Credit card approval prediction

# Technical Design Document

Version 1.0

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 21th July 2020 | 1.0 | Initial Draft | Sanjay Kumar |
|  |  |  |  |

Contributors

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

|  |  |
| --- | --- |
| Name | Section Worked Upon |
| Sanjay Kumar | Initial Draft |
| Gokul Pisharody | Initial Draft |

Contents

[1 Technical Design Document 2](#_Toc46230219)

[1. Introduction 5](#_Toc46230220)

[High level objectives 5](#_Toc46230221)

Dataset description............................................................................................................................6

[2 Workflow Overall 7](#_Toc46230222)

[Application Flow 7](#_Toc46230223)

[Exception Scenarios Overall 7](#_Toc46230224)

[3 Workflow Data Ingestion and File Conversion 9](#_Toc46230225)

[3.1 Technical solution design 11](#_Toc46230226)

[3.2 Method Definitions 12](#_Toc46230227)

[3.3 Exceptions Scenarios 16](#_Toc46230228)

[4 Data Profiling 17](#_Toc46230229)

[4.1 Method Definition 17](#_Toc46230230)

[5 Stats Based EDA 19](#_Toc46230231)

[5.1 Steps 19](#_Toc46230232)

[5.2 Technical solution design 19](#_Toc46230233)

[5.3 Exceptions Scenarios Module Wise 20](#_Toc46230234)

[6 Graph-Based EDA 21](#_Toc46230235)

[6.1 Technical solution design 21](#_Toc46230236)

[6.2 Method Definitions 21](#_Toc46230237)

[6.3 Exceptions Scenarios Module Wise 22](#_Toc46230238)

[7 Library Based Utils 23](#_Toc46230239)

[7.1 Technical solution design 23](#_Toc46230240)

[7.2 Exceptions Scenarios Module Wise 23](#_Toc46230241)

[8 Data Transformers( Pre-processing steps) 24](#_Toc46230242)

[8.1 Technical solution design 24](#_Toc46230243)

[8.2 Method Definitions 24](#_Toc46230244)

[8.3 Exceptions Scenarios Module Wise 25](#_Toc46230245)

[9 ML Model Selection 26](#_Toc46230246)

[9.1 Technical solution design 26](#_Toc46230247)

[9.2 Exceptions Scenarios Module Wise 26](#_Toc46230248)

[10 Model Tuning and Optimization 27](#_Toc46230249)

[10.1 Technical solution design 27](#_Toc46230250)

[10.2 Method Definitions 28](#_Toc46230251)

[10.3 Exceptions Scenarios Module Wise 28](#_Toc46230252)

[11 Testing Modules 29](#_Toc46230253)

[11.1 Technical solution design 29](#_Toc46230254)

[11.2 Exceptions Scenarios Module Wise 29](#_Toc46230255)

[12 Prediction Pipeline 30](#_Toc46230256)

[12.1 Technical solution design 31](#_Toc46230257)

[12.2 Exceptions Scenarios Module Wise 31](#_Toc46230258)

[13 Deployment Strategy 33](#_Toc46230259)

[13.1 Technical solution design 34](#_Toc46230260)

[13.2 Exceptions Scenarios Module Wise 34](#_Toc46230261)

[14 Monitoring 36](#_Toc46230262)

[14.1 Technical solution design 36](#_Toc46230263)

[14.2 Exceptions Scenarios Module Wise 36](#_Toc46230264)

[15 Logging 38](#_Toc46230265)

[15.1 Technical solution design 38](#_Toc46230266)

[15.2 Common Logging Framework Code 38](#_Toc46230267)

[15.3 Exceptions Scenarios Module Wise 40](#_Toc46230269)

[16 Hardware Requirements 41](#_Toc46230270)

[16.1 Requirements for model training 41](#_Toc46230271)

[16.2 Requirements for model testing 41](#_Toc46230272)

[17 Sample code and standard to be followed: 42](#_Toc46230273)

# Introduction

The goal here is to build an end to end automated Machine Learning solution where a user will be able to predict whether a bank customer should be approved for attaining the credit card or not. The user is only need to give the value of feature variables and the model will able to predict the binary outcome (Approve/ Not Approve). The model will be able take care of all intermediate functionalities like cross validation, hyper pataermeter tuning, algorithm selection etc.

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 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

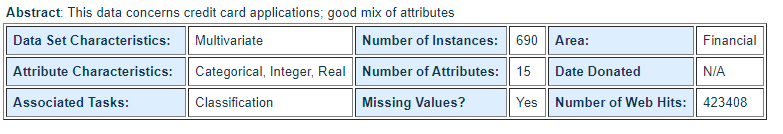
The high-level objectives are:

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. Perform statistical analytics of the data and prepare a table for the analysis and show it on screen.
4. Perform graphical analysis for the data and Showcase the results (graphs) on the screen.
5. Perform data cleaning operation with all the steps required and showcase a report on screen.
6. After data cleaning showcase the graphical analysis once again for comparison.
7. Check whether dimensionality reduction is required or not.
8. Choose the appropriate ML model for training.
9. Perform model tuning.
10. Create a list of top 3 models and show multiple metrics for them.
11. Give option for prediction.
12. Give options for docker container creation.
13. Give option for automatic cloud deployment.

# Dataset description

Data set name : **Credit Approval Data Set**

Details



Data Set Information**:**

This file concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.  
  
This dataset is interesting because there is a good mix of attributes -- continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values.

Attribute Information:

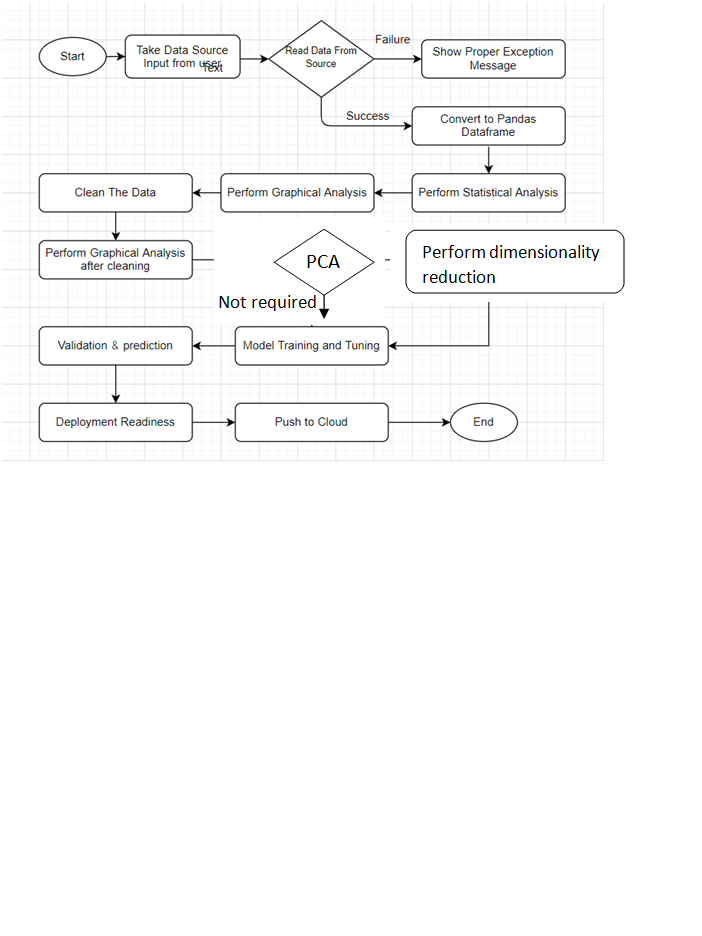
A1: b, a.  
A2: continuous.  
A3: continuous.  
A4: u, y, l, t.  
A5: g, p, gg.  
A6: c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff.  
A7: v, h, bb, j, n, z, dd, ff, o.  
A8: continuous.  
A9: t, f.  
A10: t, f.  
A11: continuous.  
A12: t, f.  
A13: g, p, s.  
A14: continuous.  
A15: continuous.  
A16: +,- (class attribute)

Source

* URL1 - <https://archive.ics.uci.edu/ml/datasets/Credit+Approval>
* URL 2 - <https://www.kaggle.com/echo9k/uci-credit-approval-data-set/activity>

# 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:**

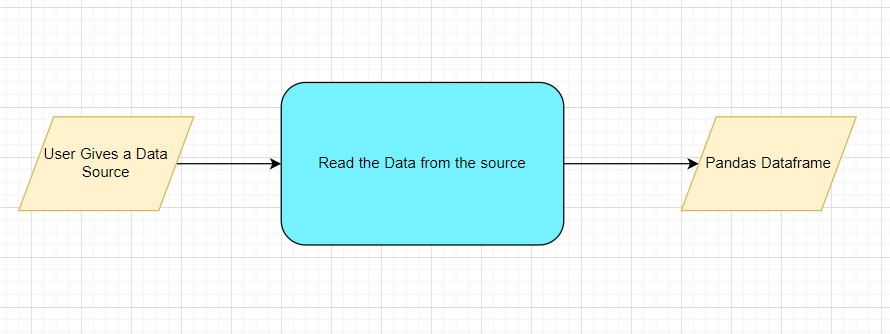
**Phase 1:**

|  |  |
| --- | --- |
| 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 |

# Statistics Based EDA

## Steps

OLS

VIF

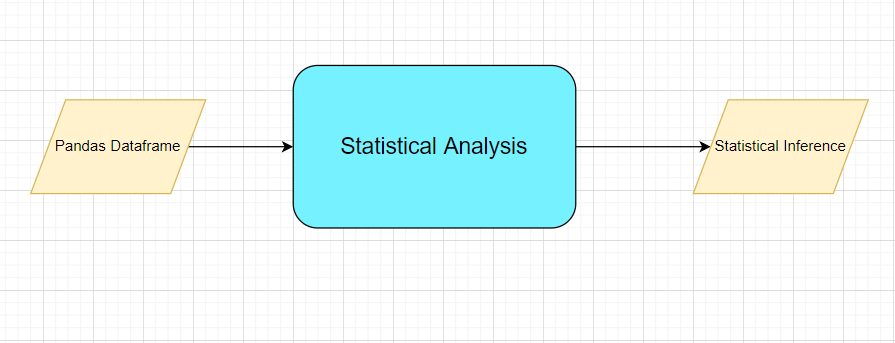
Correlation

Column contributions/ importance

Chi Square test

Z test

## Technical solution design



5.3 Exception 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:

Correlation Heatmaps

Check for balance/imbalance

Count plots

Boxplot for outliers

Piecharts for categories

Line charts for trends

Barplots

Area Charts

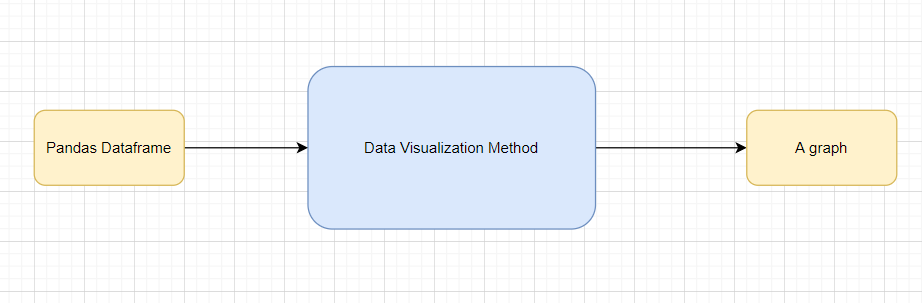
Stacked charts

Scatterplot

## Method definition

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataVisualization** |  |
| 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 |

6.2 Technical solutions design



6.3 Exception scenarios module wise

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

# Library Based Utils

## Technical solution design

## Exceptions Scenarios Module Wise

# Data Transformers( Pre-processing steps)

Null value handling

Categorical to numerical

Imbalanced data set handling

Handling columns with std deviation zero or below a threshold

Normalisation

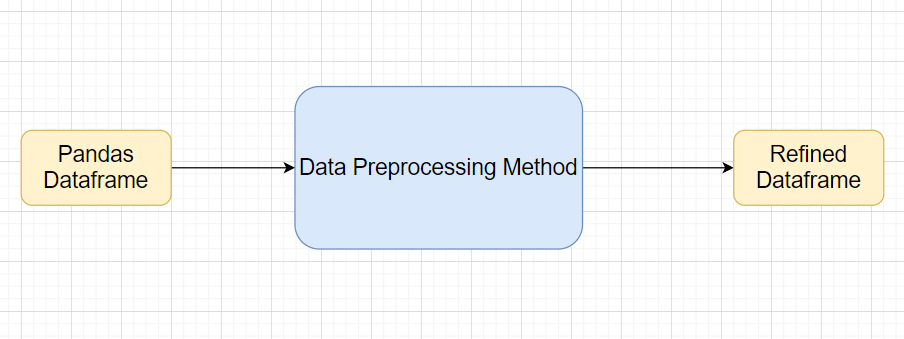
PCA

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 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 parameters passed to the methods | Handle Internally | Code should never give a wrong input |

# ML Model Selection

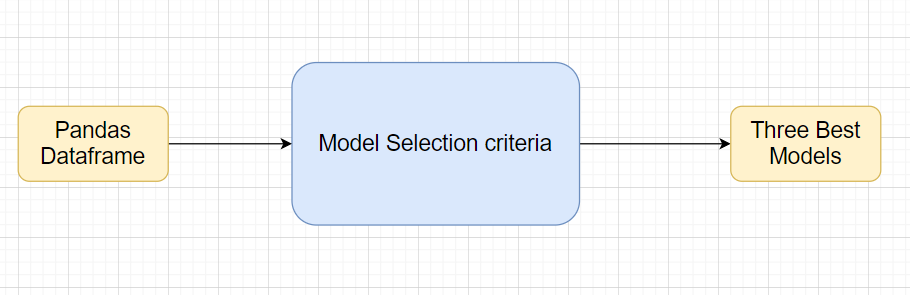
**MVP:**

3 Models—Logistic regression, Naive bayes classification, K nearest neighbour classification

**Phase1:**

Model Selection criteria

## Technical solution design



# Exception scenario for model selection

|  |  |  |
| --- | --- | --- |
| **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.

Logistic Regression

Decision Tree

Random Forest

XG Boost

Support Vector Classifier

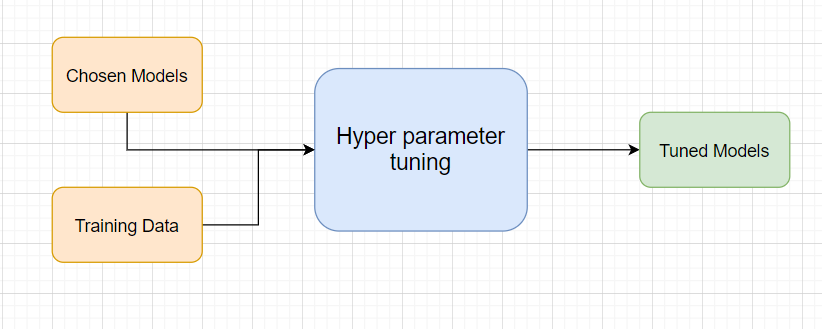
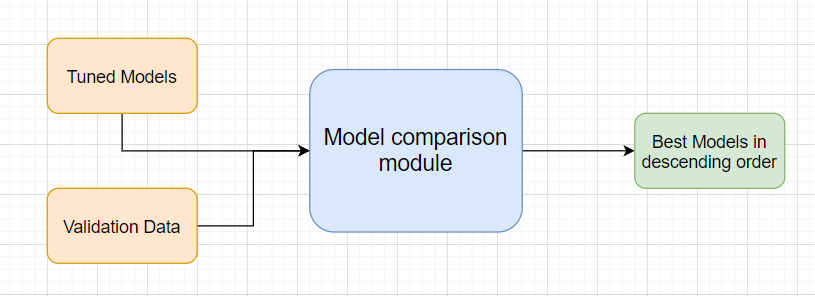
KNN Classifier

Naïve Baye’s

Model selection criteria:

Accuracy, AUC, Precision, Recall, F1 Score

## 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 |  |
|  | 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

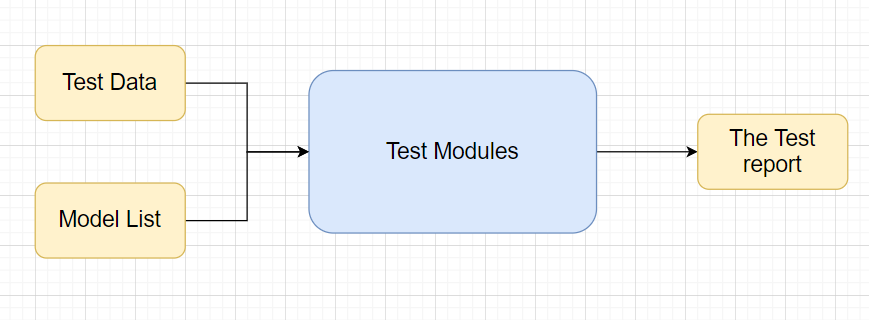
Use test data to have tests run on the three best models

