AutoNeuro

# Technical 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 |
| 25th June 2020 | 1.0 | Initial Draft | Virat Sagar |

Contributors

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

|  |  |
| --- | --- |
| Name | Section Worked Upon |
| Virat Sagar | 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](#_Toc44177265)

[1. Introduction 5](#_Toc44177266)

[High level objectives 5](#_Toc44177267)

[2 Workflow Overall 6](#_Toc44177268)

[Application Flow 6](#_Toc44177269)

[Exception Scenarios Overall 6](#_Toc44177270)

[3 Workflow Data Ingestion and File Conversion 8](#_Toc44177271)

[3.1 Technical solution design 10](#_Toc44177272)

[3.2 Exceptions Scenarios 10](#_Toc44177273)

[4 Stats Based EDA 11](#_Toc44177274)

[4.1 Steps 11](#_Toc44177275)

[4.2 Technical solution design 11](#_Toc44177276)

[4.3 Exceptions Scenarios Module Wise 11](#_Toc44177277)

[5 Graph-Based EDA 12](#_Toc44177278)

[5.1 Technical solution design 12](#_Toc44177279)

[5.2 Exceptions Scenarios Module Wise 13](#_Toc44177280)

[6 Library Based Utils 14](#_Toc44177281)

[6.1 Technical solution design 14](#_Toc44177282)

[6.2 Exceptions Scenarios Module Wise 14](#_Toc44177283)

[7 Data Transformers( Pre-processing steps) 15](#_Toc44177284)

[7.1 Technical solution design 15](#_Toc44177285)

[7.2 Exceptions Scenarios Module Wise 15](#_Toc44177286)

[8 ML Model Selection 16](#_Toc44177287)

[8.1 Technical solution design 16](#_Toc44177288)

[8.2 Exceptions Scenarios Module Wise 16](#_Toc44177289)

[9 Model Tuning and Optimization 17](#_Toc44177290)

[9.1 Technical solution design 18](#_Toc44177291)

[9.2 Exceptions Scenarios Module Wise 18](#_Toc44177292)

[10 Testing Modules 19](#_Toc44177293)

[10.1 Technical solution design 19](#_Toc44177294)

[10.2 Exceptions Scenarios Module Wise 20](#_Toc44177295)

[11 Prediction Pipeline 21](#_Toc44177296)

[11.1 Technical solution design 22](#_Toc44177297)

[11.2 Exceptions Scenarios Module Wise 23](#_Toc44177298)

[12 Deployment Strategy 24](#_Toc44177299)

[12.1 Technical solution design 25](#_Toc44177300)

[12.2 Exceptions Scenarios Module Wise 26](#_Toc44177301)

[13 Monitoring 27](#_Toc44177302)

[13.1 Technical solution design 27](#_Toc44177303)

[13.2 Exceptions Scenarios Module Wise 27](#_Toc44177304)

[14 Logging 28](#_Toc44177305)

[14.1 Technical solution design 28](#_Toc44177306)

[14.2 Exceptions Scenarios Module Wise 28](#_Toc44177307)

[15 Hardware Requirements 29](#_Toc44177308)

[15.1 Requirements for model training 29](#_Toc44177309)

[15.2 Requirements for model testing 29](#_Toc44177310)

# Introduction

The goal here is to build an end to end automated Machine Learning solution where the user will only give the data and select the type of problem, and the result will be the best performing hyper tuned Machine Learning model. The user will also get privileges to choose the deployment options.

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 technical design document gives a design blueprint of the Autoneuro 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:

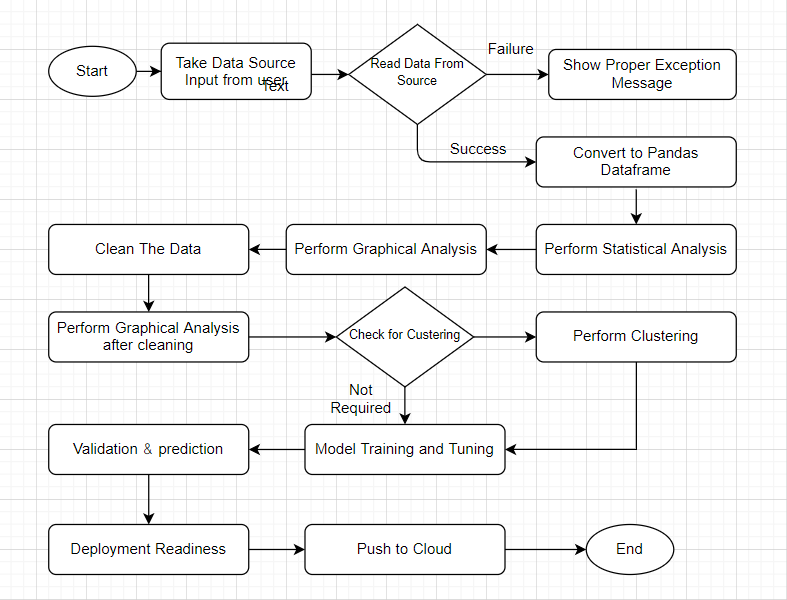
1. Enable reading/loading of data from the various sources and convert them into pandas dataframe(details mentioned in the Data Ingestion Section).
2. Enable reading various file formats and convert them into pandas dataframe(details mentioned in the Data Ingestion Section).
3. Give user the option to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

**Phase 1:** Create Pypi packages

**Phase 2:** Create UI

# Workflow Overall

## Application Flow



## 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 |  |
| User gives wrong null symbol | Give proper error message | Ask the user to provide correct symbol used for missing values |
| 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 |

