Fake News Prediction Classification Project Documentation

**Introduction:**

* **Project Overview:** This document describes the procedures and outcomes of a Fake News prediction classification project, developed using Jupyter Notebook and Streamlit for the creation of a web application.
* **Objective:** The main goal of this project is to develop machine learning and deep learning models capable of predicting whether a news article is fake or genuine based on various features from the dataset.

**Dataset Description:**

* **Source:** The dataset for this project was sourced from [Hugging Face](https://huggingface.co/datasets/GonzaloA/fake_news).
* **Features:**
  + **Independent Features:** Title (news headline), text (news description).
  + **Target Feature:** 'Label', categorical (0 for fake, 1 for real).
* **Dataset Size:**
  + **Training Dataset:** 32,470 rows
  + **Testing Dataset:** 8,117 rows
* **Preprocessing:** The dataset was pre-processed by handling missing values, cleaning text data, and encoding categorical variables before model training.

**Tools and Libraries Used:**

* NLTK
* NumPy
* Pandas
* Matplotlib
* Seaborn
* Scikit-learn
* TensorFlow
* keras

**Machine Learning Models Used:**

* Logistic Regression
* Naive Bayes
* Decision Tree
* Random Forest

**Deep Learning Model Used:**

* Artificial Neural Network (ANN) using TensorFlow's Keras .

**Methodology:**

* **Exploratory Data Analysis (EDA):** Conducted initial exploration to understand feature distributions.
* **Data Preprocessing:** Involved cleaning data, encoding categorical variables, and utilized Natural Language Processing techniques to convert text data into numerical format. In this process, TF-IDF (Term Frequency-Inverse Document Frequency) was utilized to enhance machine comprehension.
* **Model Training:** Various machine learning and deep learning techniques were used to train models on the training data.
* **Model Evaluation:** Model performance was assessed using metrics such as accuracy, precision, recall, and F1-score.
* **Hyperparameter Tuning:** Models were fine-tuned to optimize their performance.
* **Model Comparison:** The performance of different models was compared to select the best-performing one.

**Results:**

* **Model Performance:** Evaluation metrics (accuracy, precision, recall, F1-score) were recorded for each model on the test dataset.
* **Comparison:** The performance of various models was evaluated, and the top two models for this classification task were identified as Random Forest and ANN, providing the highest accuracy, precision, recall, and F1-score. However, due to constraints such as space and time, Random Forest is preferred over ANN for implementation. Though ANN performs better than random forest but the difference is very less.
* **Key Findings:** Insights were derived from analyzing the models' effectiveness and predictive capabilities.

**Future Work:**

* Potential future directions include incorporating additional features, experimenting with different algorithms, and exploring advanced deep learning architectures.

**References:**

* Google, ChatGPT