NEWS ARTICLES CLASSIFICATION

DETAIL PROJECT REPORT

TABLE OF CONTENT

INTRODUCTION

- The objective of this project is to build a machine learning model to classify news articles into different categories based on their content.
- The problem being solved is the need to categorise large amounts of news articles into meaningful categories in order to make it easier for users to find relevant information.
- The dataset used for this project consists of around 1500 news articles, each labelled with one of several possible categories.

PROBLEM STATEMENT

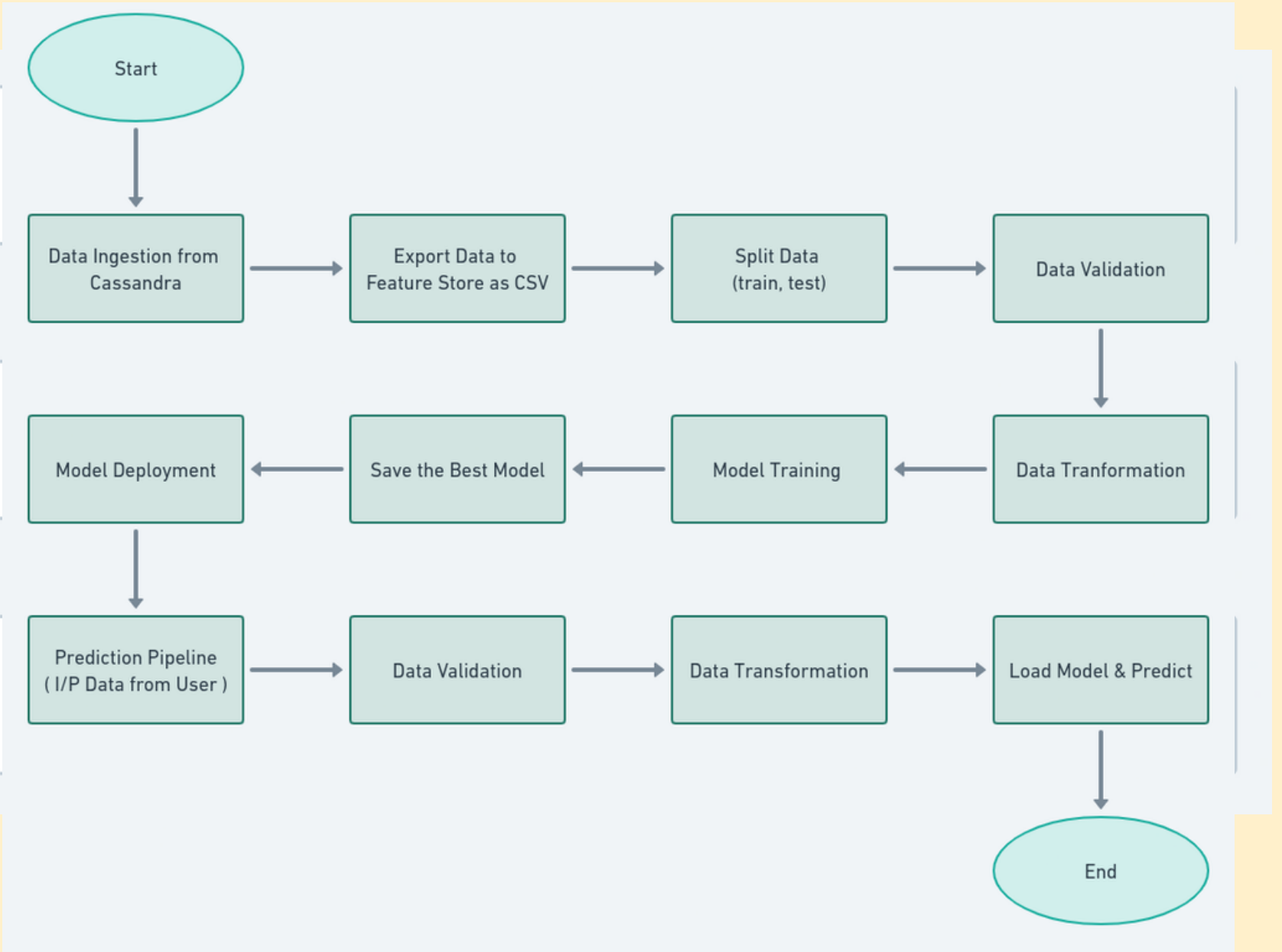
- The problem that this project aims to solve is to develop a machine learning model that can accurately classify news articles into different categories based on their content.
- The need for such a model arises from the need to quickly and accurately categorize news articles in today's fast-paced world, where the volume of news being generated is massive.

METHODOLOGY

Methodology Includes the following steps

- Data Collection
- Data Ingestion
- Data Validation
- Data Transformation
- Model Training
- Model Evaluation
- Model Deployment

ARCHITECTURE



DATA COLLECTION

- The dataset used in this project was obtained from a publicly available repository Kaggle's BBC News Classification
- The dataset consisted of around 1500 news articles, each labelled with one of several categories such as sports, politics, entertainment, etc.
- The dataset was downloaded in a pre-processed format, such as CSV and stored to Cassandra Database for further processing.

DATA INGESTION

- The data to be ingested was stored in a Cassandra database, which is a highly scalable, distributed NoSQL database management system.
- The database contained around 1500 records of various news articles.
- Access to the database was provided through a configured connection string and the Cassandra client library
- The first step in the data ingestion process was to extract the data from the Cassandra database.
- This was done by executing a query against the database, which returned the desired data as a result set.
- The result set was then stored in feature store in csv format.

DATA VALIDATION

- The Data Validation done at several levels as below
    a. Number of Columns Validation
    b. Column Names Validation
    c. Column Data Type Validation
    d. Target Label Validation
- All the Validation Results are stored and then created a yaml report of the validations.

DATA TRANSFORMATION

- The first step in preparing the data for modelling was to clean and normalise it, removing any irrelevant or redundant information.
- This included removing punctuation, converting text to lowercase, removing stop words, and lemmatising words to their root form wit h POS tagging.
- Additionally, any missing or inconsistent data was identified and dealt with, either by removing the affected samples.
- The input features in the dataset were text-based, so they needed to be transformed into numerical representations suitable for modelling.
- This was done using techniques Term Frequency - Inverse Document Frequency(TF-IDF) of natural language processing tasks.
- The output labels were also transformed into numerical representations, for use in training the model.

MODEL TRAINING

- The model was then trained using the pre-processed training data which is splitted into train and test.
- The Logistic Regression, Multinomial Naive Bayes, Random Forest and XG Boost models were used to train the model.
- Multinomial Naive Bayes Classifier Produced best results among them.
- The Naive Bayes classifier was fit to the data using the scikit-learn library in Python.
- The training process involved estimating the model parameters from the training data, such as the probabilities of each word occurring in each class.
- Hyper parameter tuning was performed to optimize the model's performance.

MODEL EVALUATION

- The trained Naive Bayes Model was able to classify 95% of the news article correctly on the Test Data.
- Naive Bayes Model was able to classify most of the article with very few around 2 False Positives and False Negatives.
- Naive Bayes Model was able to provide 98% as the F1 Score.
- Naive Bayes Model gave an accuracy of 95% on cross validation with 5 folds.

MODEL DEPLOYMENT

- The trained model was saved as a Pickle Object and served using Flask, a web framework for Python.
- The deployed model was integrated into an existing application using APIs, allowing users to make predictions by submitting news articles to the model.
- The performance of the deployed model was monitored and the model was updated and retrained as necessary to ensure that it remained accurate and up-to-date.

RESULTS & CONCLUSION

- The deployed model was able to accurately categorise over 95% of news articles.
- The results showed that the model was effective for categorising news articles, and could be further improved by adding additional training data or adjusting the model architecture.
- Overall, the project successfully demonstrated the feasibility of using machine learning to categorise news articles, and has the potential for further development and improvement in the future.

FUTURE WORK

- Improving Model Accuracy by exploring different algorithms and techniques.
- Incorporating additional features into the model, such as sentiment and source of the article.
- Training the model on larger and more diverse datasets.
- Developing a real-time classification system for categorizing news articles.
- Extending the model to classify news articles in multiple languages.