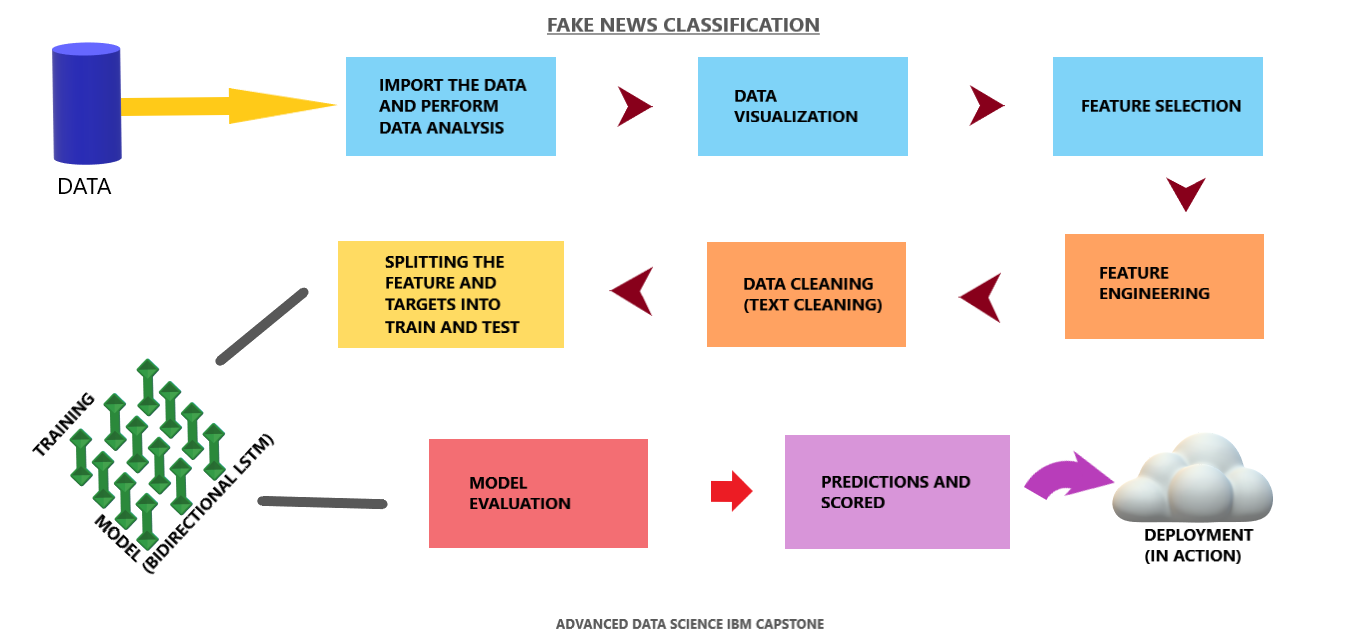
**Architectural Decisions Document**

IBM Coursera Advanced Data Science Capstone

* The data science project process



* Data Source (Data Collection):

Data set and use case:

* + - The data is used from the Kaggle website you can download it from here <https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset>
    - The data is in CSV format. The data is almost in a structured form. The data quality is also good.

Justification:

* This dataset was chosen because it was an amazing and interesting dataset and I felt to do the project on this data. The data was a little structured and was in CSV format.
* Extraction transformation and loading:
* Technology Choice:

I used the Numpy, Pandas to play with the dataset.

* Justification:

The Numpy and pandas are very helpful and easy data handling libraries which makes our work more easy and fast. **Pandas** is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures, and data analysis tools. The development of **NumPy and pandas** libraries has extended python's multi-purpose nature to solve machine learning problems

* Technology choice

IBM Watson studios, Jupyter Notebooks, Tensorflow, Keras, Seaborn, Matplotlib, Numpy, Pandas, sci-kit-learn, Github.

* Data Quality Assessment:

The data analysis was done using pandas and there where four-column from which the single text column was usable.

Data Visualization:

The data was visualized for the sake of finding the Nan and null values in the data frame

* Feature selection:

Technologies :

Pandas, Numpy

Justification:

There where 4 features the title, Datetime, News, and the news type, So I found the news feature important and else of no use as they will not contribute in a news to be fake or true so I dropped them.

* Feature engineering and Data cleaning:

Technologies used:

Pandas, Numpy, NLP(Tensorflow, Keras)

Justification:

The data was a text data used for the classification weather it’s a fake or true news data classification problem so I used certain NLP (natural language processing) methods and techniques like the tokenizer and techniques to create sequences. A second feature was created as a class column which denotes whether the news is fake or true. The text data that is the news has to be clean which consists of the special characters and

* Feature and target:

Technologies:

Sklearn

Justification:

The features and targets are distributed as news and classes and then split to train and text.

* Model Selection and Training

Technologies:

Tensorflow, Keras

Justification:

The model I had used is the Bidirectional LSTM RNN. This propagates the input forward and backward through the RNN layer and then concatenates the output. This helps the RNN to learn long-range dependencies. A recurrent neural network (RNN) processes sequence input by iterating through the elements. RNNs pass the outputs from one timestep to their input—and then to the next.

