Assignment 2
Supervised Learning Competition

REPORT

TABLE OF CONTENTS

TEAM: = (3 MEMBERS)

1. Vidhan Dholakia (20151326)

SOFTWARE USED:

We have used Python 3.7 for the coding purpose with the PyCharm IDE. We also used TPU and GPU from Google Colab for acceleration purpose

Instructions on how to Download & Install the software and the data sets:

How to Download Python & Install

Step 1: = Go to https://www.python.org/downloads/

Step 2: = Select our version and directly download as per the requirement and system adaptability.

Step 3: = The installation process is quite simple and straightforward. Just need to follow the obvious steps. We have used PyCharm IDE for execution of Python.

Using Google Colab:

Step 1: Go to https://colab.research.google.com

Installing Pandas, Numpy, Keras, Matplotlib:

Pandas:

Install Python on your PC. Open the command prompt and type

pip install pandas

Numpy and Matplotlib can be similarly installed by using the following commands:

pip install numpy

pip install Matplotlib

For Keras, we need to first install the TensorFlow

pip install --upgrade TensorFlow

Now we can install Keras. All we have to do is import it.

import Keras

How to Download the Data Set?

We have been given the data set of the diabetes patients over a decade (1999 – 2008) at 130 US hospitals. It includes over 50 features representing patient and hospital outcomes.

ANALYSIS PROCESS USED:

We have the data (train and test) about the diabetes patients over a decade (1999 – 2008) of clinical care at 130 US hospitals and integrated delivery network. We have various categorical as well nominal data in the data set.

Initially, we had the total data of 90767 items! This data set is not useful yet as it is not clean, and it is totally unfit for the data modelling processes.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	encounte	patient_n	race	gender	age	weight	admission	discharge_	admission	time_in_h	payer_coc	medical_s	num_lab_	num_proc	num_med	number_c	number_e	number_i	diag_1	diag_2	diag_
2	5283	48330653	Caucasian	Female	[80-90)	?	2	1	4	13	?	?	68	2	28	0	0	0	398	427	
3	8499	63555809	Caucasian	Female	[90-100)	?	3	3	4	12	?	InternalM	33	3	18	0	0	0	434	198	
4	9441	42519137	Caucasian	Male	[40-50)	?	1	1	7	1	?	?	51	0	8	0	0	0	197	157	
5	20997	89868902	AfricanAm	Female	[40-50)	?	1	1	7	9	?	?	47	2	17	0	0	0	250.7	403	
6	28515	82637321	Caucasian	Male	[50-60)	?	2	1	2	3	?	?	31	6	16	0	0	0	414	411	
7	29661	77391041	AfricanAm	Male	[60-70)	?	2	1	4	7	?	?	62	0	11	0	0	0	157	288	
8	33687	85504775	Caucasian	Female	[40-50)	?	1	3	7	7	?	Family/Ge	60	0	15	0	1	0	428	250.43	2
9	35331	77586152	Caucasian	Male	[80-90)	?	1	6	7	10	?	Family/Ge	55	1	31	0	0	0	428	411	
10	48603	84259679	Caucasian	Male	[60-70)	?	3	1	2	4	?	?	70	1	21	0	0	0	414	411	V45
11	55017	49726661	AfricanAm	Female	[60-70)	?	3	1	2	1	?	?	49	5	2	0	0	0	518	998	
12	56529	1.15E+08	Caucasian	Male	[70-80)	?	1	1	7	5	?	?	73	0	12	0	0	0	428	492	
13	57171	86047745	AfricanAm	Female	[20-30)	?	1	1	7	2	?	?	11	5	13	2	0	1	648	250	V27
14	66339	86328689	AfricanAm	Male	[60-70)	?	1	3	7	12	?	?	75	5	13	0	0	0	999	507	
15	69837	92519222	AfricanAm	Male	[50-60)	?	1	1	7	4	?	?	45	4	17	0	0	0	410	411	
16	76983	1.09E+08	Caucasian	Female	[50-60)	?	1	1	7	3	?	Cardiolog	29	0	11	0	0	0	682	174	
17	82443	1.07E+08	AfricanAm	Male	[70-80)	?	1	1	7	5	?	?	35	5	23	0	0	0	402	425	
18	141291	69422081	?	Male	[70-80)	?	3	6	2	6	?	?	42	2	23	0	0	0	737	427	
19	141951	55629059	Caucasian	Female	[10-20)	?	1	1	7	3	?	?	59	0	18	0	0	0	276	250.01	
20	142767	22864001	?	Female	[50-60)	?	2	1	4	2	?	?	66	1	19	0	0	0	410	427	
21	142809	21239051	?	Male	[60-70)	?	2	1	4	2	?	?	36	2	11	0	0	0	572	456	
22	175557	62999978	AfricanAm	Female	[70-80)	?	2	1	4	2	?	?	47	0	12	0	0	0	410	401	
23	176691	1.07E+08	Caucasian	Female	[80-90)	?	2	6	1	11	?	?	42	2	19	0	0	0	V57	715	V43
24	208917	62718746	AfricanAm	Female	[70-80)	?	3	1	2	3	?	?	19	4	18	0	0	0	189	496	
25	214395	21861626	Other	Female	[50-60)	?	1	1	7	1	?	?	33	0	7	0	0	0	786	401	

Fig. Trained Data Un-Cleaned

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	encounte	patient_n	race	gender	age	weight	admission	discharge_	admission	time_in_h	payer_coc	medical_s	num_lab_	num_proc	num_med	number_c	number_e	number_i	diag_1	diag_2	diag_
2	1.69E+08	88565423	Caucasian	Male	[60-70)	?	3	1	1	2	MC	Radiologis	3	3	12	0	0	0	414	414	
3	1.69E+08	88590695	Caucasian	Male	[60-70)	?	5	1	1	1	BC	?	7	0	9	3	0	0	786	424	
4	1.69E+08	61086362	AfricanAm	Female	[70-80)	?	1	1	1	2	?	?	37	0	5	1	0	1	723	428	
5	1.69E+08	85993970	Caucasian	Male	[70-80)	?	1	1	7	2	?	InternalM	49	0	13	0	0	0	786	403	
6	1.69E+08	45884291	Caucasian	Male	[50-60)	?	2	1	2	5	?	Nephrolo	4	3	15	0	0	0	403	585	
7	1.69E+08	43898279	Caucasian	Female	[60-70)	?	3	1	1	8	MC	?	52	1	21	0	0	0	511	174	
8	1.69E+08	1.02E+08	AfricanAm	Female	[50-60)	?	2	3	1	6	?	Cardiolog	47	2	17	0	0	2	427	402	
9	1.69E+08	23346689	AfricanAm	Female	[40-50)	?	1	6	7	7	BC	Family/Ge	29	3	19	1	0	0	996	198	
10	1.69E+08	24053153	AfricanAm	Male	[50-60)	?	3	1	1	1	MC	Surgery-V	26	2	17	0	0	0	996	404	
11	1.69E+08	91316453	Hispanic	Male	[80-90)	?	1	1	7	3	?	?	83	0	20	0	0	0	427	303	
12	1.69E+08	88785761	Caucasian	Female	[20-30)	?	1	1	7	8	OG	Emergenc	75	1	22	3	2	12	250.11	540	
13	1.69E+08	94244882	Caucasian	Male	[70-80)	?	2	2	1	5	HM	?	1	0	4	0	0	0	577	576	
14	1.69E+08	92636789	Caucasian	Male	[50-60)	?	3	1	7	7	UN	Emergenc	63	0	13	0	0	0	969	967	E950
15	1.69E+08	85993475	Caucasian	Female	[60-70)	?	1	2	7	7	MC	?	56	1	25	0	0	0	410	398	
16	1.69E+08	29135660	Caucasian	Male	[50-60)	?	1	1	2	2	?	?	58	0	8	0	0	0	933	414	V45
17	1.69E+08	43943639	Caucasian	Female	[70-80)	?	3	1	1	2	MD	Cardiolog	13	0	13	0	0	0	922	401	
18	1.69E+08	86869958	AfricanAm	Female	[40-50)	?	1	1	7	7	HM	?	53	3	34	0	0	2	536	276	
19	1.69E+08	32486882	Caucasian	Female	[80-90)	?	2	1	1	2	HM	?	32	0	5	0	0	0	558	276	
20	1.69E+08	71416778	Caucasian	Female	[70-80)	?	1	1	7	3	MC	?	58	1	22	0	0	0	486	276	
21	1.69E+08	84562493	Caucasian	Female	[70-80)	?	1	6	7	7	MC	?	44	1	20	1	0	4	996	403	
22	1.69E+08	1.01E+08	Caucasian	Male	[80-90)	?	1	6	7	6	MC	?	48	0	10	0	0	4	584	276	
23	1.69E+08	24256373	AfricanAm	Female	[60-70)	?	3	1	1	2	MC	Orthoped	26	5	32	1	1	0	736	727	
24	1.69E+08	24724202	Caucasian	Female	[60-70)	?	3	6	1	4	MC	Orthoped	38	2	33	0	0	0	715	250	
25	1.69E+08	1.01E+08	Caucasian	Female	[70-80)	?	5	6	1	5	MC	?	65	0	28	0	0	0	427	486	

Fig. Test Data Un-Cleaned

DATA PREPARATION & DATA CLEANING: =

We have updated our data as per our requirements. Firstly, we have cleaned the data by dealing with the missing values and bifurcating between the relevant and irrelevant features.

We changed the 'change' column values to 0 if value is 'No' else 1 if value is 'Ch'

Discharge_Disposition_ID column value were changed from

- (1,6,7,8,13) were changed to 3
- (2,5,10,14,16,22,23,24,27,28,29,30) were changed to 1
- (3,4,9,12,15,17) were changed to 2
- (11,18,19,20,21,25,26) were changed to 4

Admission_type_id column values were changed from

- 1,2,7,3,4,5,6,8 to 1

Admission_Source_Id column values were changed from

- 2,3 to 1
- 5,6,10,18,22,25,26 to 2
- 9,15,17,20,21,11,13,14 to 3

Fore the *drugs* value, we changed

- None -99
- Norm to 0
- >200 to 1
- >300 to 1
- >7 to 1
- >8 to 1
- No to 0
- Steady to 1
- Up to 1
- Down to 1

The *age* range were change from

- 10-20 to 15
- 20-30 to 25
- 30-40 to 35
- 40-50 to 45
- 50-60 to 55
- 60-70 to 65
- 70-80 to 75
- 80-90 to 85
- 90-100 to 95

We also changed the *race* categorical values to numerical values as shown below:

- Asian' to 2
- Other to 3
- Hispanic' to 4
- AfricanAmerican to 5

In the *diag_1, diag_2, diag_3* columns, if any value started with V, we assigned those values as 0

If the *diag_1* value were

- >=390 and <460 or ==785: changed to 1
- >=460 and <520 or ==786: changed to 2
- >=520 and <580 or ==787: changed to 3
- >=800 and <999 changed to 5
- >=710 and <740 changed to 6
- >=580 and <630 or ==788: changed to 7
- >=140 and <240 or ==780/781/784: changed to 8
- >=790 and <800 : changed to 8
- >=240 and <280 : changed to 8
- >=680 and <710 or ==782 /781/784: changed to 8

Same changes were applied to *diag_2* and *diag_3* as well.

The above operations were applied to the test as well as train data:

As you can compare from the diagram 2 and diagram 4, we have removed the irrelevant features like “encounter id”, “weight”, “payer code” as they had many values as missing etc. The records were also eliminated who had missing values from the columns having few missing values as these columns couldn’t be removed.

Also, we have applied various feature engineering approaches on the relevant features as shown in the diagram 4. We have standardized our data set on both categorical as well numeric data. However, before this we had converted our categorical to numerical data.

The labels were initially labeled as 0 for patients not admitted. For patients admitted within 30 days and after 30 days, we labelled them as 1.

We also removed the records where patients were not admitted and then in the new data, we labelled the patients admitted before 30 days as 2 and after 30 days as 1. In this way, we won’t leak data of those patients who won’t be admitted into the hospital again. This helped us to reduce data and time as well and achieve accuracy higher than with compiled data. We also added 3 new columns. Which were the product of

- *time_in_hospital* and *num_medications*
- *time_in_hospital* and *num_lab_procedures*
- *time_in_hospital* and number_diagnoses

The data was then standardized for 5 columns namely

- *time_in_hospital*
- *num_lab_procedures*
- *num_medications*
- *number_diagnoses*
- *num_procedures*

The above procedures were carried out in MS-Excel and were not done in Python. As we can compare from the diagram 3 and diagram 5, we have removed the irrelevant features like “weight”, “payer code”, etc.

	patient_nbr	race	gender	age	admission	discharge_	admission	time_in_h	num_lab_	num_proc	num_med	number_c	number_e	number_i	diag_1	diag_2	diag_3	number_c	max_glu_	A1Cresult
2	48330653	1	1	85	1	2	4	2.849256	1.265139	0.378287	1.479305	0	0	0	1	1	0	0.284946	0	0
3	63555809	1	1	95	2	2	4	2.516486	-0.49864	0.963839	0.245092	0	0	0	1	8	2	0.284946	0	0
4	42519137	1	0	45	1	2	7	-1.14398	0.408444	-0.79282	-0.98912	0	0	0	8	8	4	-1.34022	0	0
5	89868902	5	1	45	1	2	7	1.518177	0.206869	0.378287	0.12167	0	0	0	4	1	5	0.826668	0	0
6	82637321	1	0	55	1	2	1	-0.47844	-0.59943	2.720493	-0.00175	0	0	0	1	1	4	0.826668	0	0
7	77391041	5	0	65	1	2	4	0.852637	0.962776	-0.79282	-0.61886	0	0	0	8	0	8	-0.25678	0	0
8	85504775	1	1	45	1	2	7	0.852637	0.861988	-0.79282	-0.12517	0	1	0	1	4	4	0.284946	0	0
9	77586152	1	0	85	1	2	7	1.850947	0.610019	-0.20726	1.849569	0	0	0	1	1	1	0.284946	0	0
10	84259679	1	0	65	2	2	1	-0.14567	1.365926	-0.20726	0.615356	0	0	0	1	1	0	-0.25678	0	0
11	49726661	5	1	65	2	2	1	-1.14398	0.307656	2.134941	-1.72965	0	0	0	2	5	7	0.284946	0	0
12	114882854	1	0	75	1	2	7	0.187097	1.517108	-0.79282	-0.49544	0	0	0	1	2	4	0.284946	0	0
13	86047745	5	1	25	1	2	7	-0.81121	-1.60731	2.134941	-0.37202	2	0	1	0	4	0	-0.7985	0	0
14	86328689	5	0	65	1	2	7	2.516486	1.617895	2.134941	-0.37202	0	0	0	5	2	5	0.826668	0	0
15	92519222	5	0	55	1	2	7	-0.14567	0.106081	1.54939	0.12167	0	0	0	1	1	1	0.284946	0	0
16	108662531	1	1	55	1	2	7	-0.47844	-0.70022	-0.79282	-0.61886	0	0	0	8	8	4	-2.42367	0	0
17	107389193	5	0	75	1	2	7	0.187097	-0.39786	2.134941	0.862199	0	0	0	1	1	1	0.826668	0	0
18	55629059	1	1	15	1	2	7	-0.47844	0.811594	-0.79282	0.245092	0	0	0	8	4	8	0.826668	0	0
19	62999978	5	1	75	1	2	4	-0.81121	0.206869	-0.79282	-0.49544	0	0	0	1	1	7	0.284946	0	0
20	107400632	1	1	85	1	2	1	2.183717	-0.0451	0.378287	0.368513	0	0	0	0	6	0	0.284946	0	0
21	62718746	5	1	75	2	2	1	-0.47844	-1.20416	1.54939	0.245092	0	0	0	8	2	1	-0.7985	0	0
22	21861626	3	1	55	1	2	7	-1.14398	-0.49864	-0.79282	-1.11254	0	0	0	2	1	4	-2.42367	0	0
23	40523171	1	0	85	1	2	7	0.519867	1.063564	0.963839	0.245092	0	0	0	1	1	1	-0.25678	0	1
24	115196648	1	1	55	1	2	1	-0.81121	-0.9018	0.378287	-0.61886	0	0	0	5	7	4	-2.42367	0	0
25	41605934	1	0	25	1	2	1	1.850947	0.509232	-0.79282	0.491935	0	0	0	8	4	8	-0.7985	0	0

Fig. Cleaned Train Data

	encounte	patient_n	race	gender	age	admission	discharge_	admission	time_in_h	num_lab_	num_proc	num_med	number_c	number_e	number_i	diag_1	diag_2	diag_3	number_c	max_glu_	A1Cr
2	1.69E+08	88565423	1	0	65	2	2	1	-0.7938	-2.28102	0.97762	-0.61243	0	0	0	1	1	1	-0.48417	-99	
3	1.69E+08	88590695	1	0	65	2	2	1	-1.1396	-2.06473	-0.77287	-0.9824	3	0	0	2	1	1	0.107716	-99	
4	1.69E+08	61086362	5	1	75	1	2	1	-0.7938	-0.44254	-0.77287	-1.4757	1	0	1	6	1	1	0.699606	-99	
5	1.69E+08	85993970	1	0	75	1	2	7	-0.7938	0.206331	-0.77287	-0.4891	0	0	0	2	1	1	0.699606	-99	
6	1.69E+08	45884291	1	0	55	1	2	1	0.243601	-2.22695	0.97762	-0.24246	0	0	0	1	7	0	0.699606	-99	
7	1.69E+08	43898279	1	1	65	2	2	1	1.281001	0.36855	-0.18938	0.497487	0	0	0	2	8	8	0.107716	-99	
8	1.69E+08	1.02E+08	5	1	55	1	2	1	0.589401	0.098186	0.394122	0.004191	0	0	2	1	1	1	0.107716	-99	
9	1.69E+08	23346689	5	1	45	1	2	7	0.935201	-0.87513	0.97762	0.250839	1	0	0	5	8	8	-0.48417	-99	
10	1.69E+08	24053153	5	0	55	2	2	1	-1.1396	-1.03734	0.394122	0.004191	0	0	0	5	1	1	0.699606	-99	
11	1.69E+08	91316453	4	0	85	1	2	7	-0.448	2.044808	-0.77287	0.374163	0	0	0	1	0	1	-2.25985	-99	
12	1.69E+08	88785761	1	1	25	1	2	7	1.281001	1.612225	-0.18938	0.620811	3	2	12	4	3	8	0.699606	-99	
13	1.69E+08	94244882	1	0	75	1	1	1	0.243601	-2.38917	-0.77287	-1.59902	0	0	0	3	3	3	-0.48417	-99	
14	1.69E+08	92636789	1	0	55	2	2	7	0.935201	0.963351	-0.77287	-0.4891	0	0	0	5	5	0	-1.07607	-99	
15	1.69E+08	85993475	1	1	65	1	1	7	0.935201	0.584841	-0.18938	0.990783	0	0	0	1	1	7	0.699606	-99	
16	1.69E+08	29135660	1	0	55	1	2	1	-0.7938	0.692987	-0.77287	-1.10573	0	0	0	5	1	0	-0.48417	-99	
17	1.69E+08	43943639	1	1	75	2	2	1	-0.7938	-1.74029	-0.77287	-0.4891	0	0	0	5	1	1	-0.48417	-99	
18	1.69E+08	86869958	5	1	45	1	2	7	0.935201	0.422623	0.97762	2.100699	0	0	2	3	8	1	0.699606	-99	
19	1.69E+08	32486882	1	1	85	1	2	1	-0.7938	-0.71291	-0.77287	-1.4757	0	0	0	3	8	8	-0.48417	-99	
20	1.69E+08	71416778	1	1	75	1	2	7	-0.448	0.692987	-0.18938	0.620811	0	0	0	2	8	1	0.699606	-99	
21	1.69E+08	84562493	1	1	75	1	2	7	0.935201	-0.06403	-0.18938	0.374163	1	0	4	5	1	7	0.699606	-99	
22	1.69E+08	1.01E+08	1	0	85	1	2	7	0.589401	0.152259	-0.77287	-0.85908	0	0	4	7	8	1	0.699606	-99	
23	1.69E+08	24256373	5	1	65	2	2	1	-0.7938	-1.03734	2.144616	1.854051	1	1	0	6	6	1	-2.25985	-99	
24	1.69E+08	24724202	1	1	65	2	2	1	-0.1022	-0.38847	0.394122	1.977375	0	0	0	6	4	0	-2.25985	-99	
25	1.69E+08	1.01E+08	1	1	75	2	2	1	0.243601	1.071497	-0.77287	1.360755	0	0	0	1	2	1	0.699606	-99	

Fig. Cleaned Test Data

APPROACHES USED AND REVIEWS:

Our approach was divided into two categories.

- We trained the model to predict whether a patient will be admitted or not. This was done by setting the column 'readmitted' as 0 for 'NO' and 1 for '>30' OR '<30'. Initially, we were only concerned with predicting whether a patient will go to hospital or not. To predict this, we used Random Forest algorithm with 10 trees with depth of 55. The impurity criteria were 'gini'. With the Random forest algorithm, we were able to get a maximum accuracy of 87% as shown in the screenshot below. The output of the code whether a patient was admitted or not is saved as ***Admitted Or Not.csv***

- To predict whether a patient was admitted before 30 or after 30 days, we separated the patients who were admitted from the training data. This way, we won't get any leaked data from the patients who didn't go the hospital at all. This also gave us benefit of reduced data which inherently reduced our time. The new dataset had 40266 rows only. Now we trained the new dataset using a neural network. We have 45 features, which is the input layer. We decided to go with two hidden layers. The first layer consisted of 250 neurons and the second layer consisted with 140 neurons. We ran the code for 380 epochs with a batch size of 180. With the above approach, we got an output accuracy of 91% in average and the highest accuracy was 94.52%. The patients readmitted were saved into a separate file named ***Patients_admit_30_days.csv*** The labels for these were save separately in a file named ***Patients_admit_30_days_labels.csv.*** We have added the files in the zip as well. The python code for the above process is named ***Within 30 Code Neural Network.py.*** The output of the code is saved as ***Output_2_Within_30_days.csv. The output label 1 indicates patients admitted after 30 days and output label 2 indicates admitted within 30 days.***

EVALUATION AND ACCURACY:

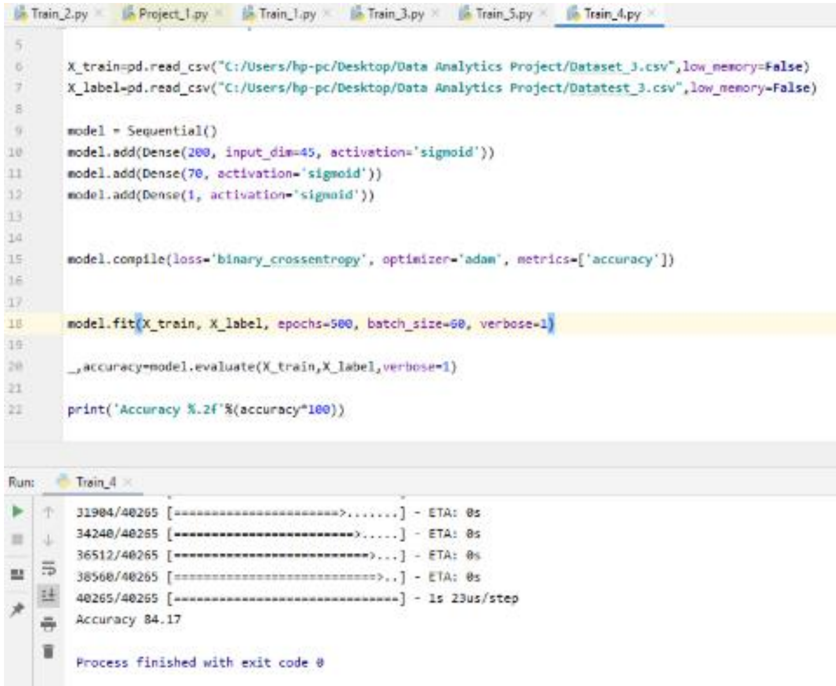
a. Predicting if the person is going to be readmitted or not

```
Train_2.py ×  Project_1.py ×  Train_1.py ×  Train_3.py ×  Train_5.py ×  Train_4.py ×

    rm_prd = rm.predict(X_trai)

    print("Train Data Accuracy")
    print("Accuracy is {0:.2f}".format(accuracy_score(y_trai, rm_prd)))
    print("Precision is {0:.2f}".format(precision_score(y_trai, rm_prd)))
    print("Recall is {0:.2f}".format(recall_score(y_trai, rm_prd)))

    return rm

X_train = pd.read_csv("C:/Users/hp-pc/Desktop/Data Analytics Project/Train_C2T1.csv", low_memory=False)
X_test = pd.read_csv("C:/Users/hp-pc/Desktop/Data Analytics Project/Data_Test.csv", low_memory=False)
rm = transform(X_train, False)
input_t, patien = transform(X_test, True)
pre = np.asarray(rm.predict(input_t)).reshape(-1, 1)
patien = patien.reshape(-1, 1).astype(int)
out = np.append(patien, pre, 1).astype(int)
np.savetxt('output.csv', out, delimiter=',')
print(out)
print(pre.shape)
transform()

Run:  Train_5 ×
Train Data Accuracy
Accuracy is 0.87
Precision is 0.89
Recall is 0.82
[[ 88565423         0]
 [ 88590695         0]
 [ 61086362         1]
 ...
 [ 89262995         1]
```

