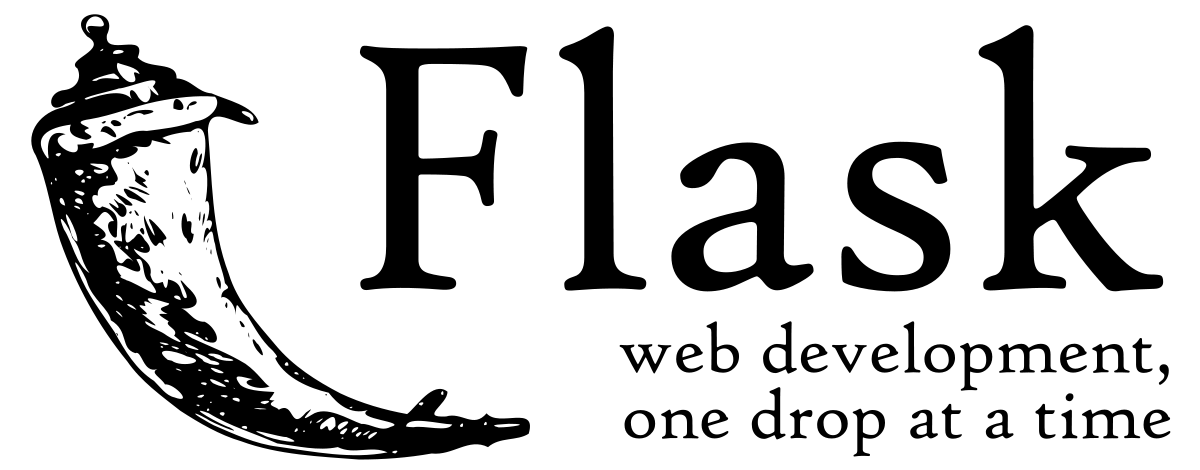
## 3.6 Flask

Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.

It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.



# Chapter 4

## SYSTEM REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may include a set of use cases that describe user interactions that the software must provide. In order to fully understand one’s project, it is very important that they come up with an SRS listing out their requirements, how are they going to meet it and how will they complete the project. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the parent document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

### Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. Following are the functional requirements on the system:

* + - A dataset is uploaded on the system that makes the base object of the system
    - The date set will be subjected to various data mining techniques and predictive models are fitted on the data
    - The baseline models are then subjected to hyperparameter tuning and a profiling mechanism.

### Non Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emerge system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user's needs, because of budget constraints, organizational policies and the need for interoperability with other software and hardware systems.

### Product Requirements

Correctness: It followed a well-defined set of procedures and rules to engage with data and and a predefined path of model building and evaluation to achieve the end goal of the system

Modularity: The complete product is broken up into many modules and well defined interfaces are developed to explore the benefit of flexibility of the product.

Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevance and correctness. Non-functional requirements are also called the qualities of a system.

## 

### Basic Operational Requirements:

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points: -

Mission profile or scenario: It describes the procedures used to accomplish mission objectives. It also finds out the effectiveness or efficiency of the system.

Performance and related parameters: It points out the critical system parameters to accomplish the mission.

Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.

Operational life cycle: It defines the system lifetime.

### Hardware system configuration

Processor: Pentium Processor and Above

Ram: 4GB

## 

### Software system configuration

Operating system: Linux, Windows 10

Programming Language: Python

Framework: Anaconda

Browser - Chrome, Firefox

DL Libraries: Numpy, Pandas, Scikit-learn

# Chapter 5

## System Analysis

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluate the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

## Feasibility Study

All systems are feasible when provided with unlimited resources and infinite time. It is a formally documented output that summarizes results of the analysis and evaluations conducted to review the proposed solution and investigate project alternatives for the purpose of identifying if the project is really feasible, cost-effective and profitable. It describes and supports the most feasible solution applicable to the project. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. If project risk is great, the feasibility of producing quality software is reduced. In this case there are three key considerations involved in the feasibility analysis.

#### Economical Feasibility

The cost of modeling data for predictive analysis requires the concerned party to incur the cost of a trained professional, a dedicated computer system. This system reduces that cost by performing similar tasks on a platform.

#### Technical Feasibility

The framework takes in the dataset and gives the best possible data modelling pipeline which is ensured by evaluation checks at several places throughout the system.

#### Social Feasibility

As the machine does most of the analyzing part of the data generated, the users can focus on more specific domain and business related tasks that go hand in hand with data analysis and modelling.

## Analysis

#### Technical Analysis

The performance of the system can be increased if the technical analysis is done well. The system's hardware requirements must be taken into consideration. The software must go hand in hand with the hardware else the efficiency of the system deteriorates.

1. **Changes to bring in the system:** All changes should be in a positive direction, there will be increased level of efficiency and better customer service.
2. **Required skills:** Platforms tools used in this project are widely used.
3. **Acceptability:** The structure of the system is kept feasible enough so that there should not be any problem from the users point of view.

