**Canvas for 10-Slide Presentation on Fake News Classifier**

**Slide 1: Title & Objectives**

**Title:** Fake News Classifier: From Data to Deployment

**Content to include:**

* Brief project tagline: "Detecting real vs. fake news with TF-IDF & Logistic Regression."
* Objectives:
  + Build a reliable text classifier for fake/real news.
  + Explain each step: EDA, feature engineering, modeling, calibration, CLI.
* (Optional) Your name, date, and organization.

**Slide 2: Data Overview & Motivation**

**Title:** Data Sources & Problem Statement

**Content to include:**

* Describe two CSVs:
  + Fake.csv: 23 502 fake articles
  + True.csv: 21 417 real articles
* Why solving fake news detection matters (misinformation, scale).
* Quick dataset stats:
  + Total articles: 44 898
  + Balanced distribution (52.3% fake, 47.7% real).

**Slide 3: Exploratory Data Analysis (Part 1)**

**Title:** Class Balance & Subjects

**Content to include:**

* Pie chart of fake vs real (52.3% vs 47.7%) — place image
* Bar chart: Number of articles by subject split by label — place image
* Key observations:
  + Major topics: worldnews & politicsNews dominated by real.
  + Fake concentrated in "News" and "left-news".

**Slide 4: Exploratory Data Analysis (Part 2)**

**Title:** Temporal & Length Distributions

**Content to include:**

* Histogram of publication dates (2016–2018) — place image
* KDE plots:
  + Text length in characters — place image
  + Word count in body — place image
* Insights:
  + Fake news peaked in 2016–2017.
  + Real articles slightly longer on average.

**Slide 5: Textual Patterns**

**Title:** Word Clouds & N-grams

**Content to include:**

* WordCloud for fake vs. real — place both images side by side
* