EssayAutoGrade

# 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 |
| 27th November 2020 | 1.0 | Initial Draft | Akram Ahmad |

Contributors

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

|  |  |
| --- | --- |
| Name | Section Worked Upon |
| Akram Ahmad | 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 Natural Language Processing solution where the user will only give the data as input and the result will be the best performing hyper tuned Deep Learning and Natural Language Processing 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 EssayAutoGrade 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 3.6**

## High level objectives

1. To Import Required Libraries
2. Read the Data
3. Inspect the Data
4. Split Text into Sentences
5. Word Embeddings
6. Text Preprocessing
7. Vector Representation of Sentences
8. Similarity Matrix Preparation
9. Sentence ranking
10. Summary Extraction
11. Split of test and train dataset
12. Model building
    1. Instantiate keras tensor
13. Building of embedding layes
14. Building LSTM network
15. Merging of LSTM with features
16. Building dense layesr
17. Building final model
18. Model compilation and fitting
19. Model evalution
20. Perform model Tuning.
21. Create a list of top 3 models and show multiple metrics for them.
22. Give option for prediction.
23. Give options for docker container creation.
24. Give option for automatic cloud deployment.

Automatic Text Summarization is a hot topic of research, and in this article, we have covered just the tip of the iceberg. Going forward, we will explore the abstractive text summarization technique where deep learning plays a big role. In addition, we can also look into the following summarization tasks:

Problem-specific

* Multiple domain text summarization
* **Single document summarization**
* Cross-language text summarization (source in some language and summary in another language)

Algorithm-specific

1. **Text summarization using RNNs and LSTM**
2. Text summarization using Reinforcement Learning
3. Text summarization using Generative Adversarial Networks (GANs)

The high-level objectives are:

1. Enable Text Summarization using NLP ?
2. Sequence-to-Sequence (Seq2Seq) Modeling
3. Encoder – Decoder Architecture
4. Limitations of the Encoder – Decoder Architecture
5. The Intuition behind the Attention Mechanism
6. Understanding the Problem Statement
7. Implementing a Text Summarization Model in Python using NLTK
8. What’s Next ?
9. How does the Attention Mechanism Work ?
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

**Life cycle of an NLP Project**

Input  
Text Document

Text Pre-processing (Parsing)  
Tokenization

Stemming

Lemmatization

Removing stopwords

Text Pre-processing (Parsing)  
POS tagging

Parsing

NER(Named Entity Recognization)

Chunking

Feature Engineering  
TF-IDF

Bag of words

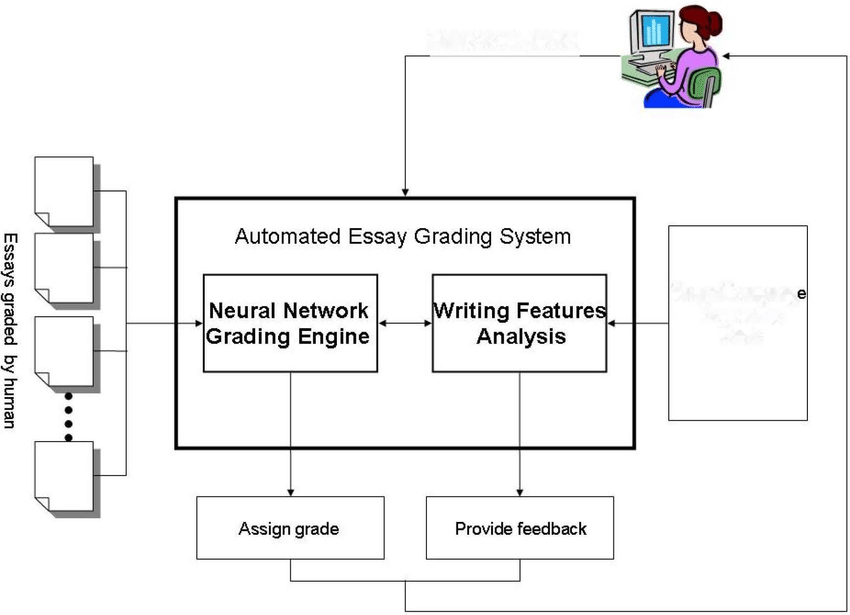
WordtoVec

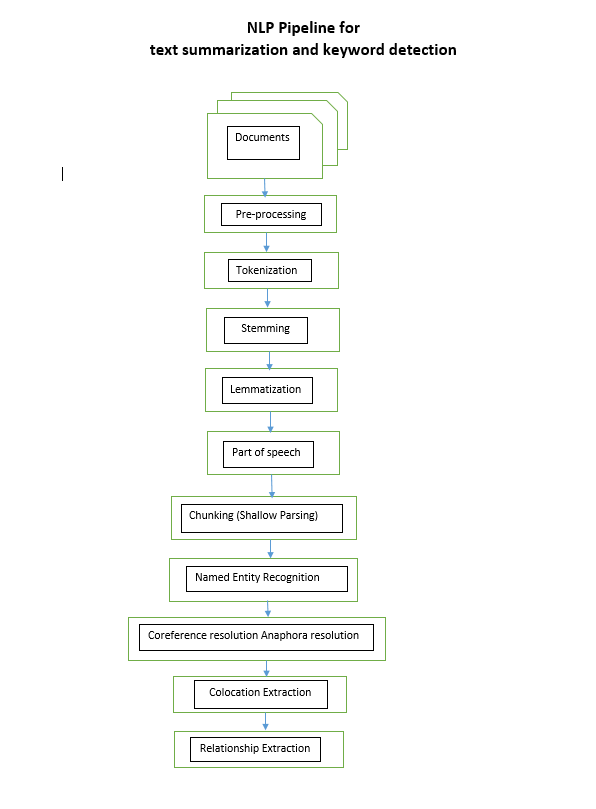
Word embedding

Modelling

Deployment

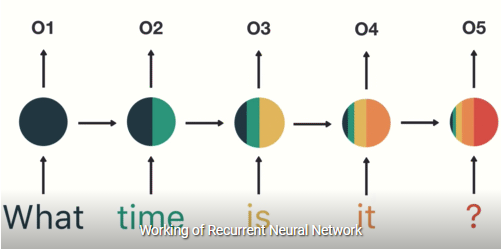
## Concept Flow



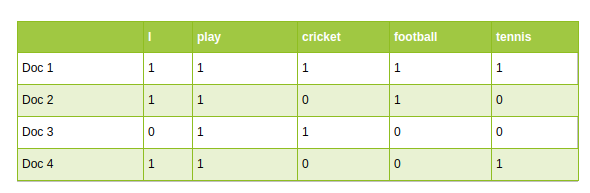


* **Tokenization**

Tokens are the building blocks of Natural Language. Creating Vocabulary is the ultimate goal of Tokenization.



In Advanced Deep Learning-based NLP architectures, vocabulary is used to create the tokenized input sentences. Finally, the tokens of these sentences are passed as inputs to the model



* 1. **Word Tokenization**

Word Tokenization is the most commonly used tokenization algorithm. It splits a piece of text into individual words based on a certain delimiter. Depending upon delimiters, different word-level tokens are formed. [**Pretrained Word Embeddings**](https://www.analyticsvidhya.com/blog/2020/03/pretrained-word-embeddings-nlp/?utm_source=blog&utm_medium=what-is-tokenization-nlp) such as Word2Vec and GloVe comes under word tokenization.

* 1. **Character Tokenization**

Character Tokenization splits apiece of text into a set of characters to deal with out of vocabulary words during word tokenization.

Character tokens solve the OOV problem but the length of the input and output sentences increases rapidly as we are representing a sentence as a sequence of characters. As a result, it becomes challenging to learn the relationship between the characters to form meaningful words.

* 1. **Subword Tokenization**

Subword Tokenization splits the piece of text into subwords (or n-gram characters). For example, words like lower can be segmented as low-er, smartest as smart-est, and so on.

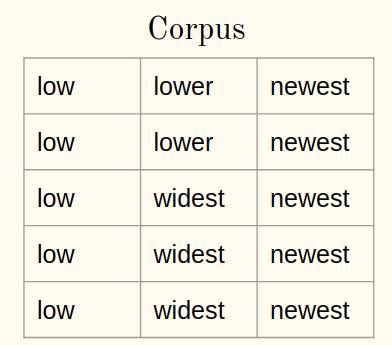
Transformed based models – the SOTA in NLP – rely on Subword Tokenization algorithms for preparing vocabulary. Now, I will discuss one of the most popular Subword Tokenization algorithm known as Byte Pair Encoding (BPE).

**Steps to learn BPE**

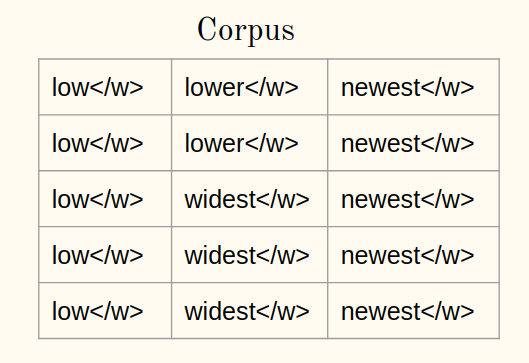
1. Split the words in the corpus into characters after appending </w>
2. Initialize the vocabulary with unique characters in the corpus
3. Compute the frequency of a pair of characters or character sequences in corpus
4. Merge the most frequent pair in corpus
5. Save the best pair to the vocabulary
6. Repeat steps 3 to 5 for a certain number of iterations

We will understand the steps with an example.

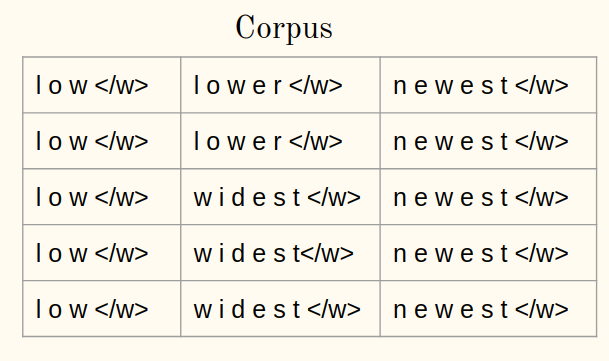
Consider a corpus:



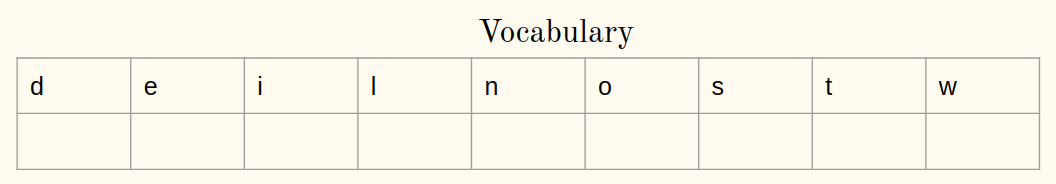
1a) Append the end of the word (say </w>) symbol to every word in the corpus:



1b) Tokenize words in a corpus into characters:

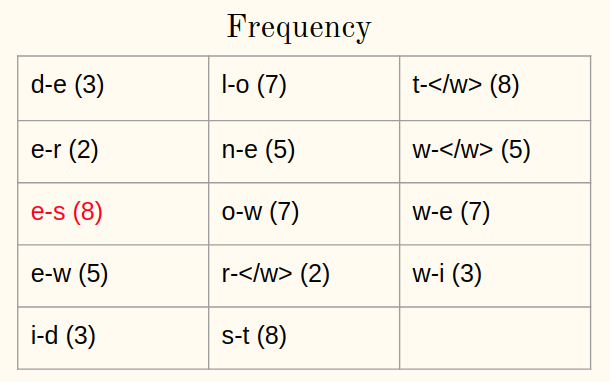


1. **Initialize the vocabulary:**

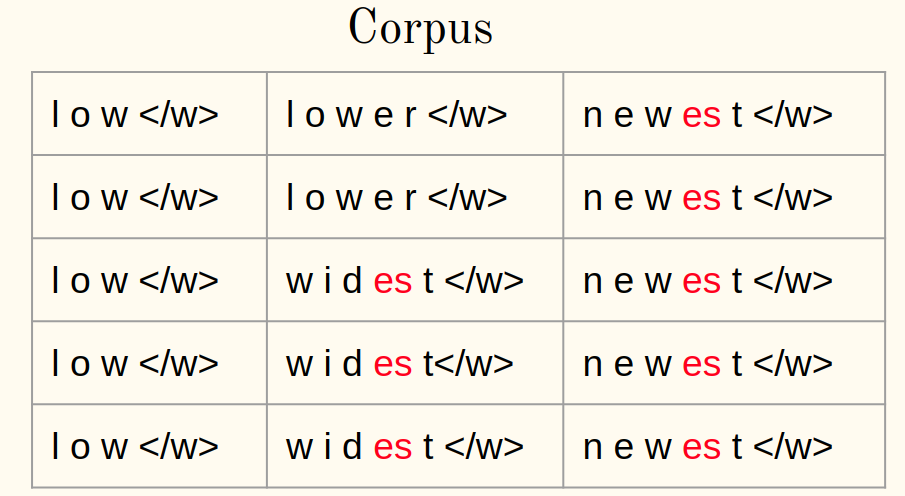


**Iteration 1:**

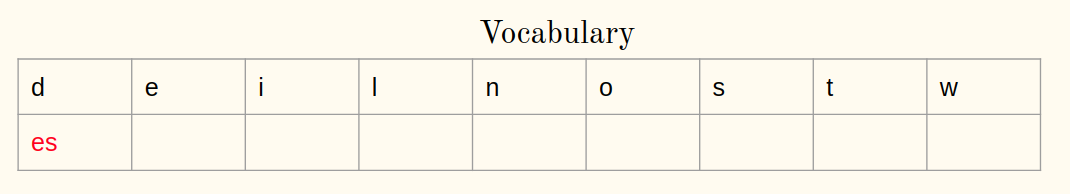
1. **Compute frequency:**



1. **Merge the most frequent pair:**



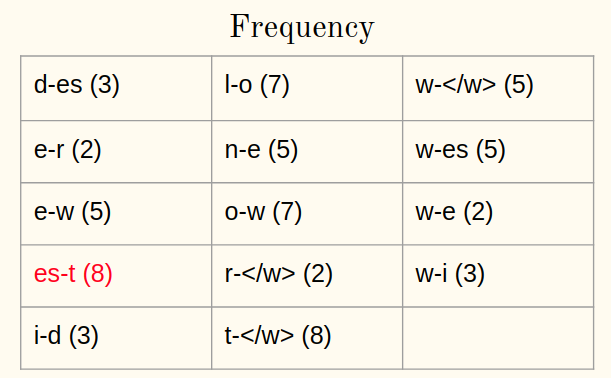
1. **Save the best pair:**



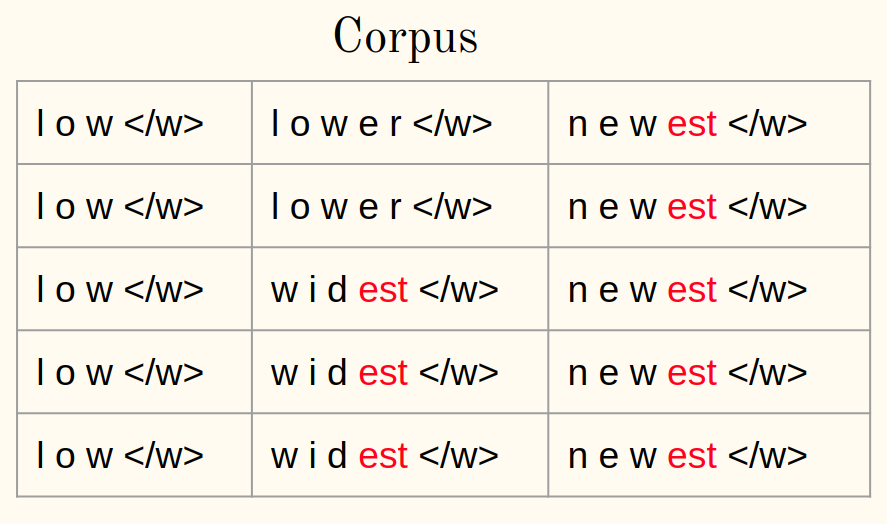
Repeat steps 3-5 for every iteration from now. Let me illustrate for one more iteration.

**Iteration 2**

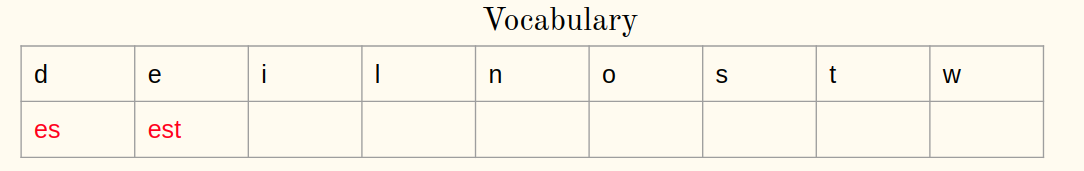
* 1. **Compute frequency:**



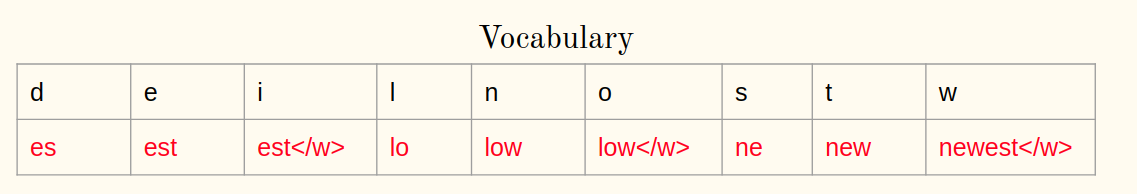
* 1. **Merge the most frequent pair:**



* 1. **Save the best pair:**



After 10 iterations, BPE merge operations looks like:



**Note :** The python code for BPE is already available in the original paper itself (Neural Machine Translation of Rare Words with Subword Units, 2016) for [CODE](https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/#:~:text=Tokenization%20is%20a%20common%20task%20in%20Natural%20Language%20Processing%20(NLP).&text=Tokenization%20is%20a%20way%20of,words%2C%20characters%2C%20or%20subwords.)

* **Stemming**

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. **Stemming** is important in **natural language understanding** (NLU) and **natural language processing (NLP)**. Stemming is also a part of queries and Internet search engines.

Just observe the following

wait (infinitive)

wait (imperative)

waits (present, 3rd person, singular)

wait (present, other persons and/or plural)

waited (simple past)

waited (past participle)

waiting (progressive)

Sometime spelling may also change in order to make a new word.

beauty, duty + -ful → beautiful, dutiful (-y changes to i)

heavy, ready + -ness → heaviness, readiness (-y changes to i)

able, possible + -ity → ability, possibility (-le changes to il)

permit, omit + -ion → permission, omission (-t changes to ss)

There are majorly 2 errors in Stemming Algorithms which are as follows.

1. **Over Stemming**

Over-stemming is when two words with different stems are stemmed to the same root. This is also known as a false positive.

universal

university

universe

All the above 3 words are stemmed to univers which is wrong behavior.

1. **UnderStemming**

Under-stemming is when two words that should be stemmed to the same root are not. This is also known as a false negative. Below is the example for the same.

alumnus

alumni

alumnae

therew are 3 popular ways to performing stemming which falls under **truncating Stemming Algorithm.**

1. Porter stemmer (English stemmer or porter2 stemmer)
2. Snowball stemmer
3. Lancaster stemmer

Note : for example code [CODE](https://medium.com/@tusharsri/nlp-a-quick-guide-to-stemming-60f1ca5db49e)

Other categories of stemmer algorithms are :

* **Statistical stemmer**
* **Mixed stemmer**
* **Lammetization**

In simpler forms, a method that switches any kind of a word to its base root mode is called*Lemmatization***.**

**‘troubled’ -> Lemmatization -> ‘trouble’**

**‘troubled’ -> Stemming -> ‘troubl’**

* **Part of Speech**

Now, if we talk about Part-of-Speech (**PoS**) tagging, then it may be defined as the process of assigning one of the parts of speech to the given word. It is generally called POS tagging. In simple words, we can say that POS tagging is a task of labelling each word in a sentence with its appropriate part of speech.

We already know that parts of speech include nouns, verb, adverbs, adjectives, pronouns, conjunction and their sub-categories.

**Rule-based POS Tagging**

Rule-based taggers use dictionary or lexicon for getting possible tags for tagging each word. If the word has more than one possible tag, then rule-based taggers use hand-written rules to identify the correct tag. Disambiguation can also be performed in rule-based tagging by analyzing the linguistic features of a word along with its preceding as well as following words. For example, suppose if the preceding word of a word is article then word must be a noun.

**Stochastic POS Tagging**

The model that includes frequency or probability (statistics) can be called stochastic. Any number of different approaches to the problem of part-of-speech tagging can be referred to as stochastic tagger.

### **Word Frequency Approach**

### **Tag Sequence Probabilities**

**Transformation-based Tagging**

Transformation based tagging is also called Brill tagging. It is an instance of the transformation-based learning (TBL), which is a rule-based algorithm for automatic tagging of POS to the given text. TBL, allows us to have linguistic knowledge in a readable form, transforms one state to another state by using transformation rules.

**Hidden Markov Model (HMM) POS Tagging**

he POS tagging process is the process of finding the sequence of tags which is most likely to have generated a given word sequence. We can model this POS process by using a Hidden Markov Model (HMM), where tags are the hidden states that produced the observable output, i.e., the words.

* **Chunking or Shallow parsing**

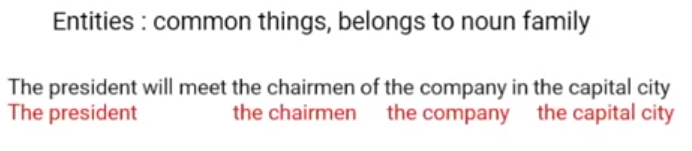
**Chunk extraction or Partial parsing or Shallow Parsing** is a process of meaningful extracting short phrases from the sentence (tagged with Part-of-Speech). Chunks are made up of words and the kinds of words are defined using the part-of-speech tags. One can even define a pattern or words that can't be a part of chuck and such words are known as **chinks.** It works on top of POS tagging. It uses POS-tags as input and provides chunks as output.

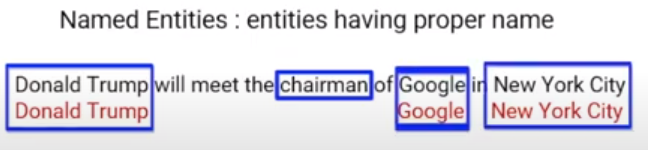
Chunking can break sentences into phrases that are more useful than individual words and yield meaningful results.

* **Named Entity Recognition**

In any text document, there are particular terms that represent specific entities that are more informative and have a unique context. These entities are known as named entities , which more specifically refer to terms that represent real-world objects like people, places, organizations, and so on, which are often denoted by proper names. A naive approach could be to find these by looking at the noun phrases in text documents. Named entity recognition (NER) , also known as entity chunking/extraction , is a popular technique used in information extraction to identify and segment the named entities and classify or categorize them under various predefined classes.

SpaCy has some excellent capabilities for named entity recognition.







* **Coreference Resolution**

Coreference is to identify all noun phrases (mentions) that refers to the same entity.

Examples

Barak Obama, The US former presedent tweeter his new year wishes.

**(Coreference)**

The music was so loud that it could not be enjoyed.

**(Anaphora)**

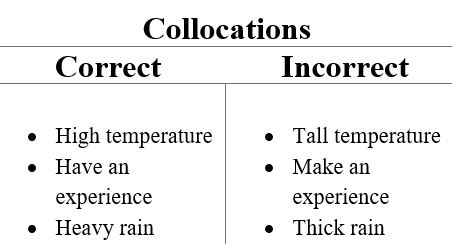
Despite her difficulty, Swati went ahead to help him.

**(Cataphora)**

* **Collocation Extraction**

**Collocations** are two or more words that tend to appear frequently together, for example – *United States*. There are many other words that can come after United, such as the United Kingdom and United Airlines. As with many aspects of natural language processing, context is very important. And for collocations, context is everything.

In the case of collocations, the context will be a document in the form of a list of words. Discovering collocations in this list of words means to find common phrases that occur frequently throughout the text.



* **Relationship Extraction**

Relationship extraction is the task of extracting semantic relationships from a text. Extracted relationships usually occur between two or more entities of a certain type (e.g. Person, Organisation, Location) and fall into a number of semantic categories (e.g. married to, employed by, lives in).

## 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:**

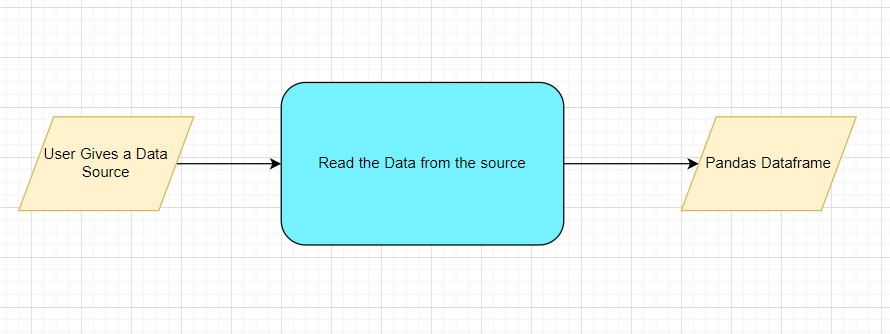
**Phase 1:**

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

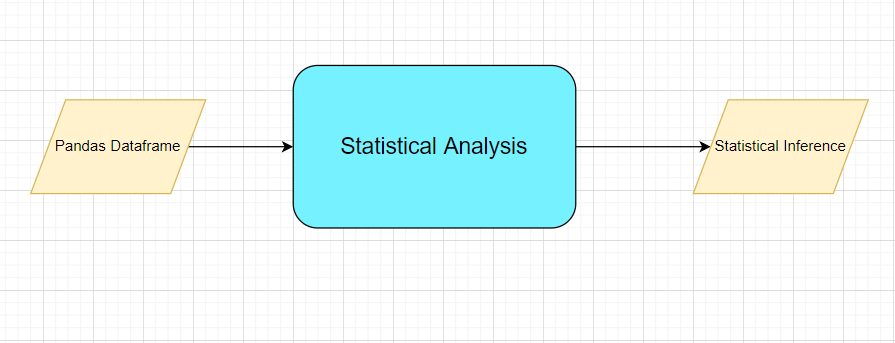
|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataGetter** |  |
| Method Name | read\_data\_from\_csv |  |
|  | Method Description | This method will be used to read data from a csv file or a flat file |
|  | Input parameter names | self,file\_name, header,names, use\_cols, separator |
|  | Input 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 |  |