# Chapter 6

## System Design

#### System Development Methodology

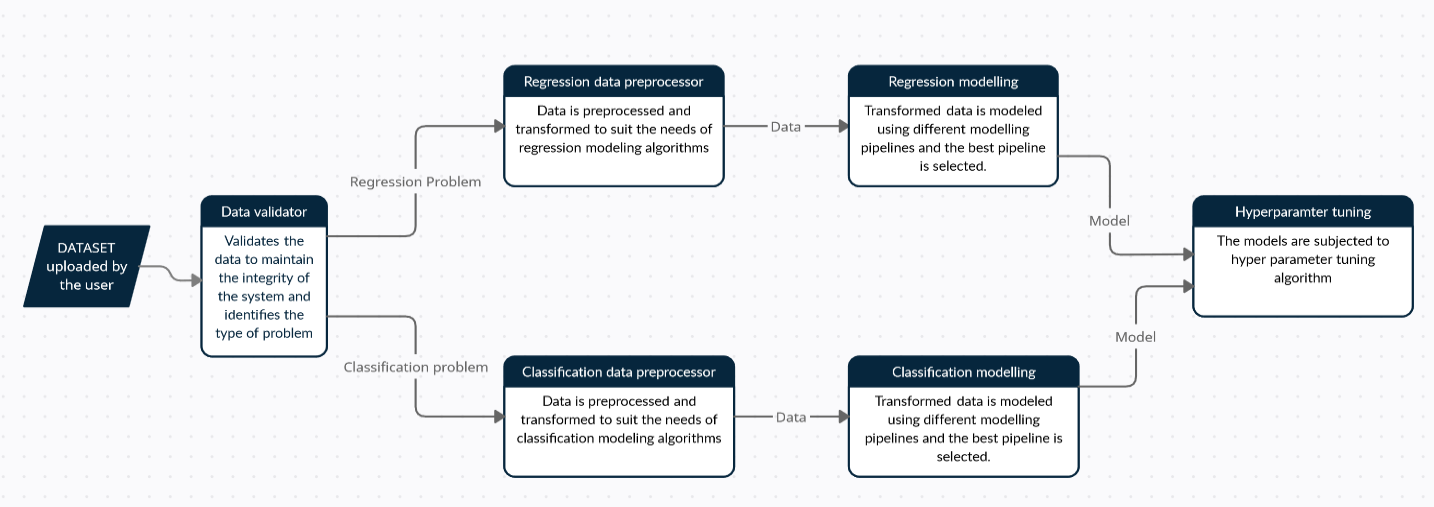
System Development methodology is the development of a system or method for a unique situation. Having a proper methodology helps us in bridging the gap between the problem statement and turning it into a feasible solution. It is usually marked by converting the System Requirements Specifications (SRS) into a real world solution.

System design takes the following inputs:

* Statement of work.
* Requirement determination plan.
* Current situation analysis.
* Proposed system requirements including a conceptual data model and metadata (data about data).

#### Data Flow Diagram

Data Flow Diagrams (DFD) are used to graphically represent the flow of data in a program or information system. DFD describes the processes that are involved in a system to perform a particular task. It usually provides a step by step process to solve a particular problem. The visual representation is extremely useful for quick and easy understanding, making it easy to understand for new arrivals to a team and to technically challenged persons. The various levels in the DFD are used to represent various levels of abstraction in the system.

**Fig: Data Flow Diagram for NO-CODE ML**

#### Sequence Diagram

##### 

##### System Architecture

A system architecture is a conceptual model using which we can define the structure and behaviour of that system. It is a formal representation of a system. Depending on the context, system architecture can be used to refer to either a model to describe the system or a method used to build the system. Building a proper system architecture helps in analysis of the project, especially in the early stages.

The system is designed in a modular manner to make the individual components work individually in an efficient manner and also are designed to be cohesive with each other to create the core functional part of the system.

The main components of the core functional stack are:

1. Data validator
2. Data preprocessor
3. Model engine
4. Data profiler
5. Prediction mechanism
6. Data validation and preprocessing

The client end of the system is realized and implemented using web based technologies which allows the user to upload any dataset from its local computer. The uploaded dataset is passed using a data validator to check if the dataset meets certain criteria which are the base requirements for a dataset in our system for performance and design purposes.It is at this time that data is identified as a regression problem or a classification problem.

1. Data preprocessor

Once a dataset has been accepted by the validator, it is passed on to the data preprocessor of the type of machine learning problem the dataset belongs to. Certain common data preprocessing as well as data preprocessing required for each type of problem is applied on the dataset. The goal of this component is to transform the data into a model ready dataset. It is an essential prerequisite of fitting an algorithm in the best possible way on the dataset.

1. Model engine

Each type of problem - classification and regression, has its own model engine. The model engine in its essence creates a hypothesis space of all possible pipelines of model fitting on the dataset, runs all the candidates and selects the best three candidates from baseline model building. The top three candidates are then subjected to optimization using Randomized search for hyperparameter tuning. The goal of this component is to identify the best possible candidate from the search space of possible model pipelines and optimise them for best possible results.

* Evaluation criteria

The criteria to evaluate a model is as important as the model itself. Different metrics convey different information about the model performance and each metric has its own merit and demerit. For regression analysis, we concluded on using root mean squared error and adjusted R2 and for classification problems, we will be using F1score and accuracy as the metrics of choice.

1. Data profiler

Data profiler is responsible for giving insights to the data through visualizations. Some of the data insights are following:

* Correlation
* Univariate analysis of features
* Chi square analysis
* Statistical analysis of the data
* Outlier analysis

1. Prediction mechanism

Prediction mechanism is the last component of the system. Once the model is fitted to the dataset, the model can be utilized by the prediction mechanism to predict new data.

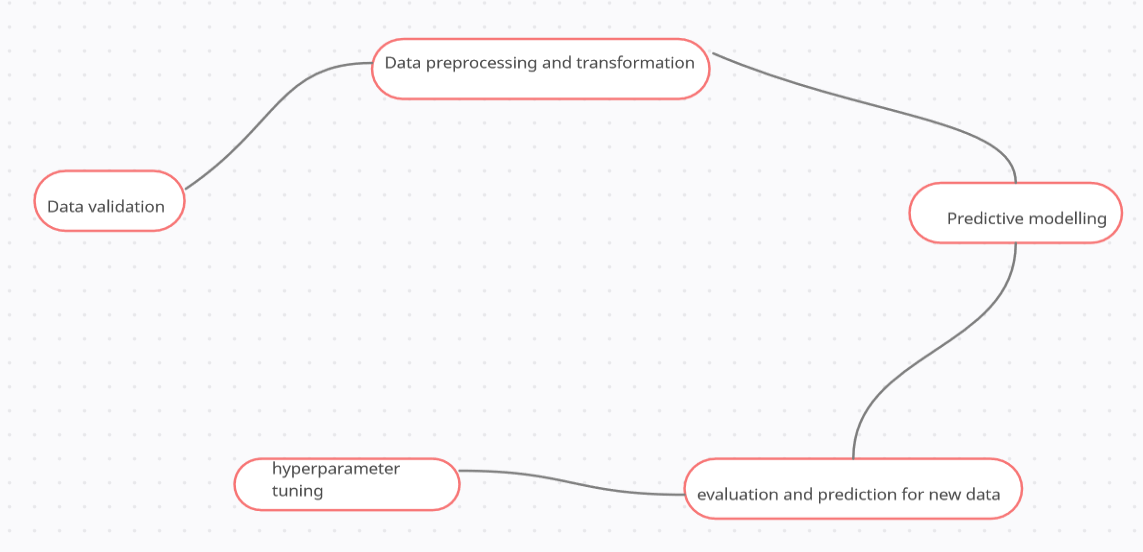


Fig: Sequence diagram for the system

# Chapter 7

**Implementation**

## Data validation

"""

data\_validation.py takes the data file given by user and performs validation

checks. The steps include:

1. Identifying type of data file.(csv xlsx tsv json)

2. Converting to data frame

3. Validation check of data (2 columns min. 100000 > Rows > 300 allowed

data types: num, text) (Clean, format and quality check).No two columns

headers should be the same.

4. Function to display data to user to select dependant variable

5. Identifying if problem is classification or regression based on the dependant variable

"""

#Packages

import pandas as pd

import numpy as np

class data\_check:

"""

Consists of methods to deal with the above steps.

Class variables

---------------

\_\_filename: TYPE: Private class string variable.

DESCRIPTION: Holds filename given by user

\_\_filetype: TYPE: Private class string variable.

DESCRIPTION: Holds filetype found using identify\_file() method

\_\_fileValid: TYPE: Private class boolean variable.

DESCRIPTION: If the file is found to have problems it prevents

usage of file\_to\_dataframe() and identify\_problem() methods

"""

def \_\_init\_\_(self, filename):

"""

Constructor to initialize class. Initializes class variable filename

with given file name and filetype to empty string.

Parameters

----------

filename : TYPE : String

DESCRIPTION : Name of file entered by user.

Returns

-------

None

"""

#private class variables

self.\_\_filename = filename

self.\_\_filetype = ""

self.\_\_fileValid = False

def identify\_file(self):

"""

identifying type of data file.(csv xlsx tsv json)

Sets filetype and filevalid variable based on whether or not file is

valid.

Parameters

----------

None

Returns

-------

None if file is of acceptable else returns log message that file type

cannot be used.

"""

self.\_\_filetype = ""

self.\_\_fileValid = False

allowed\_ext = ["csv","tsv","xlsx","json"]

extension = self.\_\_filename.split(".")[1]

if extension in allowed\_ext:

self.\_\_filetype = extension

self.\_\_fileValid = True

return "None"

else:

return "File type not supported!"

def validation\_check(self, df):

"""

Validation check of data (2 columns min. 100000 > Rows > 300 allowed

data types: num, text) (Clean, format and quality check). No two columns

headers should be same.

Sets fileValid variable based on whether or not it is valid.

Parameters

----------

df : TYPE : Dataframe

DESCRIPTION : Dataframe consisting of data extracted from file.

Returns

-------

None if file is of acceptable type else returns log message that file

has issues.

"""

if self.\_\_fileValid == False:

return "Error: File type not valid! Accepted - [csv, tsv, json, xlsx]"

else:

self.\_\_fileValid = False

self.\_\_no\_cols = len(df.columns)

self.\_\_no\_rows = df.shape[0]

if self.\_\_no\_cols < 2:

return "Error: Insufficent number of features!"

if self.\_\_no\_rows < 1 or self.\_\_no\_rows > 100000: #300

return "Error: Number of examples out of range! 100000 > examples > 300"

self.\_\_fileValid = True

return "None"

def file\_to\_dataframe(self):

"""

Converting file type to pandas data frame

Parameters

----------

None

Returns

-------

dataframe consisting of file data.

"""

if self.\_\_fileValid == False:

return "File not valid!"

else:

if (self.\_\_filetype == "csv"):

return pd.read\_csv(self.\_\_filename)

elif (self.\_\_filetype == "tsv"):

return pd.read\_csv(self.\_\_filename, sep='\t')

elif(self.\_\_filetype == "xlsx"):

#expects single sheet only in xlsx file

return pd.read\_excel(self.\_\_filename, sheet\_name=None)

elif (self.\_\_filetype == "json"):

#try this out

return pd.read\_json(self.\_\_filename)

else:

return """File type not supported! File type supported -> csv,

tsv, json and xlsx. File type given -> """ + self.\_\_filetype

def identify\_problem(self, df, y):

"""

Identifying if problem is classification or regression based on the

dependant variable

Parameters

----------

df : TYPE : Dataframe

DESCRIPTION : Dataframe consisting of data extracted from file.

y : TYPE : String

DESCRIPTION : Column header of dependant variable

Returns

-------

String stating if the problem is regression or classification. If any

problem occurs then log message is sent back.

"""

if self.\_\_fileValid == False:

return "Error: File/Data not valid!"

else:

try:

data = df[y].values

if(len(np.unique(data)) > 10):

return "Regression"

else:

return "Classification"

except KeyError:

return "Error: column header not found!"

except Exception as e:

return "Error: " + str(e)

## 

## Data Preprocessing:

Data preprocessing includes the following checks:

1. Null values
2. Duplicate values
3. Outlier analysis
4. Data imputation
5. Data encoding
6. Data balancing
7. Splitting the dataset

## Model implementation

**i. Classification**

from sklearn.linear\_model import LogisticRegression

from sklearn.linear\_model import LogisticRegressionCV

from sklearn.linear\_model import PassiveAggressiveClassifier

from sklearn.svm import LinearSVC

from sklearn.svm import NuSVC

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.neural\_network import MLPClassifier

from sklearn.discriminant\_analysis import QuadraticDiscriminantAnalysis

from sklearn.ensemble import AdaBoostClassifier

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import RandomForestClassifier

from xgboost import XGBClassifier

from catboost import CatBoostClassifier

import warnings

**ii. Regression**

from sklearn.linear\_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.neighbors import KNeighborsRegressor

from sklearn.ensemble import AdaBoostRegressor

from sklearn.svm import SVR

import xgboost as xgb

import catboost as cb

# Chapter 8

# Testing

**8.1 Software testing introduction**

Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. Software testing is the process used to measure the quality of developed software. Testing is the process of executing a program with the intent of finding errors. Software testing is often referred to as verification & validation.

## 8.2 STLC (Software Testing Life Cycle):

Testing itself has many phases i.e. is called STLC. STLC is part of SDLC (Software Development Life Cycle)

● Test Plan

● Test Development

● Test Execution

● Analyse Result

● Defect Tracking

### 8.2.1 TEST PLAN

It is a document which describes the testing environment, purpose, scope, objectives, test strategy, schedules, milestones, testing tool, roles and responsibilities, risks, training, staffing and who is going to test the application, what type of tests should be performed and how it will track the defects.

###### 

###### 

###### 

### 8.2.2 TEST DEVELOPMENT

Preparing test cases, test data, preparing test procedure, preparing test scenario, Writing test script.

### 8.2.3 TEST EXECUTION

In this phase we execute the documents that are prepared in the test development phase.

### 8.2.4 ANALYZE RESULT:

Once executed documents will get results either pass or fail. We need to analyse the results during this phase.

### 8.2.5 DEFECT TRACKING:

Whenever we get defects on the application we need to prepare the bug report file and forward it to Test Team Lead and Dev Team. The Dev Team will fix the bug. Again we have to test the application. This cycle repeats till we get the software without defects.

