**Freelancing Customer Segmentation**

# High level Design Document

**Version 1.0**

RESTRICTED DISTRIBUTION

The information is standard Company Confidential, but due to its sensitivity, it has restricted distribution and viewing within iNeuron.

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 3/11/2020 | 1.0 | Initial Draft |  |

Contributors

The content of this document has been authored with the combined input of the following group of key individuals.

|  |  |
| --- | --- |
| **Name** | **Section Worked Upon** |
|  | Initial Draft |

Document Classification

|  |  |
| --- | --- |
| **Classification** | Company Confidential |
| **Definition** | Information is Group confidential and needs to be protected |
| **Context** | Where the loss of information confidentiality would result in significant harm to the interests of the organisation, financial loss, embarrassment or loss of information |

Contents

[1 Technical Design Document 2](#_Toc55920495)

[1. Introduction 4](#_Toc55920496)

[High level objectives 5](#_Toc55920497)

[2 Workflow Overall 6](#_Toc55920498)

[Application Flow 6](#_Toc55920499)

[Exception Scenarios Overall 8](#_Toc55920500)

# Introduction

The goal here is to build an end to end automated Machine Learning solution for customer segmentation where the user will only give the data and the result will be the best performing hyper tuned Machine Learning model.

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

The High level design document gives a design blueprint of the freelancing customer segmentation project. This document communicates the technical details of the solution proposed.

In addition, this document also captures the different workflows involved to build the solution, exceptions in the workflows and any assumptions that have been considered.

Once agreed as the basis for the building of the project, the flowchart and assumptions will be used as a platform from which the solution will be designed.

Changes to this business process may constitute a request for change and will be subject to the agreed agility program change procedures.

**Note: All the code will be written in python version 3.7**

## High level objectives

The high-level objectives are:

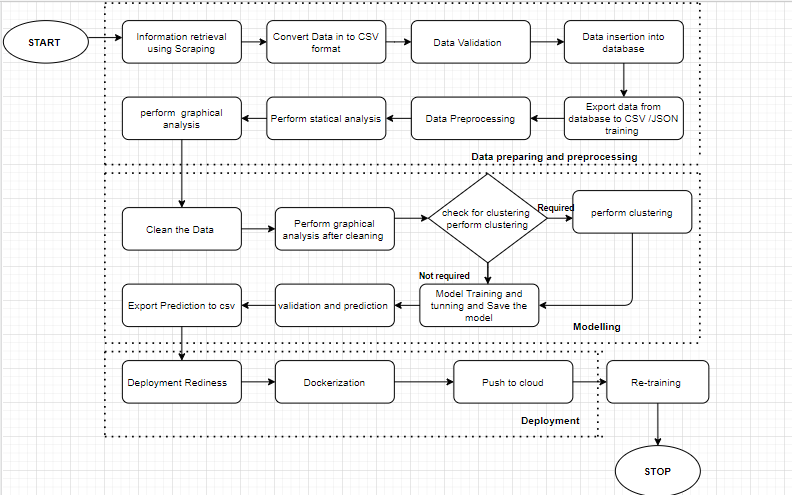
1. Enable reading/loading of data from the various sources and convert them into panda’s data frame (details mentioned in the Data Ingestion Section).
2. Enable reading various file formats and convert them into panda’s data frame (details mentioned in the Data Ingestion Section).
3. Give user the filter option to specify feature and target columns.
4. Perform statistical analytics of the data and prepare a table for the analysis
5. Perform graphical analysis for the data and Showcase the results (graphs) on the screen.
6. Perform data cleaning operation with all the steps required and showcase a report on screen.
7. After data cleaning showcase the graphical analysis once again for comparison.
8. Check whether clustering is required or not.
9. Choose the appropriate ML model for training.
10. Perform model Tuning.
11. Create a list of top 3 models and show multiple metrics for them.
12. Give option for prediction.
13. Give options for docker container creation.
14. Give option for automatic cloud deployment.

**Phase 1:** Create Pypi packages

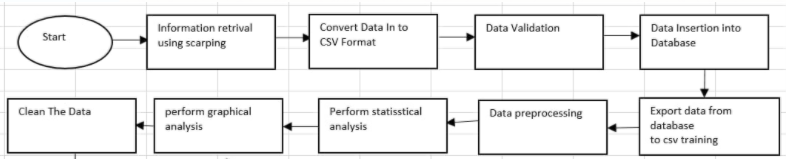
**Phase 2:** Create UI

# Workflow Overall

# Application Flow



Data preparing and Pre-processing:



1. **Information Retrieval using Scrapping**: This is an initial step which is crucial to collect the data. As a part of this module, we need to collect the data based on identified features w.r.t employer, employees, and Job description. We have identified these features based on Competitor mode of operation, Competitors Business Research, Competitors Leaders Research, Information source analysis.
2. **Convert Data into CSV format:** The information which we have scrapped we have converted that data into CSV format.
3. **Store data into tabular format:** As a next step, we need to fetch the data from the csv files and store the relative data into following tables.
   1. Employers\_Information\_tab – This table consists the information about employers, which are responsible to posting the job’s into the website.
   2. Employees\_Information\_tab – This table consists the information about employee’s information, which are applying for the jobs.
   3. Job\_description\_tab- This table consists the information about job information based on technology, location etc.
4. **Data Pre-processing: -** In this step, application will fetch the data from database and perform the data pre-processing techniques (cleaning and organizing), perform statistical analysis to identify the relationship between data and categorized in to define categories. This data is suitable for a building and training Machine Learning models.

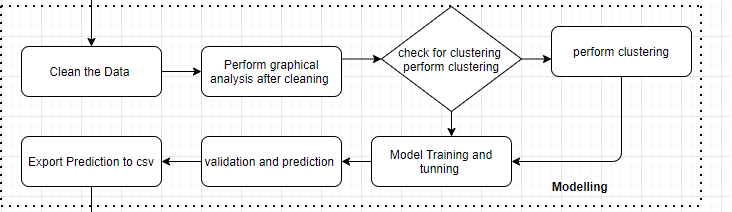
**Technology stack-**

* **Python -** The three core Python libraries used for this data set Numpy , Pandas, Matplotlib
* **Java script**
* **Graph database**
* **Database**

**Modelling:**

Feature engineering improves the quality of performance of machine learning algorithms. It is a process of using domain knowledge to extract features from row data via data mining technique. By scraping the data, we extracted some of the feature which is important to build a good model.

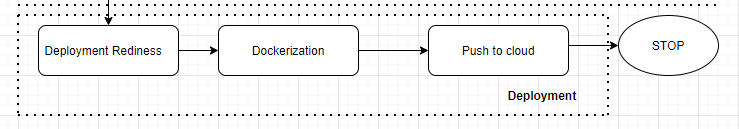
**Model Training**:



A model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data.

* K-Means Clustering
* Density-Based Spatial Clustering of Applications with Noise
* Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)
* Agglomerative Hierarchical Clustering
* OPTICS Clustering
* Affinity Propagation
* Birch Clustering

**Deployment**:



Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data.

Some deployment platform Google Cloud Platform (GCP) is one of the primary options for cloud-based deployment of ML models, along with others such as AWS, Microsoft Azure,etc

- Training a machine learning model on a local system.

- Wrapping the inference logic into a flask application.

- Using docker to containerize the flask application.

- Hosting the docker container on an AWS ec2 instance and consuming the web-service.

## Exception Scenarios Overall

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| User gives Wrong Data Source | Give proper error message | Ask the user to re-enter the details |
| User gives corrupted data | Give proper error message |  |
| If the cluster contains only one class | No error message required | Handle this exception internally. User doesn’t know. |
| Deployment credentials are wrong | Give proper error message | Ask for the details to be entered again |

# 