Software Design Document

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2019-04-29 | 0.1 | Initial Draft Version | Panduranga Nayak |
|  |  |  |  |
|  |  |  |  |

1. **Introduction**

Purpose:

This document provides the design overview of the ETL challenge for Auto 1 Group. It is intended to capture and convey the significant architectural decisions which have been made on the system.

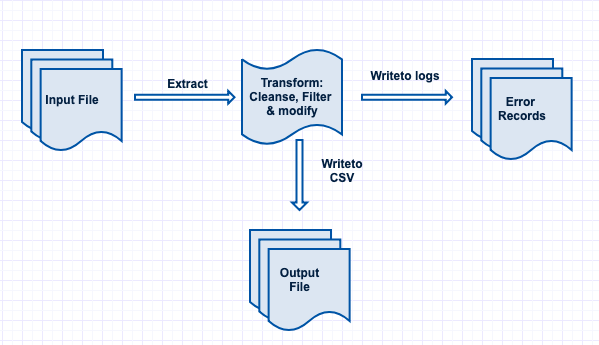
Scope:

The document provides the architectural overview of the various processes involved in extraction, transformation and load of an input file which consists of various attributes reflated to an automobile. The transformed data will be input for price prediction machine learning algorithm. The implementation of ML is not part of the scope.

1. **Architecture**

* Standard python environment **version 3.5** is used and no additional libraries are installed (pip install).
* The input file is placed in the sub-directory named data inside the ETL\_Auto1\_Group parent directory.
* The python process consumes the above input file and transforms the data and generates **the output file output.csv** in the ETL\_Auto1\_Group directory.
* The python method also **returns the output in the form of matrix** (list of lists).
* Python standard **csv module** is used to read the delimited input and also write the output file output.csv is **utf-8** encoding
* Bad or erroneous records are tracked for validation and are logged in a file separately under the logs folder.

**Architecture Diagram**

****

1. **Methods**

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Arguments** | **Description** |
| **load** | *data\_file\_name* – filename of the input delimited file.  *delimiter* – Delimiter of the input file. | This method loads the input data file and returns the row objects as a list of dictionaries for further processing. |
| **bad\_records\_filter** | *Row* – one row object of type dict. | This method filters the bad and erroneous records from getting processed. |
| **one\_hot\_encoding** | *Rows-* dictionary containing the record set  *categorical\_column* - Column containg the categorical data | This method returns the unique categorical data from the given column |
| **transform** | *file\_name* – filename of the input delimited file. | This method transforms the input file in the required format and returns the output as a matrix with the first value being the header. |
| **write\_to\_csv** | *record\_set* – matrix output from transform method  *delimiter –*Delimiter for the output file  *output\_filename –*File name of the output. | This method writes the output to a file |

1. **Mappings**

|  |  |  |  |
| --- | --- | --- | --- |
| **Target Column** | **Source Column** | **Data Type** | **Transformations** |
| engine-location | engine-location | int | One Hot encoding. This column will be further split into multiple columns based on the number of unique categorical data |
| num-of-cylinders | num-of-cylinders | int | Input is in the form of words. E.g. One, Two.. etc. Covert it into digits. 1, 2.. etc. |
| engine-size | engine-size | int | Pass through column |
| weight | weight | int | Pass through column |
| horsepower | horsepower | float | Convert to float from German notation |
| aspiration | aspiration | int | If aspiration is Turbo set this to 1. For other values set this to 0 |
| price | price | float | Convert the input which is in cents to euro. |
| make | make | String | Pass through column |

1. **Possible Improvements**

* Better management of input and output files. File movement from input directory to processing directory and processed directory can be maintained for tracking daily file load.
* Logging of important steps and counts of the source, target and bad records for validation. Maintaining audit logs for each load can be maintained for this
* Better exception handling and logging.
* Using Python pandas to implement a code which is both robust and provides better code readability.

1. **Execution**

Code to execute:

import ETL\_Auto1\_Group

x=ETL\_Auto1\_Group.perform\_etl.transform('Challenge\_me.txt')

print(x)

Output:

