Report

ML Preprocessor CLI

Duration of the project: Nov 2021 - Present

Project Mentors: Aditya Miskin, Ali Murtaza

Team Members: Tarush Shankar, Akil Vanam, Anush Udupi

INTRODUCTION

Project Statement:

To make a CLI tool to preprocess a dataset.

Inspiration:

Preprocessing is a crucial part in Machine Learning. Since the quality of the dataset directly affects the ability of the model to learn. And preprocessing requires a lot of time for ML developers which can be reduced by a significant amount by using this CLI tool.
Already existing applications do not provide such ease of handling data. Moreover, we plan to add several new features that are non-existent in current applications.

Background Info:

The aim of this CLI tool is to start off by implementing 3 essential preprocessing features: Null-Handling, Character-Encoding and Feature-Scaling, and further it by incorporating PCA(Principal Component Analysis) and certain widely used data analysis techniques.

METHODOLOGY

Firstly, the user has to input the desired dataset which should be in a .csv file format. We check if the requested dataset is available and accordingly print error statements if any.Using the PyInquirer library for making a menu driven CLI tool like checkboxes or lists, they will be prompted with multiple choices for each stage. The tool would look similar to the images provided below:
(IMAGE OF MAIN.PY FIRST CHOICES)

Handling Null Values:

As many machine learning algorithms do not support missing values, handling null values is a crucial step of preprocessing. Here, the CLI tool will prompt the user to pick different methods to handle null values.
Both the below mentioned methods have been implemented with Pandas library methods.

Remove Columns:

One method to handle null values is by removing the entire redundant column. By removing the entire column, we create a robust model. The cons include loss of information. Also, if the percentage of missing values is excessive, the model might work properly due to lack of significant data.

Fill Null Values (with mean/median/mode):

Columns in the dataset which are having numeric continuous values can be replaced with the mean, median, or mode of remaining values in the column. This method can prevent the loss of data compared to the earlier method. It is a statistical approach to handle the missing values. Though it prevents data loss, the major con here is that it works best for small dataset and when the values are numerical continuous variables.

Encoding Categorical Data:

Character encoding is the process of assigning numbers to graphical characters. Handling such variables is another integral aspect of Machine Learning. Categorical variables are basically the variables that are discrete and not continuous.

Categorical variables are further classified into 2 types - Ordinal CVs(categorical variables) and Nominal CVs.

Ordinal CVs:

These are variables that can be ordered. We could provide a mathematical relation between each other. Example: Size of Shirt. We can say Small<Medium<Large. We can handle the ordinal CVs by using a simple and neat technique known as Label Encoder.

Nominal CVs:

These are variables that cannot be ordered. Example: Colour of Shirt. We cannot say Blue > Green. Using the ordinal methods for nominal CVs would be a blunder. The correct way to handle Nominal CVs is to use One-Hot Encoding. Using the get_dummies() function from pandas library, we essentially create 'n' columns. Each category is mapped with a binary variable containing either 0 or 1. Here, 0 represents the absence, and 1 represents the presence of that category.

Feature Scaling:

Feature Scaling is the process of scaling the values of features to a more manageable scale. You'll typically perform it before feeding these features into algorithms that are affected by scale.
Used scikit-learn for implementing Normalisation/MinMaxScaling, Standardisation/z-score, PowerTransformer, RobustScaler,MaxAbsScaler, QuantileTransformer.

Data Analysis:

Dataframe description:

Users are given options to see data frame properties and analytics of whole data frame and individual columns and printing number of rows entered by user

Dataframe correlation with heatmap:

A correlation heatmap is a heat map that shows a 2D correlation matrix between two discrete dimensions, using colored cells to represent data from usually a monochromatic scale. The values of the first dimension appear as the rows of the table while of the second dimension as a column.

Univariate Analysis:

Univariate analysis is the simplest form of analysing data. “Uni” means “one”, so in other words the data has only one variable. It doesn’t deal with causes or relationships and it’s major purpose is to describe. It takes data, summarises that data and finds patterns in the data.

Bivariate Analysis:

Unlike Univariate Analysis, Bivariate Analysis consists of two variables. It will be helpful for the purpose of determining the empirical relationship between the two variables. Bivariate analysis can be helpful in testing simple hypotheses of association We have Scatter plot,Hex plot,Violin plot, BoxPlot,Correlation heat map, Simple linear regression model stats as under bivariate analyser

PCA:

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity.

Implementation is as follows:

1) The dataset size is reduced to remove redundant columns and only keep those columns with meaningful data, the 'feature' columns.

2) This dataset is then null-handled.

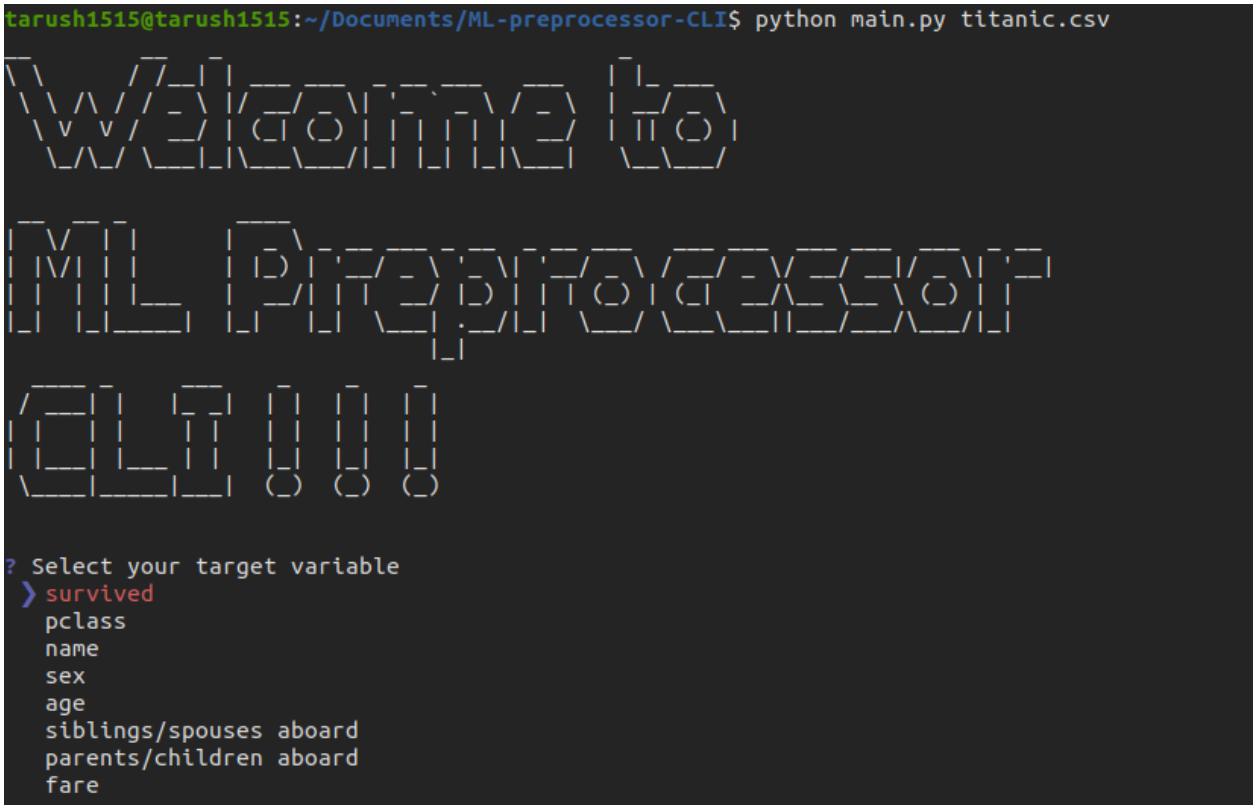
3) Non-integer/float type columns are Label Encoded using sklearn and the whole dataset is then scaled with Standard Scaler.

4) The dataset is then fit into sklearn's PCA module with n=2 components as 2 is generally the standard for PCA analysis and for ease of visualisation purposes. The eigenvalues of each component are extracted and pushed into a new dataframe along with the target column.

RESULTS:

The Titanic dataset from kaggle was used majorly for testing the features and the screenshots for each of the features are as below:

Main Menu:



We select Survived as that is the target column in the Titanic dataset.

Data Analysis:

```
? Select your target variable
Done.......😁

Tasks (Preprocessing)👇

? What do you want to do?

Tasks (Data Description)👇

? How would you like to describe the dataset?
? How would you like to describe the dataset?
 ❯ Describe a specific Column
   Show Properties of Each Column
   Dataframe correlation with heatmap
   Show the Dataset
   Univariate Analysis
   Bivariate Analysis
   Go Back
```

Describe:

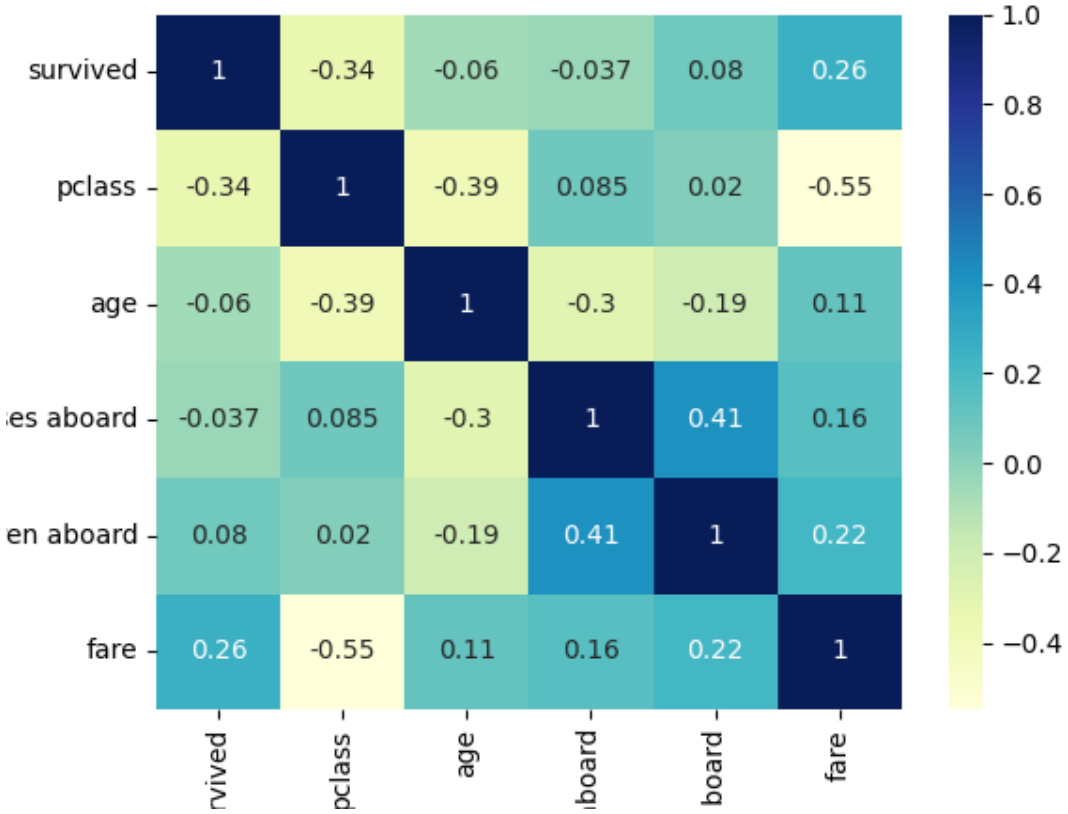
```
? How would you like to describe the dataset?
? Which Column?  sex
count      887
unique       2
top       male
freq       573
Name: sex, dtype: object
```