The layers are staked as:

1. Embedding
2. Bidirectional LSTM (returning sequences)
3. Bidirectional LSTM
4. Dense
5. Dropout
6. Dense with a single output

An embedding layer stores one vector per word. When called, it converts the sequences of word indices to sequences of vectors. These vectors are trainable. After training (on enough data), words with similar meanings often have similar vectors.

* Model Evaluation

Technologies:

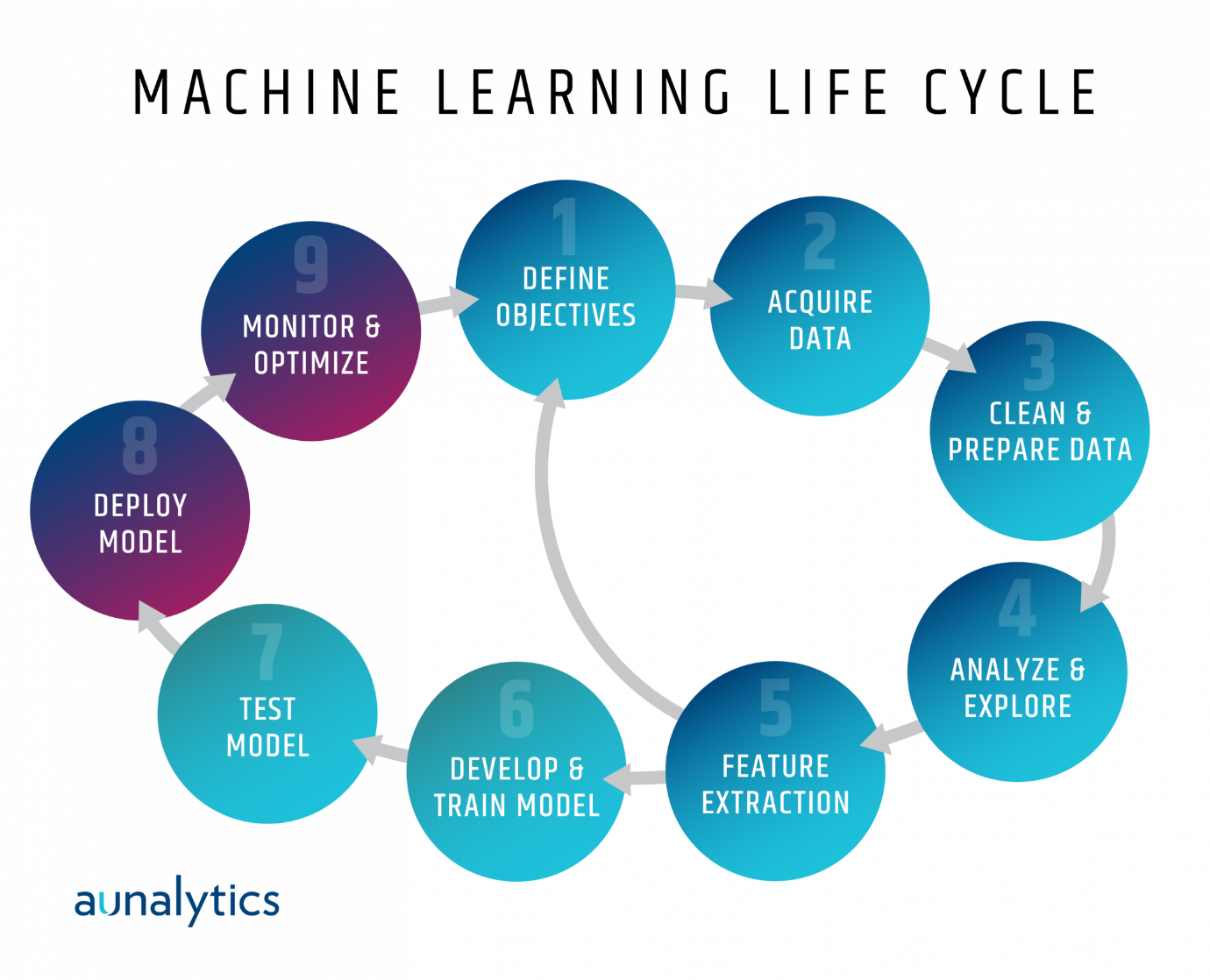
The methods I used to get the performances of the model as the F1-Score, Precision, Recall, and confusion matrix, Sklearn played a major role in getting the matrices values

Justification:

The Model used performed well with pretty good accuracy and the above metrics gave an over performance report of the model. The problem was the classification of the news with two classes true or fake. The confusion matrix evaluated the TP, TN, FP, and FN which provided a raw idea of the classification problem. F1 Score is the weighted average of Precision and Recall.  A recall is the ratio of correctly predicted positive observations. Precision is the ratio of correctly predicted positive observations of the total predicted positive observations.

Accuracy = TP+TN/TP+FP+FN+TN

**Machine Learning life cycle**



* Deployment: This is almost the last step that is to deploy the model to the web and bring it into work, the model can be deployed on the google cloud, AWS, or any other platform.

