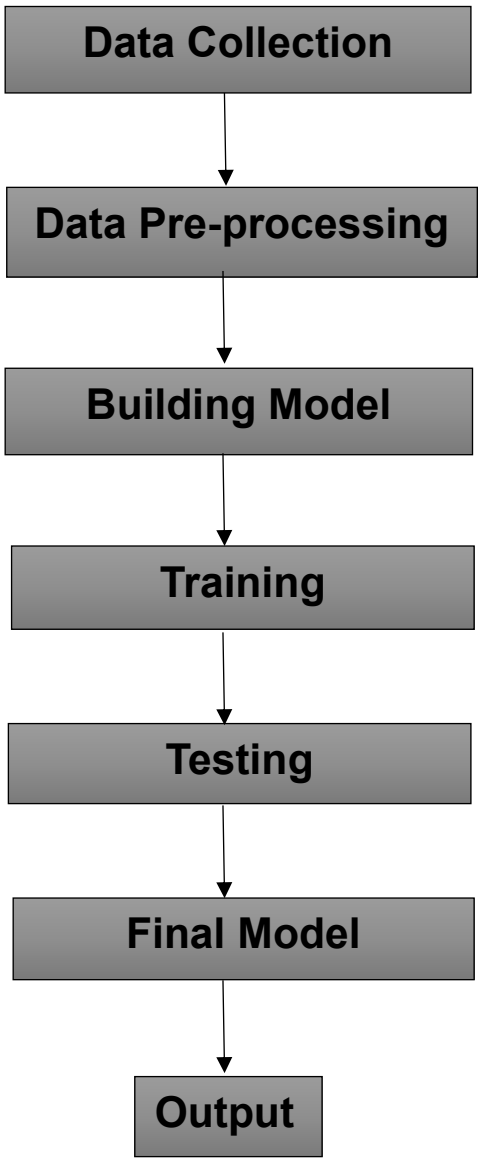
Acknowledgement

I would like to express my deep gratitude to my project guide Prof. Subhashis Chatterjee, Department of Maths and Computing, IIT Dhanbad for his guidance and unsurpassed knowledge and immense encouragement. I am also grateful to Mr. Deep Jyoti Saha, Teaching Assistant for the guidance in building this project.

I would like to thank my parents, friends and classmates for their encouragement throughout my project period. At last, but not least, I thank everyone for supporting directly or indirectly in completing this project successfully.

INDEX

PROJECT FLOW CHART



Churn Prediction (Business Analytics)

Objective- Prediction of customers which are at high risk of leaving the company.

Dataset – https://www.kaggle.com/datasets/blastchar/telco-customer-churn/download?datasetVersionNumber=1

Each row represents a customer, each column contains customer's attributes described as

The data set includes information about:

- Customers who left within the last month – the column is called Churn
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
customerI	gender	SeniorCitiz	Partner	Dependen	tenure	PhoneServ	MultipleLi	InternetSe	OnlineSec	OnlineBac	DevicePro	TechSuppc	Streaming	StreamingI	Contract	PaperlessB	P
7590-VHVI	Female	0	Yes	No	1	No	No phone	DSL	No	Yes	No	No	No	No	Month-to-	Yes	E
5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year	No	M
3668-QPYI	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-	Yes	M
7795-CFO(	Male	0	No	No	45	No	No phone	DSL	Yes	No	Yes	Yes	No	No	One year	No	B
9237-HQIT	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-	Yes	E
9305-CDSI	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	Month-to-	Yes	E
1452-KIOV	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	No	Month-to-	Yes	C

The raw data contains 7043 rows (customers) and 21 columns (features).

The "Churn" column is our target.

Data Pre-processing

In the provided dataset all the categorical data are replaced by creating dummies.

Dropping the irrelevant columns which doesn't contribute to the prediction.

Dividing the dataset into training and testing part for learning and evaluating the working of deep learning model.

Deep Learning Model Implementation –

```
from keras.models import Sequential
from keras.layers import Dense


model=Sequential()

##input layer
model.add(Dense(units=5,input_dim=9,kernel_initializer='normal',activation='relu'))

#second layer
model.add(Dense(units=5,kernel_initializer='normal',activation='tanh'))

#output layer
model.add(Dense(1,kernel_initializer='normal'))


#compiling
model.compile(loss='binary_crossentropy',optimizer='adam')
```

The models of deep learning are build using keras module. It provide the feature of adding layer according to the requirement with a facility of different activation function between different layers according to your preference.

The model contains an input layer, hidden layer and output layer. Loss function used in the model is "binary cross entropy" with "Adam optimizer".

Result

```
# Predict on the test set
y_pred = model.predict(X_test)

accuracy_score(y_test, np.round(abs(y_pred)), normalize=False)/len(y_test)
```

```
67/67 [==============================] - 0s 2ms/step

0.7477520113582584
```

Brain Tumour Detection (Health Care)

Objective – Prediction of presence of brain tumour from given MRI scan of brain.

Dataset- https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri/download?datasetVersionNumber=2

The folder contains MRI images of brain of patients having brain tumour and of the patients not having brain tumour. The data is divided into training and testing images. Number of images in each of the folder are as follows: -

Training data

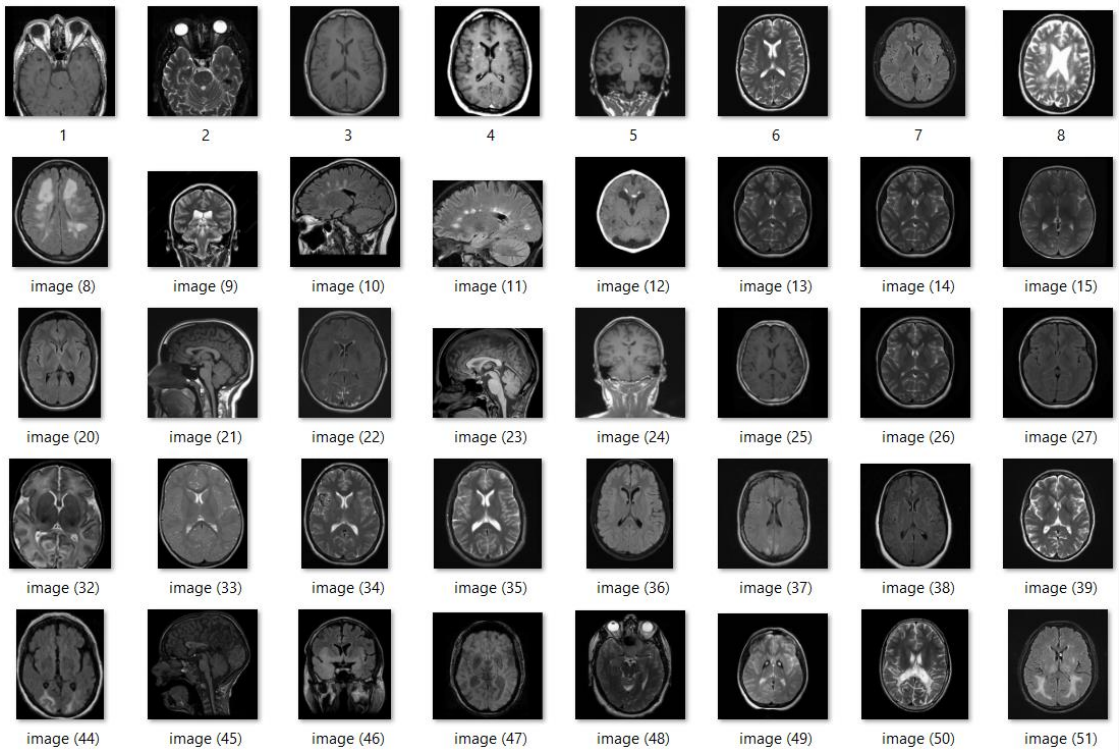
Normal -395

Tumour -822

Testing data

Normal -105

Tumour -115



Data Pre-processing

In this dataset the images are converted into numerical value. Image is divided into pixel, these pixels have colours which is represented using (R, G, B) value between 0-255.

After pre-processing the data of a single pixel looks like-

```
[0.02352941]
[0.02352941]
[0.02352941]]
```

The numerical values are scaled bringing the value between 0-1.

Deep Learning Model

```
input_shape = (64,64,1)
model = models.Sequential()
model.add(Conv2D(32,kernel_size = (2,2),strides = (1,1),activation = 'linear',input_shape = input_shape))
model.add(MaxPooling2D(pool_size = (2,2),strides = (2,2)))
model.add(Conv2D(64,kernel_size = (2,2),strides = (1,1),activation = 'linear'))
model.add(MaxPooling2D(pool_size = (2,2),strides = (2,2)))
model.add(Conv2D(128,kernel_size = (2,2),strides = (1,1),activation = 'linear'))
model.add(MaxPooling2D(pool_size = (2,2),strides = (2,2)))
model.add(Conv2D(256,kernel_size = (2,2),strides = (1,1),activation = 'linear'))
model.add(MaxPooling2D(pool_size = (2,2),strides = (2,2)))
model.add(Conv2D(512,kernel_size = (2,2),strides = (1,1),activation = 'linear'))
model.add(MaxPooling2D(pool_size = (2,2),strides = (2,2)))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(256, activation="linear"))
model.add(Dense(1, activation="sigmoid"))
model.summary()
```

This deep learning model uses CNN which is very helpful for dataset containing images.

In deep learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep neural networks, most commonly applied to analyse visual imagery. Now when we think of a neural network, we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution.

Result

```
Epoch 20/20
244/244 [==============================] - 14s 57ms/step - loss: 2.2884e-06 - accuracy: 1.0000 - val_loss: 0.0170 - val_accurac
```

Accuracy 0.95 that is 95 percent

T20 Score Prediction (Sports Analytics)

Objective- Prediction of the total score made by a team using previous data.

Dataset-https://www.kaggle.com/datasets/veeralakrishna/cricsheet-a-retrosheet-for-cricket/download?datasetVersionNumber=1

The data is provided in a number of zip files, one of which contains all of the matches, and the others certain sub-sets of matches, such as for the type of matches, matches for certain countries, teams, or genders, or periods of time.

The dataset used in this project contains data of 332 matches. Every ball delivered in the each of those matches is calibrated in the dataset with its specification.

	innings	meta.data_version	meta.created	meta.revision	info.dates	info.gender	info.match_type	info.outcome.by.wickets	info.outcome.winner	info.overs
0	[{'1st innings': {'team': 'Australia', 'delive...	0.9	2017-02-18	2	[2017-02-17]	male	T20	5.0	Sri Lanka	2(
0	[{'1st innings': {'team': 'Australia', 'delive...	0.9	2017-02-19	2	[2017-02-19]	male	T20	2.0	Sri Lanka	2(
0	[{'1st innings': {'team': 'Australia', 'delive...	0.9	2017-02-23	1	[2017-02-22]	male	T20	NaN	Australia	2(
0	[{'1st innings': {'team': 'Hong Kong', 'delive...	0.9	2016-09-12	1	[2016-09-05]	male	T20	NaN	Hong Kong	2(
0	[{'1st innings': {'team': 'Zimbabwe', 'deliver...	0.9	2016-06-19	1	[2016-06-18]	male	T20	NaN	Zimbabwe	2(

Data Pre-processing

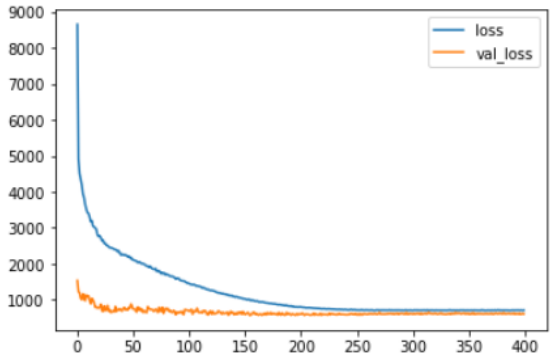
In Pre-processing dummies are created for all the categorical data columns. For teams labels is used to encode them into numerical values.

Redundant features are drop as they doesn't contribute int the prediction.

Deep Learning Model

```
model = Sequential()

model.add(Dense(43, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(22, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(11, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1))

model.compile(optimizer='adam', loss='mse')
```

Result



From the above graph we can say the loss is minimized which end up in predicting better result from the model.

	Predict	Actual
0	195.570633	194
1	154.791687	153
2	156.365829	178
3	172.642624	194
4	143.304367	166

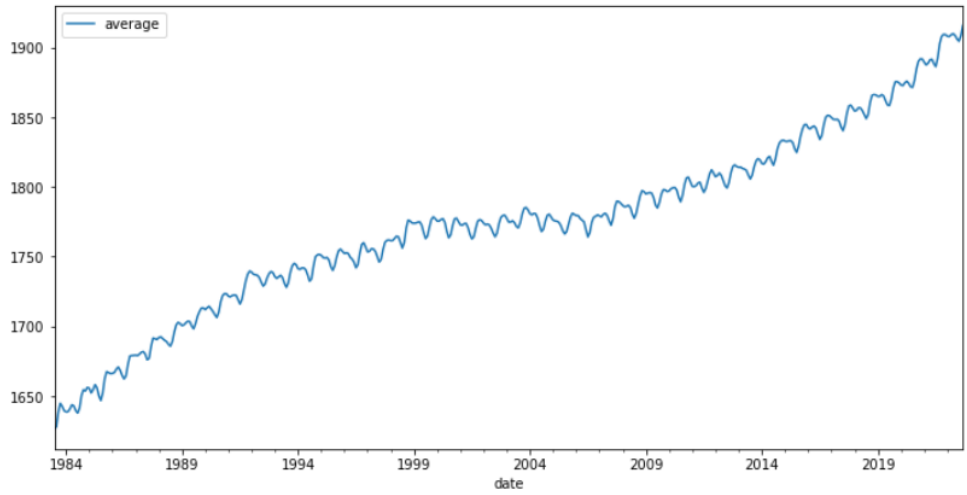
Methane Observation (Time Series)

Objective-Prediction of Methane (CH4) in parts per billions content in the air from the given data.