Show properties

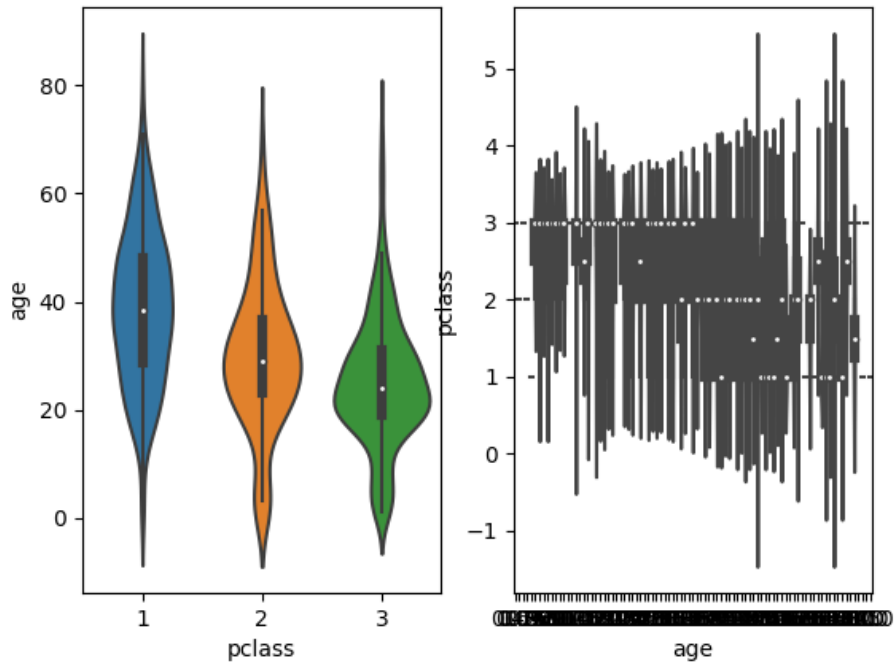
```
? How would you like to describe the dataset?
         survived      pclass  ...  parents/children aboard       fare
count  887.000000  887.000000  ...                887.000000  887.00000
mean     0.385569    2.305524  ...                  0.383315   32.30542
std      0.487004    0.836662  ...                  0.807466   49.78204
min      0.000000    1.000000  ...                  0.000000    0.00000
25%      0.000000    2.000000  ...                  0.000000    7.92500
50%      0.000000    3.000000  ...                  0.000000   14.45420
75%      1.000000    3.000000  ...                  0.000000   31.13750
max      1.000000    3.000000  ...                  6.000000  512.32920

[8 rows x 6 columns]



<class 'pandas.core.frame.DataFrame'>
RangeIndex: 887 entries, 0 to 886
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   survived                 887 non-null    int64
 1   pclass                   887 non-null    int64
 2   name                     887 non-null    object
 3   sex                      887 non-null    object
 4   age                      887 non-null    float64
 5   siblings/spouses aboard  887 non-null    int64
 6   parents/children aboard  887 non-null    int64
 7   fare                     887 non-null    float64
dtypes: float64(2), int64(4), object(2)
memory usage: 55.6+ KB
```

First table shows some standard computational values for each of the columns like mean, min, max etc. Second table displays the datatypes and Count of data.

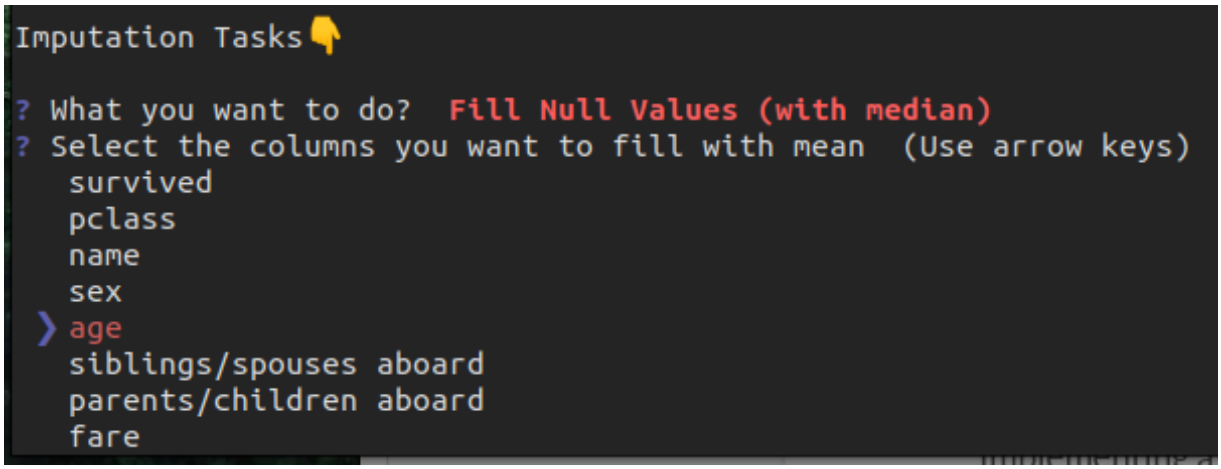
Dataframe correlation with heatmap:



Bivariate Analysis of Passenger Class and Age columns using Violin plot:

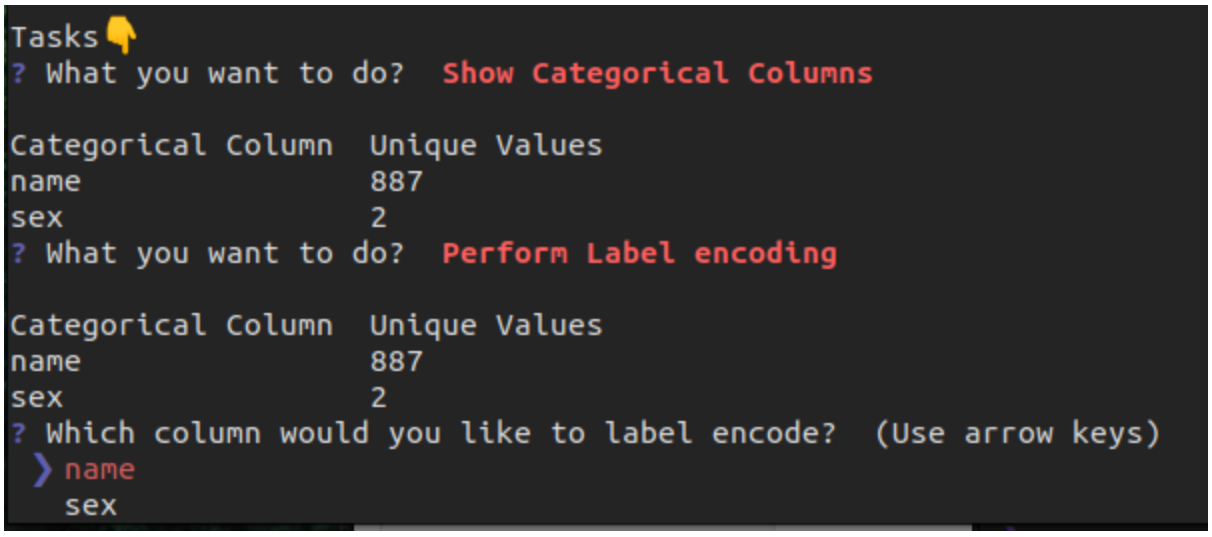


Null Value Handling:



Note: Final .csv dataset will be downloaded after all tasks. Scroll to Download section of the report to view it.

Character Encoding:



Feature Scaling:

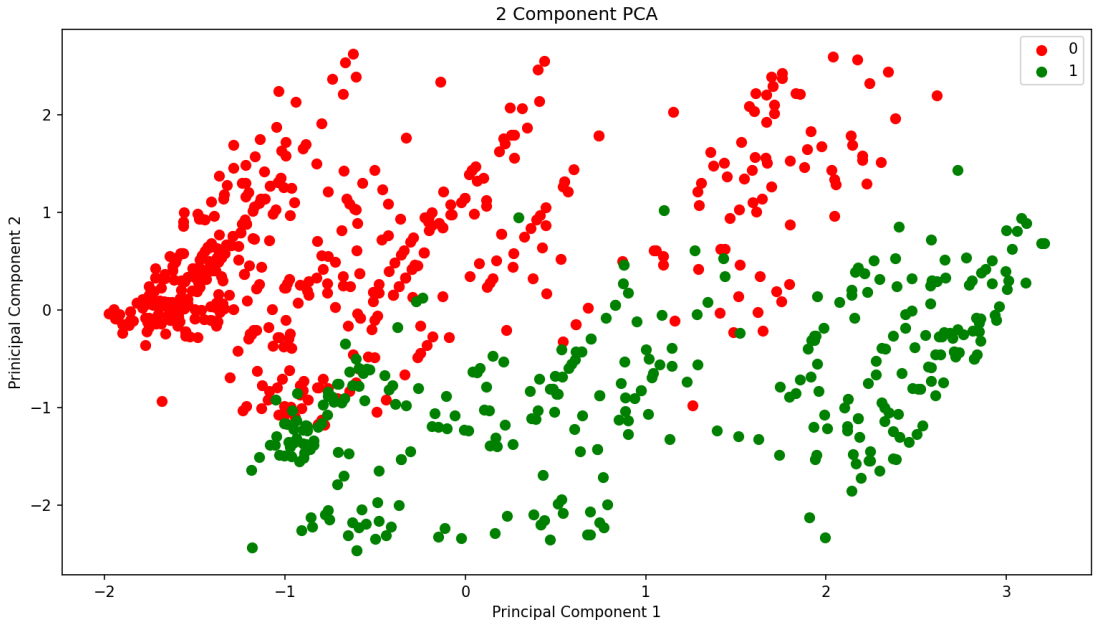
```
Tasks (Feature Scaling)👇
? Select columns for feature scaling   done (7 selections)
? Choose type of feature scaling  StandardScaler
StandardScaler
Scaling Done...👍
```

PCA:

	A	B	C	D
1		principal component 1	principal component 2	target
2	0	-1.75504313079934	0.103388346072129	-0.792163452691309
3	1	2.54954818134451	-0.256210610626372	1.26236573601393
4	2	-0.687720821353121	-0.950376689018834	1.26236573601393
5	3	2.32717795512711	-0.396326714394982	1.26236573601393
6	4	-1.21378758761001	0.877044727164616	-0.792163452691309
7	5	-1.3531265647203	0.34591124118311	-0.792163452691309
8	6	1.91412786850201	1.82765795861854	-0.792163452691309
9	7	-1.23302778118713	-1.03355604414737	-0.792163452691309
10	8	-0.38952353102548	-0.96627310627437	1.26236573601393
11	9	0.762766707982707	-1.71211157166765	1.26236573601393
12	10	-0.549812187140504	-2.19348688754182	1.26236573601393
13	11	2.40234214325151	0.848755643433065	1.26236573601393
14	12	-1.59409159869846	-0.070492751512609	-0.792163452691309
15	13	-0.148423475924948	0.886905523689901	-0.792163452691309
16	14	-1.77468454800187	-0.35947394568113	-0.792163452691309
17	15	1.27009454507126	0.609717921020877	1.26236573601393
18	16	-1.00319171761243	-1.08180120086097	-0.792163452691309
19	17	0.365559884391598	-1.11504965406095	1.26236573601393
20	18	-0.721635587427004	0.493545378873954	-0.792163452691309
21	19	-1.00425126708109	-1.16412455998387	1.26236573601393
22	20	0.360060634679749	0.836591448635573	-0.792163452691309
23	21	0.642144619728657	-0.425931487750244	1.26236573601393
24	22	-0.921220017973984	-1.55514194219676	1.26236573601393
25	23	1.95826456680256	-0.832571442956351	1.26236573601393
26	24	-1.12930850543573	-0.775136731780858	-0.792163452691309
27	25	0.610619182231921	-0.429037840728695	1.26236573601393
28	26	-1.66834874864606	0.365381373750764	-0.792163452691309
29	27	1.6469801083103	-0.217698343493885	-0.792163452691309
30	28	-0.765117799626008	-1.07422577223391	1.26236573601393
31	29	-1.55048275526139	0.107147363033246	-0.792163452691309
32	30	1.28853281440337	1.21180095620626	-0.792163452691309
33	31	3.00384769002012	0.208809142080944	1.26236573601393
34	32	-0.97150974168704	-1.4044899941408	1.26236573601393
35	33	0.403268254491416	2.46197004274016	-0.792163452691309