# Workflow Data Ingestion and File Conversion

**Data Sources:**

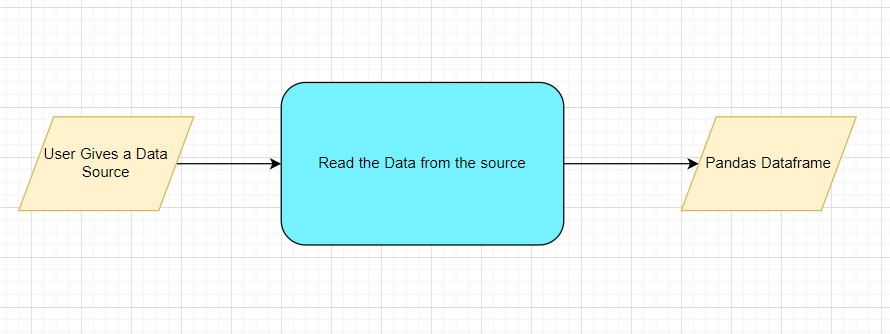
**Phase 1:**

|  |  |
| --- | --- |
| Data Connector Utils | File Conversion Utils |
| [Microsoft Access](https://help.tableau.com/current/pro/desktop/en-us/examples_access.htm) | CSV & text files, PDF |
| [Spatial File](https://help.tableau.com/current/pro/desktop/en-us/examples_spatial_files.htm) | JSON |
| [Statistical File](https://help.tableau.com/current/pro/desktop/en-us/examples_statfile.htm) | HTML |
| [Tableau Server or Tableau Online](https://help.tableau.com/current/pro/desktop/en-us/examples_tableauserver.htm) | Excel files |
| [Actian Matrix](https://help.tableau.com/current/pro/desktop/en-us/examples_actianmatrix.htm) | OpenDocument Spreadsheets |
| [Actian Vectorwise](https://help.tableau.com/current/pro/desktop/en-us/examples_vectorwise.htm) | Binary Excel (.xlsb) files |
| [Alibaba AnalyticDB for MySQL](https://help.tableau.com/current/pro/desktop/en-us/examples_alibaba_analyticdb.htm) | Clipboard |
| [Alibaba Data Lake Analytics](https://help.tableau.com/current/pro/desktop/en-us/examples_alibaba_data_lake_analytics.htm) | Pickling |
| [Alibaba MaxCompute](https://help.tableau.com/current/pro/desktop/en-us/examples_alibaba_maxcompute.htm) | msgpack |
| [Amazon Athena](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonathena.htm) | HDF5 (PyTables) |
| [Amazon Aurora for MySQL](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonaurora.htm) | Feather |
| [Amazon EMR Hadoop Hive](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonemr.htm) | Parquet |
| [Amazon Redshift](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonredshift.htm) | ORC |
| [Anaplan](https://help.tableau.com/current/pro/desktop/en-us/examples_anaplan.htm) | Google BigQuery |
| [Apache Drill](https://help.tableau.com/current/pro/desktop/en-us/examples_apachedrill.htm) | Stata format |
| [Aster Database](https://help.tableau.com/current/pro/desktop/en-us/examples_asterdata.htm) | SAS formats |
| [Azure SQL Synapse Analytics](https://help.tableau.com/current/pro/desktop/en-us/examples_azure_sql_dw.htm) | SPSS formats |
| [Box](https://help.tableau.com/current/pro/desktop/en-us/examples_box.htm) | Other file formats |
| [Cloudera Hadoop](https://help.tableau.com/current/pro/desktop/en-us/examples_hadoop.htm) | Performance considerations |
| [Databricks](https://help.tableau.com/current/pro/desktop/en-us/examples_databricks.htm) |  |
| [Denodo](https://help.tableau.com/current/pro/desktop/en-us/examples_denodo.htm) |  |
| [Dropbox](https://help.tableau.com/current/pro/desktop/en-us/examples_dropbox.htm) |  |
| [Esri ArcGIS Server](https://help.tableau.com/current/pro/desktop/en-us/examples_esri.htm) |  |
| [Exasol](https://help.tableau.com/current/pro/desktop/en-us/examples_exasolution.htm) |  |
| [Firebird 3](https://help.tableau.com/current/pro/desktop/en-us/examples_firebird.htm) |  |
| [Google Ads](https://help.tableau.com/current/pro/desktop/en-us/examples_googleads.htm) |  |
| [Google Analytics](https://help.tableau.com/current/pro/desktop/en-us/examples_googleanalytics.htm) |  |
| [Google BigQuery](https://help.tableau.com/current/pro/desktop/en-us/examples_googlebigquery.htm) |  |
| [Google Cloud SQL](https://help.tableau.com/current/pro/desktop/en-us/examples_googlecloudsql.htm) |  |
| [Google Drive](https://help.tableau.com/current/pro/desktop/en-us/examples_googledrive.htm) |  |
| [Google Sheets](https://help.tableau.com/current/pro/desktop/en-us/examples_googlesheets.htm) |  |
| [Hortonworks Hadoop Hive](https://help.tableau.com/current/pro/desktop/en-us/examples_hortonworkshadoop.htm) |  |
| [IBM BigInsights](https://help.tableau.com/current/pro/desktop/en-us/examples_biginsights.htm) |  |
| [IBM DB2](https://help.tableau.com/current/pro/desktop/en-us/examples_db2.htm) |  |
| [IBM PDA (Netezza)](https://help.tableau.com/current/pro/desktop/en-us/examples_netezza.htm) |  |
| [Impala](https://help.tableau.com/current/pro/desktop/en-us/examples_impala.htm) |  |
| [Intuit QuickBooks Online](https://help.tableau.com/current/pro/desktop/en-us/examples_quickbooksonline.htm) |  |
| [Kognitio](https://help.tableau.com/current/pro/desktop/en-us/examples_kognitio.htm) |  |
| [Kyvos](https://help.tableau.com/current/pro/desktop/en-us/examples_kyvos.htm) |  |
| [LinkedIn Sales Navigator](https://help.tableau.com/current/pro/desktop/en-us/examples_linkedin_sales_navigator.htm) |  |
| [MapR Hadoop Hive](https://help.tableau.com/current/pro/desktop/en-us/examples_maprhadoop.htm) |  |
| [MariaDB](https://help.tableau.com/current/pro/desktop/en-us/examples_mariadb.htm) |  |
| [Marketo](https://help.tableau.com/current/pro/desktop/en-us/examples_marketo.htm) |  |
| [MarkLogic](https://help.tableau.com/current/pro/desktop/en-us/examples_marklogic.htm) |  |
| [MemSQL](https://help.tableau.com/current/pro/desktop/en-us/examples_memsql.htm) |  |
| [Microsoft Analysis Services](https://help.tableau.com/current/pro/desktop/en-us/examples_msas.htm) |  |
| [Microsoft PowerPivot](https://help.tableau.com/current/pro/desktop/en-us/examples_powerpivot.htm) |  |
| [Microsoft SQL Server](https://help.tableau.com/current/pro/desktop/en-us/examples_sqlserver.htm) |  |
| [MonetDB](https://help.tableau.com/current/pro/desktop/en-us/examples_monetdb.htm) |  |
| [MongoDB BI Connector](https://help.tableau.com/current/pro/desktop/en-us/examples_mongodb.htm) |  |
| [MySQL](https://help.tableau.com/current/pro/desktop/en-us/examples_mysql.htm) |  |
| [OData](https://help.tableau.com/current/pro/desktop/en-us/examples_odata.htm) |  |
| [OneDrive](https://help.tableau.com/current/pro/desktop/en-us/examples_onedrive.htm) |  |
| [Oracle](https://help.tableau.com/current/pro/desktop/en-us/examples_oracle.htm) |  |
| [Oracle Eloqua](https://help.tableau.com/current/pro/desktop/en-us/examples_eloqua.htm) |  |
| [Oracle Essbase](https://help.tableau.com/current/pro/desktop/en-us/examples_essbase.htm) |  |
| [Pivotal Greenplum](https://help.tableau.com/current/pro/desktop/en-us/examples_greenplum.htm) |  |
| [PostgreSQL](https://help.tableau.com/current/pro/desktop/en-us/examples_postgresql.htm) |  |
| [Presto](https://help.tableau.com/current/pro/desktop/en-us/examples_presto.htm) |  |
| [Progress OpenEdge](https://help.tableau.com/current/pro/desktop/en-us/examples_progress.htm) |  |
| [Qubole Presto](https://help.tableau.com/current/pro/desktop/en-us/examples_qubole.htm) |  |
| [Salesforce](https://help.tableau.com/current/pro/desktop/en-us/examples_salesforce.htm) |  |
| [Splunk](https://help.tableau.com/current/pro/desktop/en-us/examples_splunk.htm) |  |
| [SAP HANA](https://help.tableau.com/current/pro/desktop/en-us/examples_saphana.htm) |  |
| [SAP NetWeaver Business Warehouse](https://help.tableau.com/current/pro/desktop/en-us/examples_sapbw.htm) |  |
| [SAP Sybase ASE](https://help.tableau.com/current/pro/desktop/en-us/examples_sybasease.htm) |  |
| [SAP Sybase IQ](https://help.tableau.com/current/pro/desktop/en-us/examples_sybaseiq.htm) |  |
| [ServiceNow ITSM](https://help.tableau.com/current/pro/desktop/en-us/examples_servicenow.htm) |  |
| [SharePoint Lists](https://help.tableau.com/current/pro/desktop/en-us/examples_sharepoint_lists.htm) |  |
| [Snowflake](https://help.tableau.com/current/pro/desktop/en-us/examples_snowflake.htm) |  |
| [Spark SQL](https://help.tableau.com/current/pro/desktop/en-us/examples_sparksql.htm) |  |
|  |  |
|  |  |
|  |  |
|  |  |
| [Connector Plugin](https://help.tableau.com/current/pro/desktop/en-us/examples_connector_sdk.htm) |  |
| [Web Data Connector](https://help.tableau.com/current/pro/desktop/en-us/examples_web_data_connector.htm) |  |
| [Other Databases (JDBC)](https://help.tableau.com/current/pro/desktop/en-us/examples_otherdatabases_jdbc.htm) |  |
| [Other Databases (ODBC)](https://help.tableau.com/current/pro/desktop/en-us/examples_otherdatabases.htm) |  |

