**Objective:** To develop a fact-checking plugin deploying Natural Language Processing. (for bot investigation/fake news detection)

**Implementation Plan:**

**Step 1: Data Collection**

* Acquire a diverse dataset of news articles, classifying them as legitimate and fake.
* Identify reputable sources for legitimate news articles.
* Utilize existing datasets for fake news or employ web scraping tools to collect fake news articles.
* Ensure a balance in the representation of various topics and writing styles.

**Step 2: Data Pre-processing**

* Clean and prepare the collected data for effective model training. Perform text cleaning by converting all text to lowercase, removing special characters, punctuation, and HTML tags.
* Tokenizing the text for further analysis.
* Remove common stopwords to focus on meaningful content.
* Apply stemming and lemmatization to normalize words.

**Step 3: Feature Extraction**

* Extract relevant features from the text data for model input. Utilize a pre-trained text embedding model for capturing semantic relationships between words.
* Explore additional features like sentiment analysis, word embeddings, and linguistic features.

**Step 4: Model Building using Deep Learning with Optimizer**

* **Select Appropriate Models:** In this step suitable machine learning (ML) and deep learning (DL) models will be chosen for the task, considering factors such as dataset size and complexity.
* **Optimization:** Configure optimization techniques, including the choice of optimizer (e.g., Adam,GWO, AO, ACO, etc..) and hyper parameter tuning to improve model convergence and performance.

**Step 5: Evaluate Model and Performance Metrics**

* **Cross-Validation:** Use cross-validation techniques to assess the model's performance and generalization capability.
* **Visualization:** Create visualizations (e.g., confusion matrices, ROC curves) to gain insights into model behavior and performance.
* **Performance Metrics:** Evaluate model performance using appropriate metrics, including accuracy, precision, recall, F1-score, and ROC-AUC.

**Step 6: Calculate Article Credibility Rate**

* Quantify the extent of text alteration to determine the "Article Credibility Rate."
* Utilize NLP techniques to compare original and altered versions of articles.
* Apply similarity metrics like cosine similarity or Jaccard similarity.
* Calculate the percentage of alteration based on identified differences.
* **Trustworthiness Analysis:**
* Train an NLP model to evaluate the trustworthiness of articles.
* Utilize pre-trained models or train a custom model on a labeled dataset.
* Extract features such as sentiment, tone, and writing style to assess trustworthiness.
* **Poor Framing and Clickbait Detection:**
* Train models to identify poor framing and clickbait elements.
* Use linguistic features, headline analysis, and context to detect framing and clickbait.
* Apply machine learning models or rule-based systems for classification.
* **Percentage Computation:**
* Compute the percentage of articles falling into trustworthiness, poor framing, and clickbait categories.
* Divide the count of articles in each category by the total number of articles.

**Step 7: Bias Indicator**

* **White-Box Bias Detection Model:**
* Develop a white-box model to detect biases in news articles.
* Train the model using a diverse dataset with labeled bias information.
* Features may include sentiment, word choice, and topic representation.
* **Bias Percentage Calculation:**
* Calculate the bias percentage for each article based on the model's output.
* Define thresholds for categorizing articles into low, medium, and high bias.
* **Original Article Suggestion:**
* If the bias percentage is high, provide users with the option to view the original article for a less biased perspective.

**Step 8: Reassurance Pop-up**

* Develop a pop-up message reassuring users about the platform's commitment to preventing fake news.
* Include information on detection mechanisms, technologies used, and the team's dedication to accuracy.

**Step 9: Listing and Flagging Fake Content**

**Fake Content List:**

* Maintain a database or list to store details of detected fake content.
* Include information such as article title, source, date, and reasons for flagging.

**Flagging Mechanism:**

* Flag fake content prominently on the platform.
* Use visual indicators or labels to alert users about potential misinformation.

**Step 10: Blocking Mechanism**

**Blocking Criteria:**

* Define criteria for identifying highly likely fake content.
* Consider trustworthiness, alteration percentage, and bias indicators.

**User Warning:**

* Implement a blocking mechanism to prevent users from accessing or sharing highly likely fake content.
* Display a warning message informing users about potential misinformation.

**Note:**

The implementation of this objective is application-based, requiring the development and integration of the outlined features into a functional application. Each step in the plan is designed to contribute to the creation of a comprehensive and user-friendly fact-checking plugin within an application environment. From data collection to user warnings and content blocking, the focus is on delivering a seamless and effective experience for users interacting with the application.