Data visualisation of PCA:

The clustering is not so differentiable owing to the nature of the Titanic dataset.



Downloading the modified dataset:

```
Enter the FILENAME you want? (without .csv) : titanic_new.csv
Hurray!! It is done....😁
Do you want to exit now? (y/n) y
```

.

Final Dataset after all the operations:

	A	B	C	D	E	F	G	H
1	survived	pclass	name	sex	age	siblings/spouses aboard	parents/children aboard	fare
2	-0.792163452691309	0.830523632917998	0.617055696758569	male	-0.529366007257325	0.429903948211424	-0.474980796742006	-0.503586345979705
3	1.26236573601393	-1.56127656967377	1.48015258906011	female	0.604264543188183	0.429903948211424	-0.474980796742006	0.783412448527398
4	1.26236573601393	0.830523632917998	-1.06227309821728	female	-0.245958369645948	-0.475855676642574	-0.474980796742006	-0.490019589521858
5	1.26236573601393	-1.56127656967377	1.44500384683969	female	0.39170881497965	0.429903948211424	-0.474980796742006	0.41794814823113
6	-0.792163452691309	0.830523632917998	1.12866516685586	male	0.39170881497965	-0.475855676642574	-0.474980796742006	-0.487507227214849
7	-0.792163452691309	0.830523632917998	0.078108316045389	male	-0.175106460243104	-0.475855676642574	-0.474980796742006	-0.479300846975235
8	-0.792163452691309	-1.56127656967377	0.999786445380972	male	1.73789509363369	-0.475855676642574	-0.474980796742006	0.393075761391743
9	-0.792163452691309	0.830523632917998	-1.60512589473273	male	-1.94640419531421	2.24142319791942	0.764160281817287	-0.22571907482453
10	1.26236573601393	0.830523632917998	1.55435548930323	female	-0.175106460243104	-0.475855676642574	2.00330136037658	-0.425536293605247
11	1.26236573601393	-0.365376468377885	1.54263924189642	female	-1.09618128248008	0.429903948211424	-0.474980796742006	-0.04491340409341
12	1.26236573601393	0.830523632917998	-0.976353950567355	female	-1.80470037650852	0.429903948211424	0.764160281817287	-0.313651755569838
13	1.26236573601393	-1.56127656967377	-1.28488179894664	female	2.02130273124507	-0.475855676642574	-0.474980796742006	-0.115677605777543
14	-0.792163452691309	0.830523632917998	1.14038141426267	male	-0.671069826063013	-0.475855676642574	-0.474980796742006	-0.487507227214849
15	-0.792163452691309	0.830523632917998	-0.691258597001687	male	0.675116452591027	0.429903948211424	5.72072459605446	-0.02071031057261
16	-0.792163452691309	0.830523632917998	-1.14038141426267	female	-1.09618128248008	-0.475855676642574	-0.474980796742006	-0.491442591532547
17	1.26236573601393	-0.365376468377885	1.25363847252848	female	1.80874700303654	-0.475855676642574	-0.474980796742006	-0.327720984489088
18	-0.792163452691309	0.830523632917998	-1.62465297374408	male	-1.94640419531421	3.14718282277341	0.764160281817287	-0.063922942253162
19	1.26236573601393	-0.365376468377885	-0.54285279651545	male	-0.458514097854481	-0.475855676642574	-0.474980796742006	-0.3880176798573
20	-0.792163452691309	0.830523632917998	1.51530133128054	female	0.108301177368273	0.429903948211424	-0.474980796742006	-0.287523187576947
21	1.26236573601393	0.830523632917998	1.3668955307943	female	-0.529366007257325	-0.475855676642574	-0.474980796742006	-0.504088818441107
22	-0.792163452691309	-0.365376468377885	0.253852027147512	male	0.39170881497965	-0.475855676642574	-0.474980796742006	-0.126731999928382