**Phase 2:**

|  |  |
| --- | --- |
| Data Connector Utils | File Conversion Utils |
| [Spatial File](https://help.tableau.com/current/pro/desktop/en-us/examples_spatial_files.htm) | OpenDocument Spreadsheets |
| [Statistical File](https://help.tableau.com/current/pro/desktop/en-us/examples_statfile.htm) |  |
| [Tableau Server or Tableau Online](https://help.tableau.com/current/pro/desktop/en-us/examples_tableauserver.htm) |  |
| [Actian Matrix](https://help.tableau.com/current/pro/desktop/en-us/examples_actianmatrix.htm) |  |
| [Teradata OLAP Connector](https://help.tableau.com/current/pro/desktop/en-us/examples_teradata_olap.htm) |  |
| [TIBCO Data Virtualization (Cisco Information Server)](https://help.tableau.com/current/pro/desktop/en-us/examples_ciscoinfoserver.htm) |  |
| [Vertica](https://help.tableau.com/current/pro/desktop/en-us/examples_vertica.htm) |  |
| [Teradata](https://help.tableau.com/current/pro/desktop/en-us/examples_teradata.htm) |  |

## Technical solution design



## Method Definitions

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataGetter** |  |
| Method Name | read\_data\_from\_csv |  |
|  | Method Description | This method will be used to read data from a csv file or a flat file |
|  | Input parameter names | self,file\_name, header,names, use\_cols, separator |
|  | Input Parameter Description | file\_name: name of the file to be read  header: Row number(s) to be used as column names  names : array-like, optional  List of column names to use. If file contains no header row, then you  should explicitly pass ``header=None``.  Use\_cols: To load a subset of columns  Separator: Delimiter to use |
|  | ouptput | A pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | read\_data\_from\_json |  |
|  | Method Description | This method will be used to read data from a json file. |
|  | Input parameter names | self,file\_name |
|  | Input Parameter Description | file\_name: name of the file to be read |
|  | ouptput | A pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | read\_data\_from\_html |  |
|  | Method Description | This method will be used to read data from an HTML web page |
|  | Input parameter names | self,url |
|  | Input Parameter Description | url: URL of the HTML page to be read. |
|  | ouptput | A pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | read\_data\_from\_excel |  |
|  | Method Description | This method will be used to read data from an MS Excel File |
|  | Input parameter names | self,file\_name,sheet\_name, header,names, use\_cols, separator |
|  | Input Parameter Description | file\_name: name of the file to be read  sheet\_name: Lists of strings/integers are used to request  multiple sheets. Specify None to get all sheets.  header: Row number(s) to be used as column names  names : array-like, optional  List of column names to use. If file contains no header row, then you  should explicitly pass ``header=None``.  Use\_cols: To load a subset of columns  Separator: Delimiter to use |
|  | ouptput | A pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | Connect\_to\_sqldb |  |
|  | Method Description | This method will be used to connect to a SQL Databases |
|  | Input parameter names | self,host,port, username, password |
|  | Input Parameter Description | host: the server hostname/IP where the DB server is hosted  Port: the port at which the DB Server is running  username: The username to connect to the DB server  password: The password to connect to the DB server |
|  | ouptput | A DB connection object |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | read\_data\_from\_sqldb |  |
|  | Method Description | This method will be used to read data from SQL Databases |
|  | Input parameter names | self,db\_name,host,port, username, password, schema\_name,query\_string |
|  | Input Parameter Description | db\_name: For example, SQL, MySQL, SQLLite etc.  host: the server hostname/IP where the DB server is hosted  Port: the port at which the DB Server is running  username: The username to connect to the DB server  password: The password to connect to the DB server  schema\_name: The name of the DB schema the user wants to connect to.  query\_string: the query to be executed to load the data |
|  | ouptput | A Pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | read\_data\_from\_mongdb |  |
|  | Method Description | This method will be used to read data from Mongo DB |
|  | Input parameter names | self,host,port, username, password, db\_name,collection\_name, query\_string |
| ‘ | Input Parameter Description | host: the server hostname/IP where the DB server is hosted  Port: the port at which the DB Server is running  username: The username to connect to the DB server  password: The password to connect to the DB server  db\_name: The name of the database  collection\_name: The name of the collection the user wants to connect to.  query\_string: the query to be executed to load the data |
|  | ouptput | A Pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