Dataset-https://www.kaggle.com/datasets/jarredpriester/global-monthly-methane-observations/download?datasetVersionNumber=4

average	date
1626.58	1983/7
1627.88	1983/8
1638.49	1983/9
1644.8	1983/10
1642.58	1983/11
1639.52	1983/12
1638.67	1984/1
1638.77	1984/2
1640.76	1984/3
1643.72	1984/4
1642.91	1984/5
1639.62	1984/6
1637.77	1984/7
1641.36	1984/8
1650.43	1984/9
1654.49	1984/10

The data contains average methane (in ppbs.) content in air for every month from 1983 to 2022.



Increase in the methane content in given period of time.

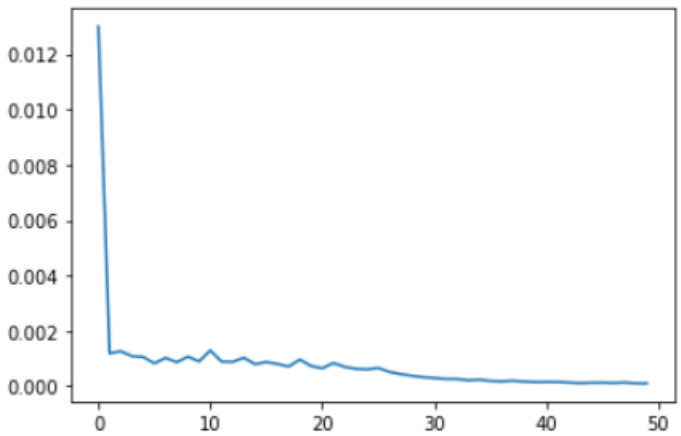
Deep Learning Model

```
# define model
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

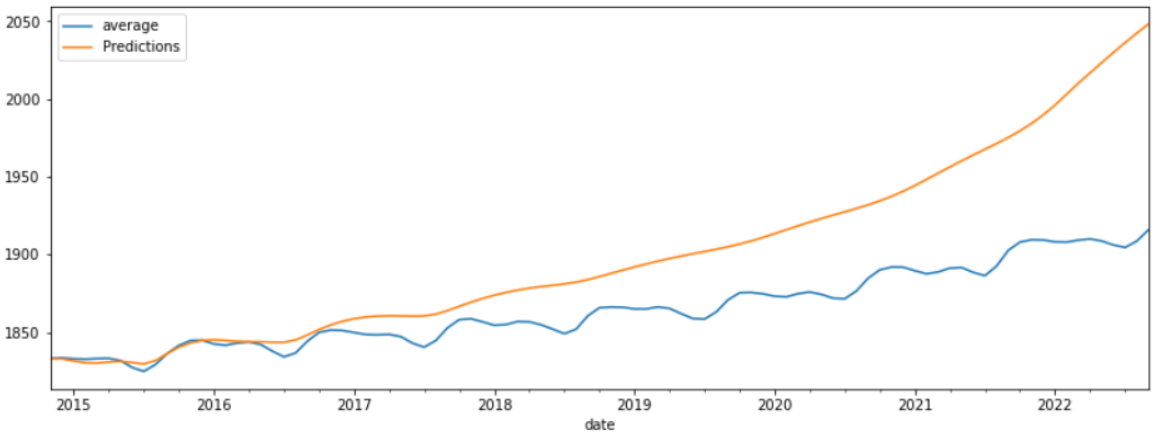
In it RNN is used for building the model. A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition etc.

LSTM is a type of RNN with higher memory power to remember the outputs of each node for a more extended period to produce the outcome for the next node efficiently.

Result



Graph representing loss function value for each iteration in training the model



Graph of the predicted and actual value over the test sample

References

- Kaggle: - https://www.kaggle.com/
- Geeks for geeks – https://www.geeksforgeeks.org/
- Analytic vidya - https://www.analyticsvidhya.com/
- Towards Science - https://towardsdatascience.com/