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

## Technical solution design



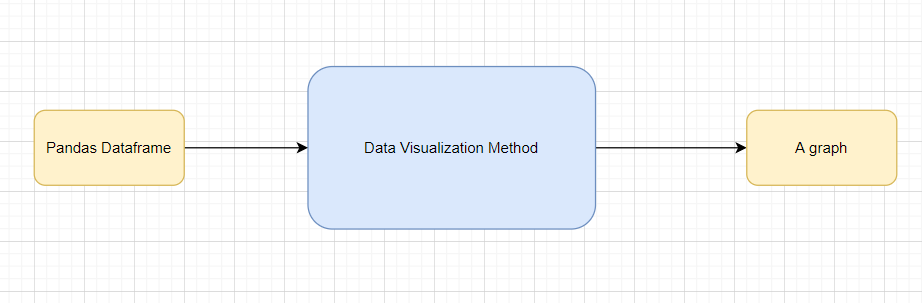
## Method Definitions

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

## 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

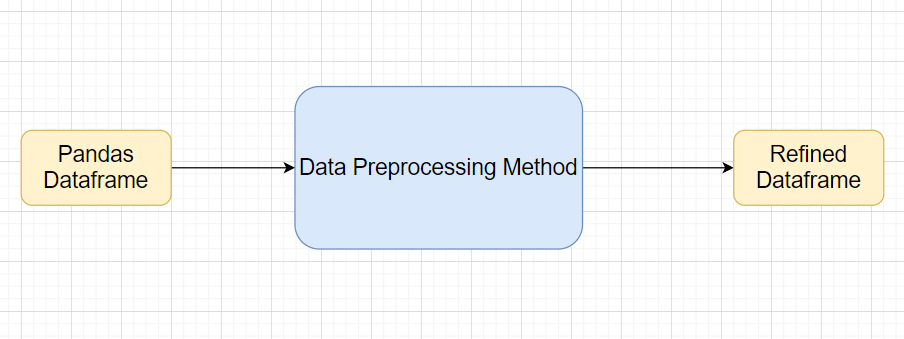
**Phase1:**

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 |

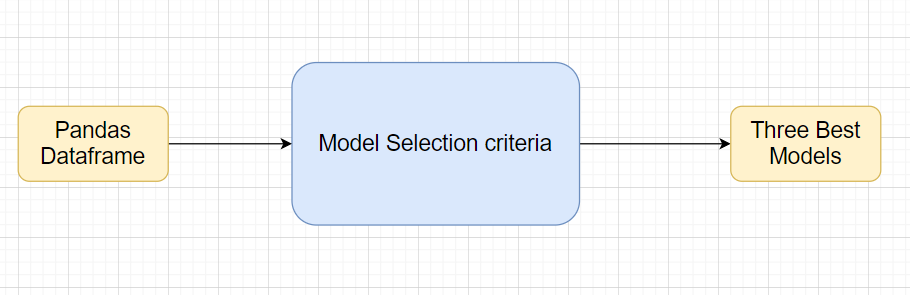
## Exceptions Scenarios Module Wise

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

# Model Selection

Model Selection criteria

## Technical solution design

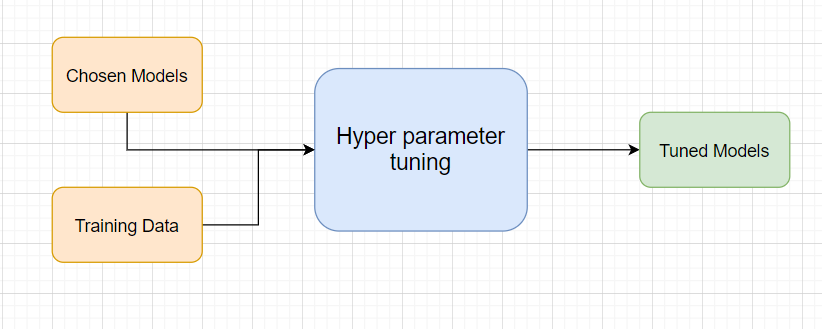
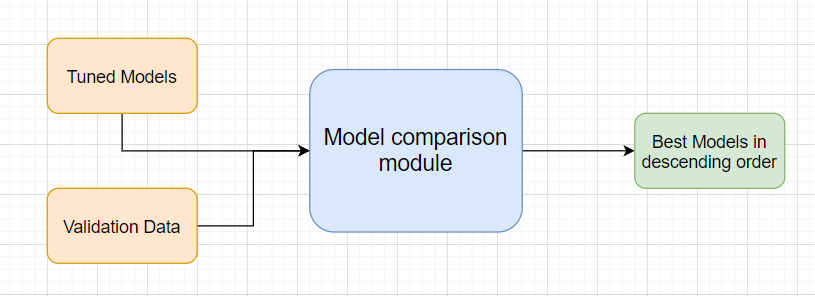


## Exceptions Scenarios Module Wise

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

# Model Tuning and Optimization

## Technical solution design

1. 
2. 

## Method Definitions 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

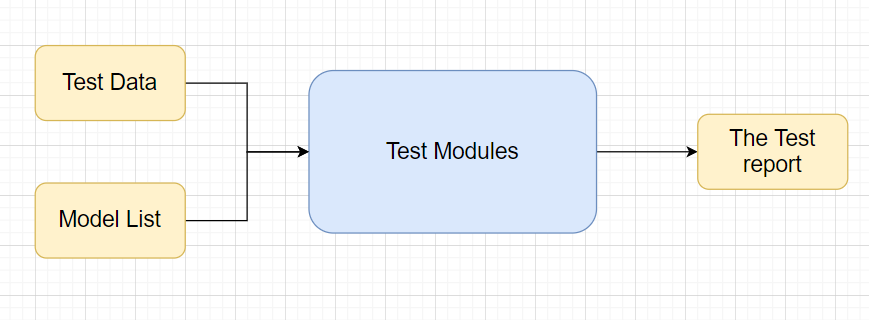
**Phase 2**

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

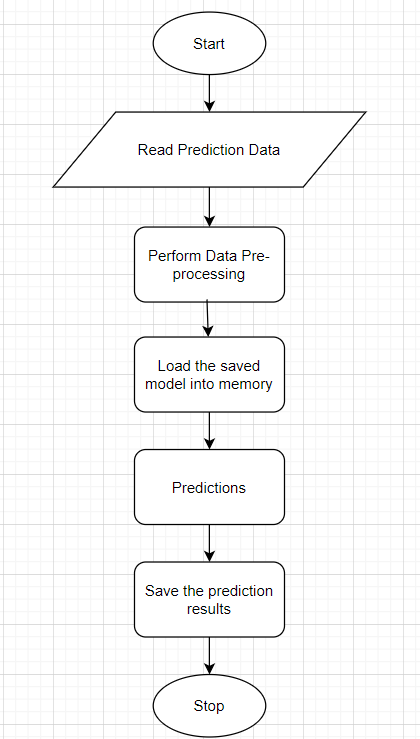
Do predictions

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:

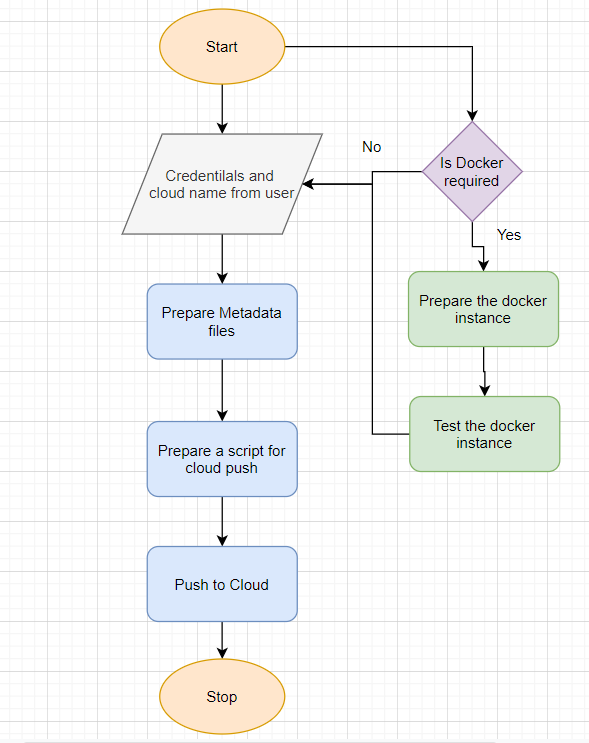
Accept the user credentials

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.)

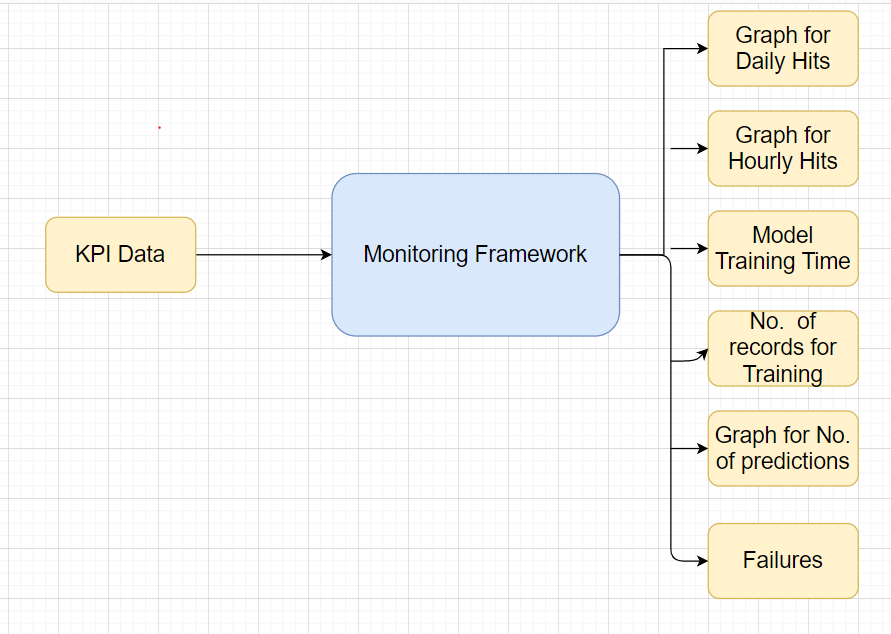
No. of hits

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 |

# 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:

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 ==

References

1. <https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/#:~:text=Tokenization%20is%20a%20common%20task%20in%20Natural%20Language%20Processing%20(NLP).&text=Tokenization%20is%20a%20way%20of,words%2C%20characters%2C%20or%20subwords>.
2. <https://medium.com/@tusharsri/nlp-a-quick-guide-to-stemming-60f1ca5db49e>
3. <https://www.analyticssteps.com/blogs/what-stemming-and-lemmatization-nlp>
4. <https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_part_of_speech_tagging.htm#:~:text=It%20is%20generally%20called%20POS,conjunction%20and%20their%20sub%2Dcategories>.

1. <https://www.aclweb.org/anthology/C18-1094.pdf>
2. <https://www.researchgate.net/figure/Automated-Essay-Grading-System-AEGS-Framework_fig1_313805557>
3. <https://www.youtube.com/watch?v=MmgjhvOSd-E>
4. <https://www.geeksforgeeks.org/nlp-word-collocations/>
5. <http://nlpprogress.com/english/relationship_extraction.html#:~:text=Relationship%20extraction%20is%20the%20task,employed%20by%2C%20lives%20in>).