## Exceptions Scenarios

|  |  |  |
| --- | --- | --- |
| **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 |  |

# Data Profiling

After reading the data, automatically the following details should be shown:

1. The number of rows
2. The number of columns
3. Number of missing values per column and their percentage
4. Total missing values and it’s percentage
5. Number of categorical columns and their list
6. Number of numerical columns and their list
7. Number of duplicate rows
8. Number of columns with zero standard deviation and their list
9. Size occupied in RAM

## Method Definition

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataProfiler** |  |
| Method Name | get\_data\_profile |  |
|  | Method Description | This method will be used to give various insighst about data. |
|  | Input parameter names | self, dataframe |
|  | Input Parameter Description | dataframe: the inpt data just loaded from source |
|  | ouptput | 1. The number of rows 2. The number of columns 3. Number of missing values per column and their percentage 4. Total missing values and it’s percentage 5. Number of categorical columns and their list 6. Number of numerical columns and their list 7. Number of duplicate rows 8. Number of columns with zero standard deviation and their list 9. Size occupied in RAM |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

# Stats Based EDA

## Steps

**MVP**

OLS

VIF

Correlation

**Phase1:**

Column contributions/ importance

Annova Test

Chi Square test

Z test

T -test

Weight of Evidence

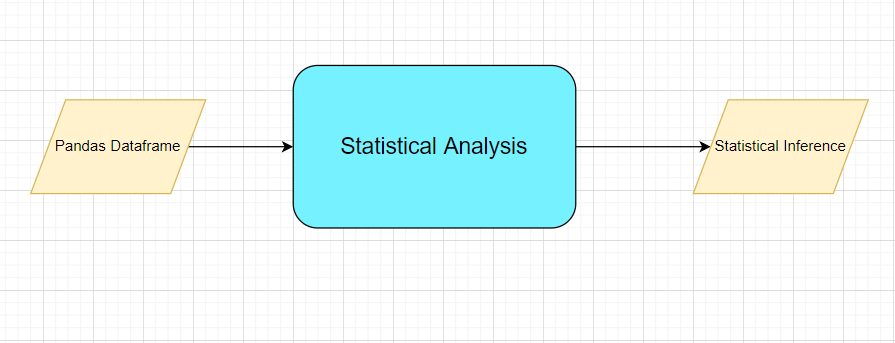
F – Test

**Phase 2:**

Seasonality

Stationary Data

## Technical solution design



## Method Definitions

|  |  |  |
| --- | --- | --- |
| **Class Name** | **SatisticalDataAnalyser** |  |
| Method Name | get\_correlation |  |
|  | Method Description | This method will be used to get correlation coefficient across all variables in a dataset and remove variables with correlation coefficient value greater than 0.60 (by default) |
|  | Input parameter names | self, dataframe, threshold |
|  | Input Parameter Description | dataframe: the input data loaded from the source  threshold: threshold value for removing highly correlated variables. By default, use 0.60 |
|  | ouptput | Multicollinearity free pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | get \_ols\_summary |  |
|  | Method Description | This method will be used to get the OLS summary of the dataset. The variables having lower p-value will be kept and others will be dropped |
|  | Input parameter names | self, dataframe |
|  | Input Parameter Description | dataframe: the input data loaded from the source |
|  | ouptput | OLS Summary |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | get\_vif\_report |  |
|  | Method Description | This method will be used to get the VIF report of the dataset |
|  | Input parameter names | self, dataframe, target variable |
|  | Input Parameter Description | dataframe: the input data loaded from the source  target variable: target variable of the dataset which will be excluded while calculating VIF of the dataset |
|  | ouptput | VIF report |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Column has mixed values(Integer & number) | Give proper error message | Ask the user to correct the data. |
| Not all values are numbers | Handle Internally | Convert categorical to numerical values |

# Graph-Based EDA

Create the following graphs:

**MVP:**

Correlation Heatmaps

Check for balance/imbalance

**Phase1:**

Count plots

Boxplot for outliers

Piecharts for categories

Geographical plots for scenarios

Line charts for trends

Barplots

Area Charts

KDE Plots

Stacked charts

Scatterplot

**Phase 2:**

Word maps

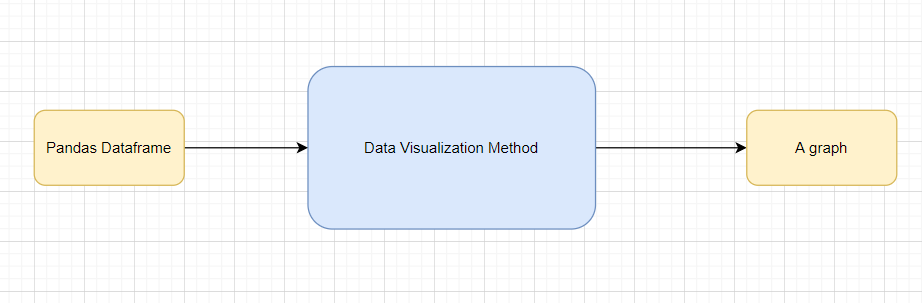
PACF

ACF

Add Custom controls sliders etc

**Note: We are going to use plotly for all the graphs.(** <https://plotly.com/python/>**)**

## Technical solution design



## Method Definitions

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataVisualization(Dummy)** |  |
| Method Name | read\_data\_from\_csv |  |
|  | Method Description | This method will be used to read data from a csv file or a flat file |
|  | Input parameter names | self,file\_name, header,names, use\_cols, separator |
|  | Input Parameter Description | file\_name: name of the file to be read  header: Row number(s) to be used as column names  names : array-like, optional  List of column names to use. If file contains no header row, then you  should explicitly pass ``header=None``.  Use\_cols: To load a subset of columns  Separator: Delimiter to use |
|  | ouptput | A pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong input to the methods | Handle Internally | Code should never give a wrong input |

# Data Transformers( Pre-processing steps)

**MVP:**

Null value handling

Categorical to numerical

Imbalanced data set handling

Handling columns with std deviation zero or below a threshold

Normalisation

PCA

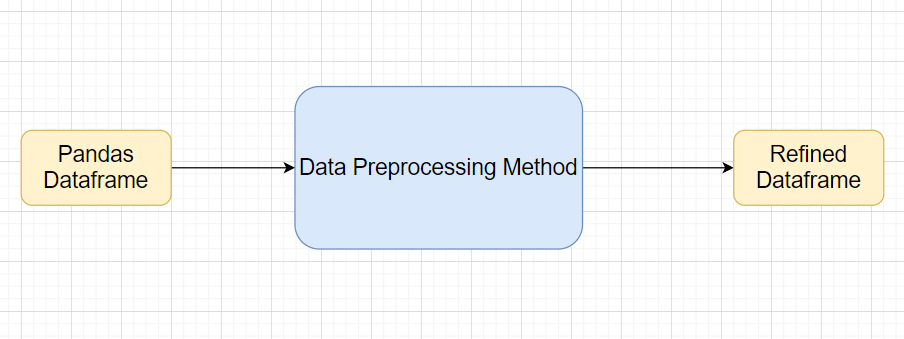
**Phase1:**

Outlier detection

Data Scaling/ Normalisation

Feature Selection: <https://scikit-learn.org/stable/auto_examples/index.html#feature-selection>

## Technical solution design



## Method Definitions

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataPreprocessor** |  |
| Method Name | impute\_missing\_values |  |
|  | Method Description | This method will be used to impute missing values in the dataframe |
|  | Input Parameter Names | self, data, strategy, impute\_val, missing\_vals, mv\_flag |
|  | Input Parameter Description | data : name of the input dataframe  strategy : strategy to be used for MVI (Missing Value Imputation)  --‘median’ : default for continuous variables, replaces missing value(s) with median of the concerned column  --‘mean’  --‘mode’ : default for categorical variables  --‘fixed’ : replaces all missing values with a fixed ‘explicitly specified’ value  impute\_val : None(default), can be assigned a value to be used for imputation in ‘fixed’ strategy  missing\_vals : None(default), a list/tuple of missing value indicators. By default, it considers only NaN as missing. Dictionary can be passed to consider different missing values for different columns in format – {col\_name:[val1,val2, …], col2: […]}  mv\_flag : None(default), can be passed list/tuple of columns as input for which it creates missing value flags |
|  | output | A DataFrame with missing values imputed |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | type\_conversion |  |
|  | Method Description | This method will be used to convert column datatype from numerical to categorical or vice-versa, if possible. |
|  | Input Parameter Names | self, dataset, cat\_to\_num, num\_to\_cat |
|  | Input Parameter Description | dataset : input DataFrame in which type conversion is needed  cat\_to\_num : None(default), list/tuple of variables that need to be converted from categorical to numerical  num\_to\_cat : None(default), list/tuple of variables to be converted from numerical to categorical |
|  | output | A DataFrame with column types changed as per requirement |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | remove\_imbalance |  |
|  | Method Description | This method will be used to handle unbalanced datasets(rare classes) through oversampling/ undersampling techniques |
|  | Input Parameter Names | self, data, threshold |
|  | Input Parameter Description | data: the input dataframe with target column.  threshold: the threshold of mismatch between the target values to perform balancing. |
|  | output | A balanced dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | remove\_columns\_with\_minimal\_variance |  |
|  | Method Description | This method drops any numerical column with standard deviation below specified threshold |
|  | Input Parameter Names | self, data, threshold |
|  | Input Parameter Description | data: input DataFrame in which we need to check std deviations  threshold : the threshold for std deviation below which we need to drop the columns |
|  | output | A DataFrame with numerical columns with low std dev dropped |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | normalize\_data |  |
|  | Method Description | This method will be used to do a standardization, normalisation, min-max scaling of numerical variables the input DataFrame |
|  | Input Parameter Names | self, data, strategy, mean, std |
|  | Input Parameter Description | data : input DataFrame in which transformation is to be applied  strategy : transformation to be used on the numerical columns  -- ‘normal’ : transforms data to std. normal distribution with mean=0 and std=1.  -- ‘standardize’ : standardizes data using mean and std specified  -- ‘minmax’ : does a min-max scaling for numerical columns  mean : 0(default), mean around which standardisation needs to be dome  std : 1(default), standard deviation that needs to be applied for transformation  \*further mathematical transformations(for instance:log, inverse) can also be included in strategy and an additional function parameter to take input function. |
|  | output | A DataFrame with all the numerical columns transformed as per requirement |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | pca |  |
|  | Method Description | This method will be used to do the Principal Component Analysis on input dataframe and select the most important components |
|  | Input Parameter Names | self, data, var\_explained |
|  | Input Parameter Description | data : input DataFrame in which pca is to be applied  var\_explained : 0.90(default), Total variation(0 to 1) that we want the selected variables to be able to explain |
|  | output | A DataFrame with original variables and its principal components. |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | get\_categorical\_encoding |  |
|  | Method Description | This method does categorical encoding and is largely dependent based on this below package:  <https://pypi.org/project/category-encoders/> |
|  | Input Parameter Names | Will depend on how this package is being used |
|  | output | A DataFrame with encoded features and the original categorical columns both. Original categorical columns can be dropped, if perceived necessary |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong parameters passed to the methods | Handle Internally | Code should never give a wrong input |

# ML Model Selection

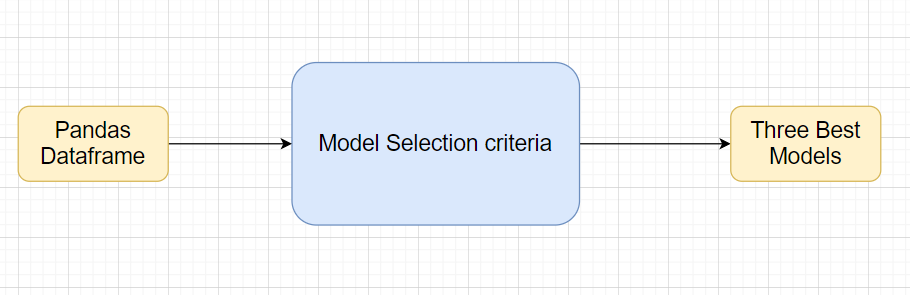
**MVP:**

3 Models—KNN, RandomForest, XGBoost

**Phase1:**

Model Selection criteria

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong parameters passed to the methods | Handle Internally | Code should never give a wrong input |

# Model Tuning and Optimization

**Note:** The data should have been divided into train and validation set before this.

Methods for hyper tuning all kinds of models.

**Regression:**

Linear Regression

Decision Tree

Random Forest

XG Boost

Support Vector Regressor

KNN Regressor

Model selection criteria:

MSE, RMSE, R squared, adjusted R squared

**Classification:**

Logistic Regression

Decision Tree

Random Forest

XG Boost

Support Vector Classifier

KNN Classifier

Naïve Baye’s

Model selection criteria:

Accuracy, AUC, Precision, Recall, F Beta

**Clustering:**

K-Means

Hierarchial

DBSCAN

**Phase 2:**

GLM

GAM (<https://www.statsmodels.org/stable/regression.html>)

Time Series

Anomaly Detection

Novelty Detection

Optics

Gaussian Mixtures

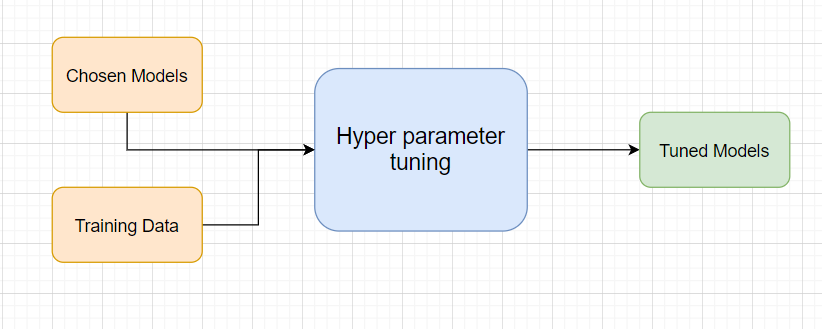
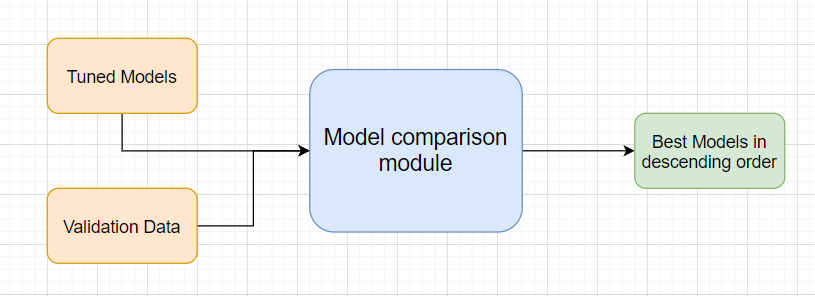
BIRCH

NLP

Deep Learning

Regularization modules if necessary

## Technical solution design

1. 
2. 

## Method Definitions

|  |  |  |
| --- | --- | --- |
| **Class Name** | **ModelTuner** |  |
| Method Name | get\_tuned\_knn\_model |  |
|  | Method Description | This method will be used to get the hypertuned KNN Model |
|  | Input parameter names | self,data |
|  | Input Parameter Description | Data: the training data |
|  | Hyperparameters to tune | **n\_neighbors:**Number of neighbors to use by default for [kneighbors](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html" \l "sklearn.neighbors.KNeighborsClassifier.kneighbors" \o "sklearn.neighbors.KNeighborsClassifier.kneighbors) queries.  **weights:** weight function used in prediction. Possible values:   * ‘uniform’ : uniform weights. All points in each neighborhood are weighted equally. * ‘distance’ : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.   **algorithm**{‘auto’, ‘ball\_tree’, ‘kd\_tree’, ‘brute’}, default=’auto’  Algorithm used to compute the nearest neighbors:  **leaf\_size:** int, default=30  Leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem.  **n\_jobs:** int, Keep it as -1 |
|  | ouptput | A hyper parameter tuned model object |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | get\_tuned\_random\_forest\_model |  |
|  | Method Description | This method will be used to get the hypertuned Random Forest Model |
|  | Input parameter names | self,data |
|  | Input Parameter Description | Data: the training data |
|  | Hyperparameters to tune | **Classifier🡪**  **n\_estimators:** The number of trees in the forest.  **criterion**{“gini”, “entropy”}, default=”gini”  The function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain.  **max\_depth**: int, default=None  The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.  **min\_samples\_split:** int or float, default=2  The minimum number of samples required to split an internal node:  **n\_jobs=** -1  **Regressor🡪**  **n\_estimators:** The number of trees in the forest.  **criterion**{“mse”, “mae”}, default=”mse”  The function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain.  **max\_depth**: int, default=None  The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.  **min\_samples\_split:** int or float, default=2  The minimum number of samples required to split an internal node: |
|  | ouptput | A hyper parameter tuned model object |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | get\_tuned\_xgboost\_model |  |
|  | Method Description | This method will be used to get the hypertuned XGBoost Model |
|  | Input parameter names | self,data |
|  | Input Parameter Description | Data: the training data |
|  | Hyperparameters to tune | eta [default=0.3, alias: learning\_rate]   * Step size shrinkage used in update to prevents overfitting. After each boosting step, we can directly get the weights of new features, and eta shrinks the feature weights to make the boosting process more conservative. * range: [0,1]   gamma [default=0, alias: min\_split\_loss]   * Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be. * range: [0,∞]   max\_depth [default=6]   * Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. 0 is only accepted in lossguided growing policy when tree\_method is set as hist and it indicates no limit on depth. Beware that XGBoost aggressively consumes memory when training a deep tree.   Objective: The objective function |
|  | ouptput | A hyper parameter tuned model object |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
|  |  |  |

# Testing Modules

Divide the training data itself into train and test sets

Use test data to have tests run on the three best models

Give the test report

1. R2 Score
2. Adjusted R2 score
3. MSE
4. Accuracy
5. Precision
6. Recall
7. F Beta
8. Cluster Purity
9. Silhouette score

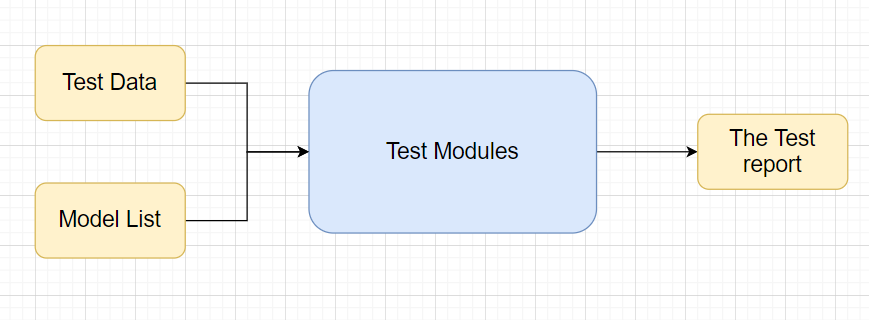
**Phase 2**

AIC

BIC

**Note**: Save the best model after validation is completed.

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Number of Parameters do not match | Handle internally | Check the test data creation and verify the columns |
| Only once class present in test data | Handle Internally |  |

# Prediction Pipeline

Use the existing data read modules

Use the existing pre-processing module

Load the model into memory

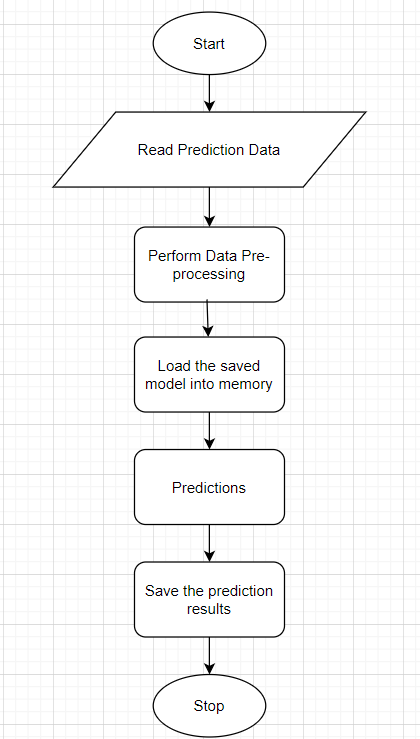
Do predictions

Store prediction results(show sample predictions)

Phase 2:

UI for predictions

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Columns don’t match in training and Prediction data | Show error message | The user enters the correct data |
|  |  |  |

# Deployment Strategy

Take the cloud name as input

Prepare the metadata files based on cloud

Phase 2:

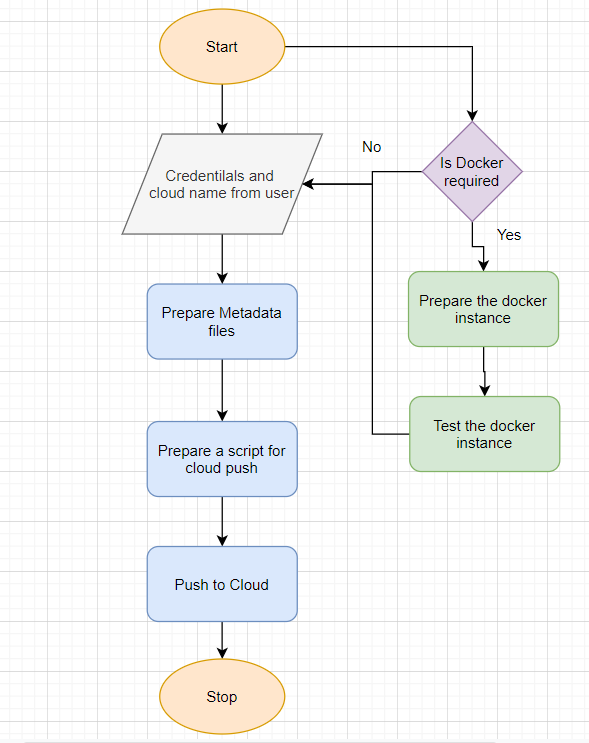
Accept the user credentials

Prepare a script file to push changes

Docker instance

Push of the docker instance to cloud

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong Cloud credentials | Show error message | The user enters the correct data |
| Docker instance not working | Show error message | Fix the error |
| Cloud push failed | Show the error | Make corrections to the metadata  files |
| Cloud app not starting |  | Ask the user for cloud logs for debugging |

# Monitoring

Phase 2

No. Of predictions for individual classes

No. of predictions (per day, per hour, per week etc.)

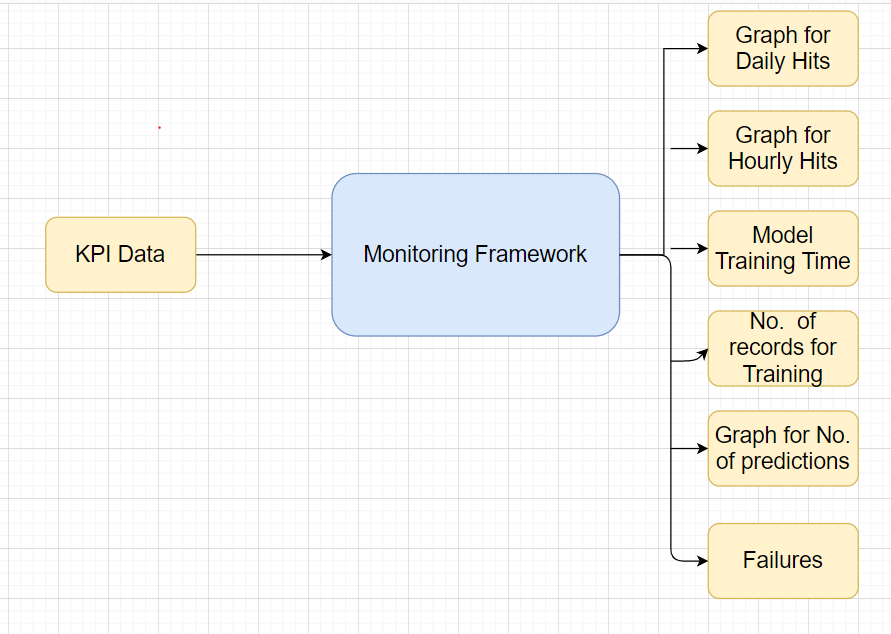
No. of hits

Training data size (number of rows)

Time spent in training

Failures

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
|  |  |  |

# Logging

Separate Folder for logs

Logging of every step

Entry to the methods

Exit from the methods with success/ failure message

Error message Logging

Model comparisons

Training start and end

Prediction start and end

Achieve asynchronous logging

**Phase 2:**

Options for Logging in DB

Options for Log Publish

## Technical solution design



## Common Logging Framework Code

|  |  |
| --- | --- |
| Class Name | App Logger |
| Method Name | log |
| Method Description | This method will be used for logging all the information to the file. |
| Input parameter names | self,file\_object, log\_message |
| Input Parameter Description | file\_object: the file where the logs will be written  log\_message: the message to be logged |
| ouptput | A log file with messages |

# from datetime import datetime class App\_Logger: def \_\_init\_\_(self): pass def log(self, file\_object, log\_message):“””This method will be used for logging all the information to the file.””” self.now = datetime.now() self.date = self.now.date() self.current\_time = self.now.strftime("%H:%M:%S") file\_object.write( str(self.date) + "/" + str(self.current\_time) + "\t\t" + log\_message +"\n")

## Exceptions Scenarios Module Wise

Ideally, the logging should never fail.

# Hardware Requirements

## Requirements for model training

The minimum configuration should be:

* 8 GB RAM
* 2 GB of Hard Disk Space
* Intel Core i5 Processor

## Requirements for model testing

The minimum configuration should be:

* 4 GB RAM
* 2 GB of Hard Disk Space
* Intel Core i5 Processor

# Sample code and standard to be followed:

Sample Code:

class Data\_Getter:  
 *"""  
 This class shall be used for obtaining the data from the source for training.  
  
 Written By: iNeuron Intelligence  
 Version: 1.0  
 Revisions: None  
  
 """* def \_\_init\_\_(self, file\_object, logger\_object):  
 self.training\_file='Training\_FileFromDB/InputFile.csv'  
 self.file\_object=file\_object  
 self.logger\_object=logger\_object  
  
 def get\_data(self):  
 *"""  
 Method Name: get\_data  
 Description: This method reads the data from source.*

*Input Description:   
 Output: A pandas DataFrame.  
 On Failure: Raise Exception  
  
 Written By: iNeuron Intelligence  
 Version: 1.0  
 Revisions: None  
  
 """* self.logger\_object.log(self.file\_object,'Entered the get\_data method of the Data\_Getter class') # Logging entry to the method  
 try:  
 self.data= pd.read\_csv(self.training\_file) # reading the data file  
 self.logger\_object.log(self.file\_object,'Data Load Successful.Exited the get\_data method of the Data\_Getter class') # Logging exit from the method  
 return self.data # return the read data to the calling method  
 except Exception as e:  
 self.logger\_object.log(self.file\_object,'Exception occured in get\_data method of the Data\_Getter class. Exception message: '+str(e)) # Logging the exception message  
 self.logger\_object.log(self.file\_object,  
 'Data Load Unsuccessful.Exited the get\_data method of the Data\_Getter class') # Logging unsuccessful load of data  
 raise Exception() # raising exception and exiting

Coding Standard:

1. Imports should usually be on separate lines
2. Avoid trailing whitespace anywhere. Because it's usually invisible, it can be confusing.
3. Compound statements (multiple statements on the same line) are generally discouraged
4. Comments should be complete sentences. Always make a priority of keeping the comments up-to-date when the code changes. Ensure that your comments are clear and easily understandable to other speakers of the language you are writing in.
5. Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
6. The name of the variables should start with small case capital letters and a multi word variable should be named as: word1\_word2\_word3.
7. The variable name should be appropriate based on the things that they do. DO NOT USE NAMES LIKE x, k, y etc. Always use a meaningful English word. For example, customer\_name, nearest\_neighbour etc.
8. Method names should start with small case characters. They should start with a verb and make a meaningful sense of what they are supposed to accomplish. For e.g.: load\_data\_from\_sql()
9. Always use self for the first argument to instance methods.
10. Class names should normally use the CapWords convention. Class name should also represent the functionality of the class. For e.g. DataLoader()
11. Modules/Packages/Folders should have short, all-lowercase names. Underscores can be used in the module name if it improves readability. For e.g.: data\_ingestion
12. Constants are usually defined on a module level and written in all capital letters with underscores separating words. Examples include MAX\_OVERFLOW and TOTAL.
13. Comparisons to singletons like None should always be done with is or is not, never the equality operators
14. The code should be properly enclosed withing try and exception blocks and the exceptions should be handled with proper error messages.
15. Additionally, for all try/except clauses, limit the try clause to the absolute minimum amount of code necessary. Again, this avoids masking bugs
16. When a resource is local to a particular section of code, use a with statement to ensure it is cleaned up promptly and reliably after use.
17. Be consistent in return statements. Either all return statements in a function should return an expression, or none of them should. If any return statement returns an expression, any return statements where no value is returned should explicitly state this as return None, and an explicit return statement should be present at the end of the function (if reachable)
18. Object type comparisons should always use isinstance() instead of comparing types directly
19. Don't compare boolean values to True or False using ==