23	1.26236573601393	-0.365376468377885	0.363203669611056	male	0.320856905576806	-0.475855676642574	-0.474980796742006	-0.3880176798573
24	1.26236573601393	0.830523632917998	-1.44890926264196	female	-1.02532937307723	-0.475855676642574	-0.474980796742006	-0.487925284302735
25	1.26236573601393	-1.56127656967377	1.19115181969217	male	-0.10425455084026	-0.475855676642574	-0.474980796742006	0.064207535404288
26	-0.792163452691309	0.830523632917998	-0.820137318476578	female	-1.52129273889714	2.24142319791942	0.764160281817287	-0.22571907482453
27	1.26236573601393	0.830523632917998	1.29659804635345	female	0.604264543188183	0.429903948211424	5.72072459605446	-0.018449184496302
28	-0.792163452691309	0.830523632917998	-0.289000769367937	male	-0.245958369645948	-0.475855676642574	-0.474980796742006	-0.504088818441107
29	-0.792163452691309	-1.56127656967377	-0.554569043922258	male	-0.741921735465858	2.24142319791942	2.00330136037658	4.636706934160 34
30	1.26236573601393	0.830523632917998	-1.25754388833075	female	-0.387662188451636	-0.475855676642574	-0.474980796742006	-0.490940119071146
31	-0.792163452691309	0.830523632917998	0.359298253808787	male	-0.458514097854481	-0.475855676642574	-0.474980796742006	-0.490606477356775
32	-0.792163452691309	-1.56127656967377	-1.71838295299855	male	0.745968361993871	-0.475855676642574	-0.474980796742006	-0.092145815465176
33	1.26236573601393	-1.56127656967377	1.67542337917358	female	1.31278363721663	0.429903948211424	-0.474980796742006	2.29560332111601
34	1.26236573601393	0.830523632917998	-0.933394376742391	female	-0.812773644868702	-0.475855676642574	-0.474980796742006	-0.49353689675167

Note the Sex column has retained its values since we performed label encoding only for Name, which was then Standard Scaled.

Future Work:

Implementing a GUI using tkinter would be at the highest priority in the future. Not only making a GUI makes it easy for newbies to understand and apply preprocessing techniques, it also makes it simple to understand. Other than that, adding different complex preprocessing techniques would be the second highest priority.

REFERENCES:

- **Pandas documentation**
  Pandas library has been used to read ,write csv file and for handling data frames in the entire code

- **PyInquirer - PyPI documentation**
  PyInquirer library used for creating command line interface for our ML preprocessor which has features resembling GUI options

- **Scikit-learn documentation**
  Scikit-learn library is used for implementing feature scaling,character encoding and PCA

- **Matplotlib — Visualization with Python**
  Matplotlib library is used for creating plots of dataset

- **Seaborn: statistical data visualization**
  Seaborn library is used for creating plots of dataset

- **Statsmodels**
  Statsmodels library is used for getting stats of simple linear regression model

- **Pyfiglet**
  Pyfiglet library is used for printing with beautiful fonts in CLI

- **https://medium.com/geekculture/build-interactive-cli-tools-in-python-47303c50d75**
  Reference article used for understanding command line interface tools

- **https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d**
  Reference article used for understanding Data preprocessing