## 8.3 TYPES OF TESTING:

● White Box Testing

● Black Box Testing

● Grey box testing

### 8.3.1 WHITE BOX TESTING:

White box testing as the name suggests gives the internal view of the software. This type of testing is also known as structural testing or glass box testing as well, as the interest lies in what lies inside the box.

### 8.3.2 BLACK BOX TESTING:

It’s also called behavioural testing. It focuses on the functional requirements of the software. Testing either functional or non-functional without reference to the internal structure of the component or system is called black box testing.

### 

### 8.3.3 GREY BOX TESTING:

Grey box testing is the combination of black box and white box testing. Intention of this testing is to find out defects related to bad design or bad implementation of the system.

## 

## 8.4 LEVEL OF TESTING USED IN PROJECT

### 8.4.1 Unit testing

Initialization testing is the first level of dynamic testing and is the first responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable.

### 8.4.2 Integration testing

All modules which make applications are tested. Integration testing is to make sure that the interaction of two or more components produces results that satisfy functional requirements.

### 8.4.3 System testing

To test the complete system in terms of functionality and non-functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

### 8.4.4 Functional testing

The outgoing links from all the pages from the specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields. Wrong inputs to the fields in the forms.

### 8.4.5 Alpha testing

Alpha testing is final testing before the software is released to the general public. This testing is conducted at the developer site and in a controlled environment by the end user of the software.

### 8.4.6 Beta testing

The beta test is conducted at one or more customer sites by the end user of the software. The beta test is conducted at one or more customer sites by the end user of the software.

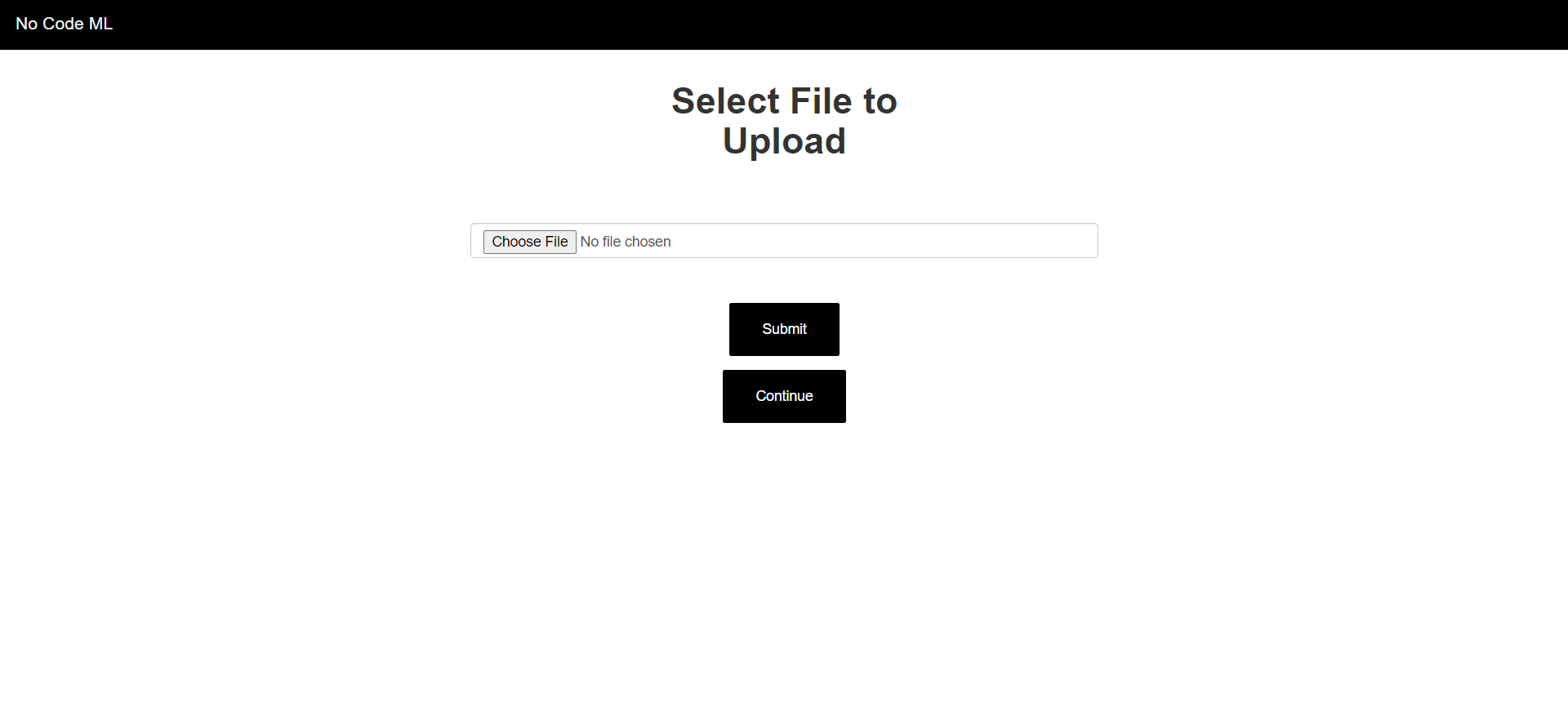
### 8.5 Quality Assurance

Quality Assurance is popularly known as QA Testing, is denied as an activity to ensure that an organization is providing the best possible product or service to customers. QA focuses on improving the processes to deliver Quality Products to the customer. An organization has to ensure that processes are effective as per the quality standards needed for software products.

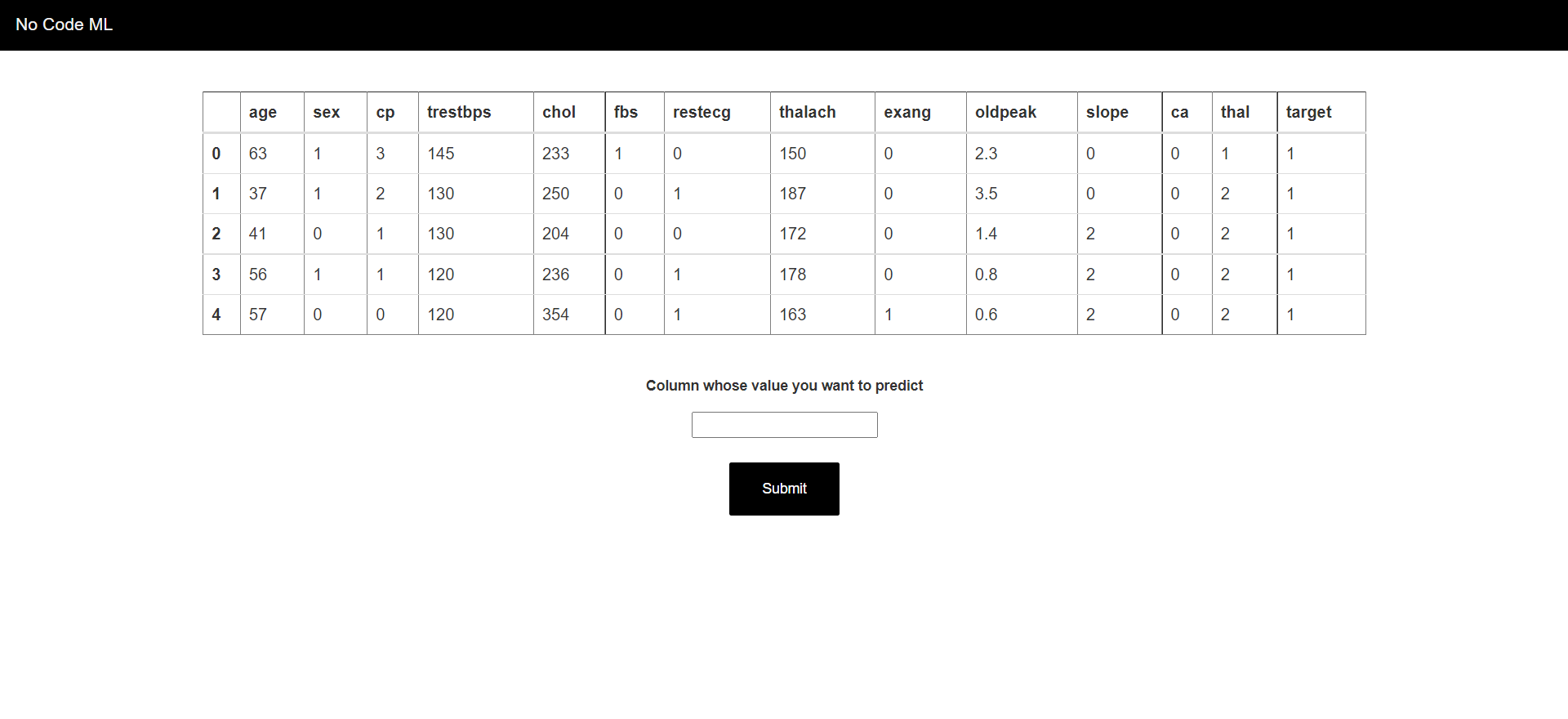
# Chapter 9

# 

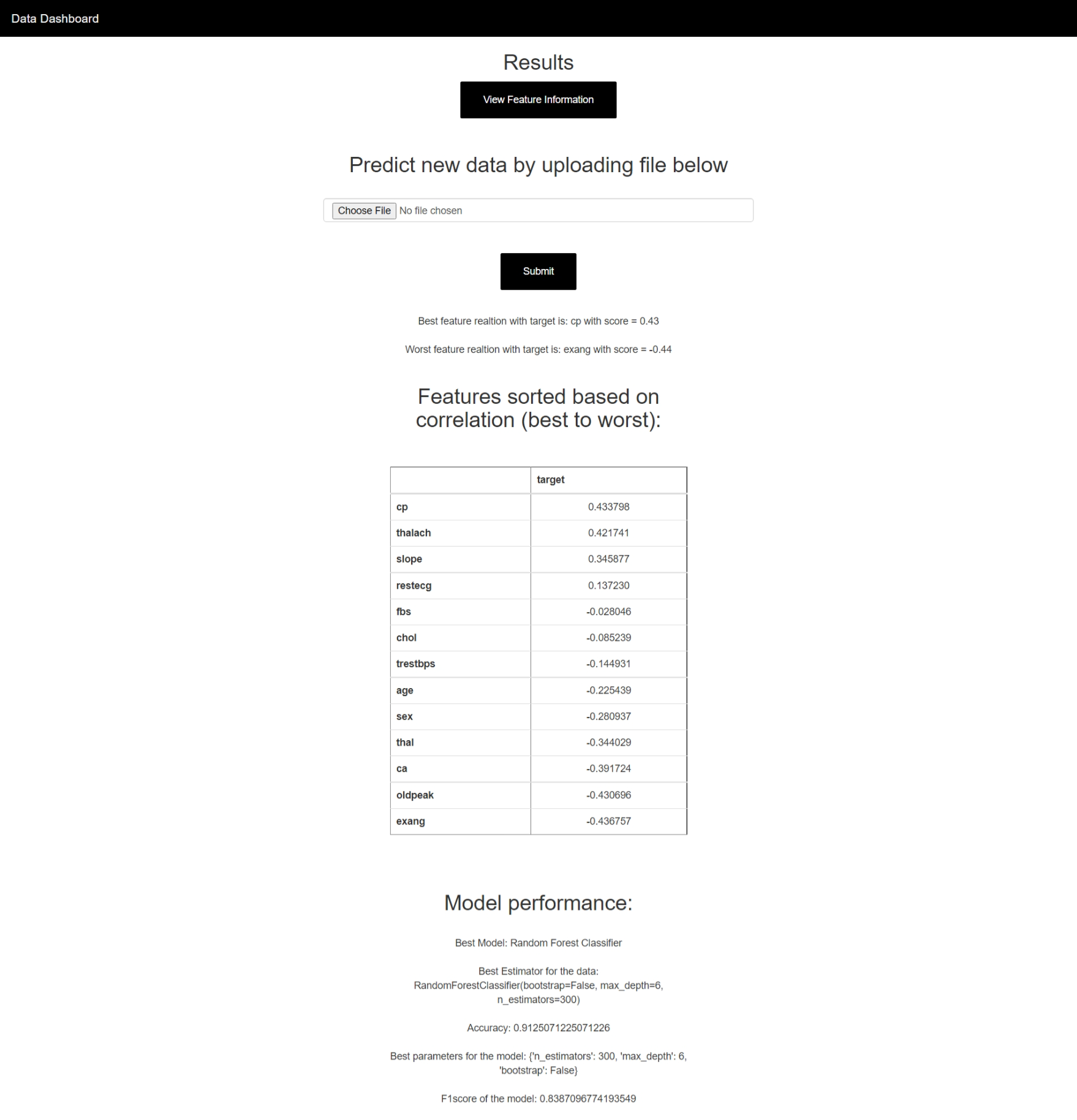
# Screen shots

****

**Fig 9.1. Landing Page**

****

**Fig 9.2 Column Selector for Prediction**

****

**Fig 9.3 Data Dashboard**

#### 9.1 Modules

This project has two modules

1. Regression : Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome' or 'response' variable) and one or more independent variables (often called 'predictors', 'covariates', 'explanatory variables' or 'features'). The most common form of regression analysis is linear regression, in which one finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion. For example, the method of ordinary least squares computes the unique line (or hyperplane) that minimizes the sum of squared differences between the true data and that line (or hyperplane). For specific mathematical reasons (see linear regression), this allows the researcher to estimate the conditional expectation (or population average value) of the dependent variable when the independent variables take on a given set of values. Less common forms of regression use slightly different procedures to estimate alternative location parameters (e.g., quantile regression or Necessary Condition Analysis) or estimate the conditional expectation across a broader collection of non-linear models (e.g., nonparametric regression).

Regression analysis is primarily used for two conceptually distinct purposes. First, regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Second, in some situations regression analysis can be used to infer causal relationships between the independent and dependent variables. Importantly, regressions by themselves only reveal relationships between a dependent variable and a collection of independent variables in a fixed dataset. To use regressions for prediction or to infer causal relationships, respectively, a researcher must carefully justify why existing relationships have predictive power for a new context or why a relationship between two variables has a causal interpretation. The latter is especially important when researchers hope to estimate causal relationships using observational data.

Regression models predict a value of the Y variable given known values of the X variables. Prediction within the range of values in the dataset used for model-fitting is known informally as interpolation. Prediction outside this range of the data is known as extrapolation. Performing extrapolation relies strongly on the regression assumptions. The further the extrapolation goes outside the data, the more room there is for the model to fail due to differences between the assumptions and the sample data or the true values.

2. Classification : Classification analysis is a data analysis task within data-mining, that identifies and assigns categories to a collection of data to allow for more accurate analysis. The classification method makes use of mathematical techniques such as decision trees, linear programming, neural network and statistics.

Classification analysis can be used to question, make a decision, or predict behavior through the use of an algorithm. It works by developing a set of training data which contains a certain set of attributes as well as the likely outcome. The job of the classification algorithm is to discover how that set of attributes reaches its conclusion.

There are two steps in the construction of a classification model.

Learning Step – this is where different algorithms are used to build a classifier by making the model learn using the training set available. The model has to be trained for the prediction of accurate results.

Classification Step: this is where the model used to predict class labels, tests the constructed model on test data. Which in turn estimates the accuracy of the classification rules.

When we apply cluster analysis to a dataset, we let the values of the variables that were measured tell us if there is any structure to the observations in the data set, by choosing a suitable metric and seeing if groups of observations that are all close together can be found. If we have an auxiliary variable (like the country of origin from the cars example), it may be interesting to see if the natural clustering of the data corresponds to this variable, but it's important to remember that the idea of clustering is just to see if any groups form naturally, not to see if we can actually figure out which group an observation belongs to based on the values of the variables that we have.

When the true goal of our data analysis is to be able to predict which of several non-overlapping groups an observation belongs to, the techniques we use are known as classification techniques. We'll take a look at three classification techniques: kth nearest neighbor classification, linear discriminant analysis, and recursive partitioning.

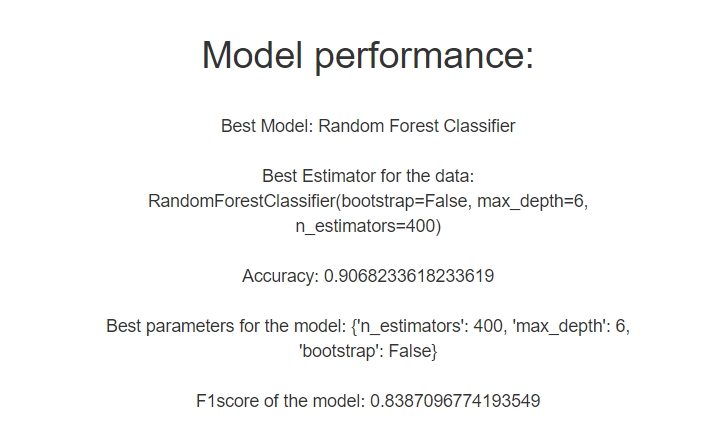
# Chapter 10

# Result Performance and Analysis

The outcomes and the results of the processed dataset is summarized in the form of a data dashboard.   
The data dashboards consist of multiple sections that summarizes all the inferences and observations that were made by the model in an easy to view and consume manner.

For feature information we create a Profile Report that helps the user in basic exploratory data analysis, apart from feature information we also share the best and the worst features based on their correlation with the target variable.

We also let the user an insight into the model performance that was applied on the dataset, for the test dataset that was applied the following was the outcome:



This model is also used to help the user in making future predictions.

# Chapter 11

# Conclusion

We demonstrated that our new AutoML system performs favourably against the previous state of the art in AutoML and that our meta-learning and ensemble improvements for AutoML yield further efficiency and robustness. In this paper, we did not evaluate the use of No code ML for interactive machine learning with an expert in the loop and weeks of CPU power, but we hope in the future to run this utilizing the power of cloud computing. As such, we believe that No Code ML is a promising system for use by both machine learning novices and experts.

Our system also has some shortcomings, which we would like to remove in future work. As one

For example, we have not yet tackled Natural language or semi-supervised problems. Most importantly, though, the focus on regression and classification implied a focus on small to medium-sized datasets, and an obvious direction for future work will be to apply our methods to modern deep learning systems that yield state-of-the-art performance on large datasets; we expect that in that domain especially automated ensemble construction will lead to tangible performance improvements over Bayesian optimization.

At the end of this project, we see the potential of an easy to use the no-code application in the business intelligence field for future predictions and also a detailed analysis of the past performance.

# Chapter 12

# Bibliography

1. <https://www.groundai.com/project/survey-on-automated-machine-learning/1#bib.bib142>
2. <https://www.groundai.com/project/survey-on-automated-machine-learning/1#bib.bib165>
3. <https://www.groundai.com/project/survey-on-automated-machine-learning/1#bib.bib40>
4. <https://www.groundai.com/project/survey-on-automated-machine-learning/1#bib.bib26>
5. <https://www.groundai.com/project/survey-on-automated-machine-learning/1#bib.bib48>
6. <https://stackoverflow.com/>
7. <https://www.youtube.com/>