Fig. Training whether a patient will be admitted or not suing Random Forest

The accuracy obtained was 87% whether a person will be admitted or not.

b. Predicting if a person will be readmitted in less than 30 OR Greater than 30 days.



The below snapshot is of the code and the output of the Neural Network we used to predict the patients, whether they were admitted before 30 days or after 30 days. The input layer is of 45 node which are the features of the patients i.e. columns from test data. The first hidden layer node has 250 neurons. The second hidden layer node has 140 neurons. The final layer has 1 output.

```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam

X_train_x=pd.read_csv("Patients_admit_30_days.csv",low_memory=False)
X_label=pd.read_csv("Patients_admit_30_days_labels.csv",low_memory=False)
patients = X_train_x['patient_nbr2'].copy()
X_train = X_train_x.drop(['patient_nbr2'], 1)
model = Sequential()
x = 250
y = 140
ep = 380
b_s = 180
model.add(Dense(x, input_dim=45, activation='relu'))
model.add(Dense(y, activation='sigmoid'))
model.add(Dense(1, activation='sigmoid'))


model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])


model.fit(X_train, X_label, epochs=ep, batch_size=b_s, verbose=0)

_,accuracy=model.evaluate(X_train,X_label,verbose=0)

print('Accuracy %.2f'%(accuracy*100),'\nHidden Layer 1 Neurons',x,'\nHidden Layer 2 Neurons',y,'\nEpochs',ep,
      '\nBatch Size',b_s)

new = model.predict_classes(X_train)
patients = patients.to_numpy()
#print(new.shape)
patients = np.asarray(patients).reshape(-1, 1)
output_2 = np.append(patients, new, 1) + 1
np.savetxt('Patients_within_after_30_days.csv', output_2, delimiter=',')
```

Fig. Neural Network Code to predict patients within 30 or after 30 days.

```
============ RESTART: C:\Users\Shashi Suman\Downloads\Train_4.py =
Using TensorFlow backend.
Accuracy 94.52
H_Layer_1 250
H_Layer_2 140
Epochs 380
Batch 180
```

Fig. Output Of NN with an accuracy of 95%

Training the separate data of admitted patients gave us an accuracy of 94.52% and an average of 91%.

We have 2 output files. The output of the code whether a patient was admitted or not is saved as ***Admitted Or Not.csv.*** The output of the code whether a patient was admitted within 30 or after 30 days is saved as ***Output_2_Within_30_days.csv***

.

REFERENCES:

[1] https://www.python.org/doc/
[2] https://keras.io/
[3] https://towardsdatascience.com/predicting-hospital-readmission-for-patients-with-diabetes-using-scikit-learn-a2e359b15f0
[4] https://medium.com/berkeleyischool/how-to-use-machine-learning-to-predict-hospital-readmissions-part-1-bd137cbdba07