Give the test report

1. Accuracy
2. Precision
3. Recall
4. F1 Score

**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

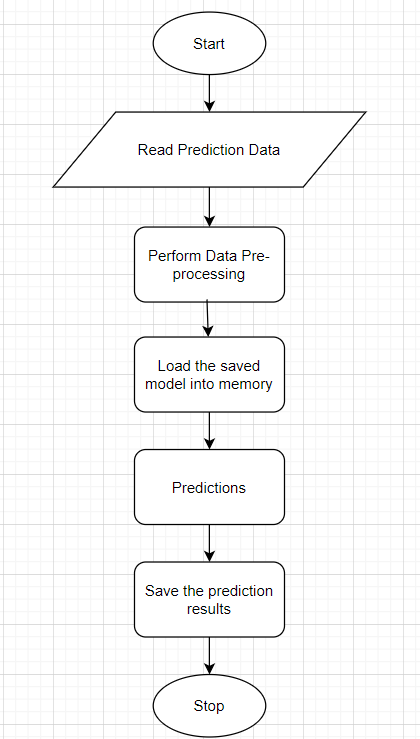
Do predictions

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:

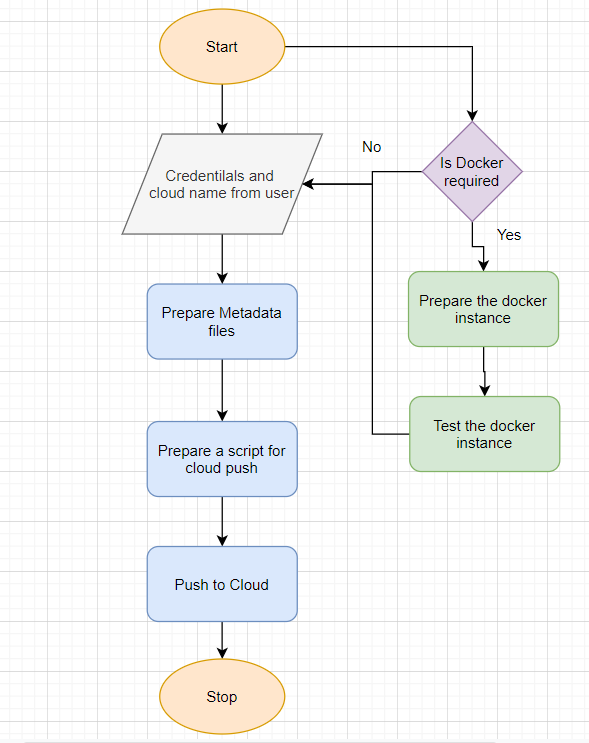
Accept the user credentials

Prepare a script file to push changes

Docker instance

Push of the docker instance to cloud

## Technical solution design



## 

## 14.2 Exception 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.)

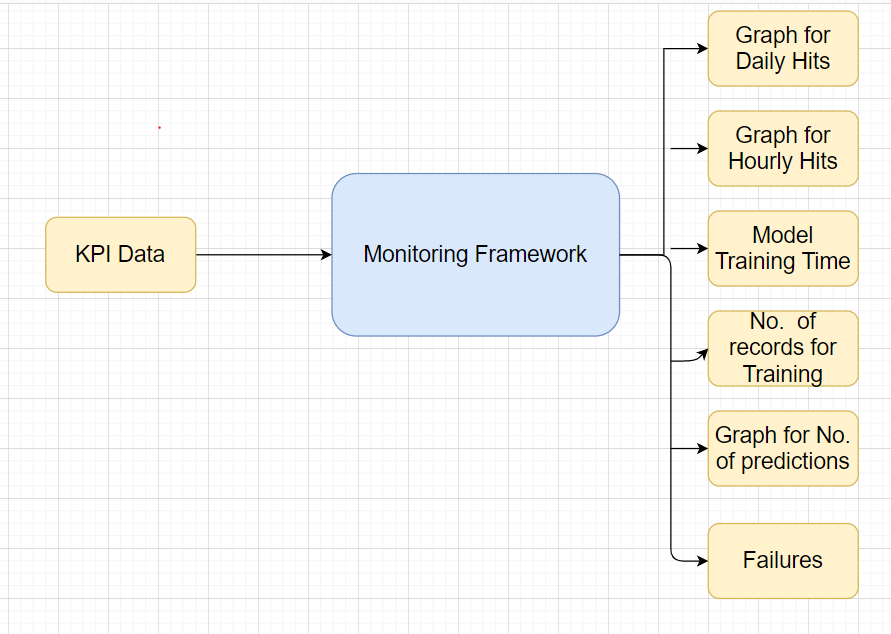
No. of hits

Training data size (number of rows)

Time spent in training

Failures

## Technical solution design



## 

## 15.1 Exception 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



|  |  |
| --- | --- |
| 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 |

# 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.  
 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 ==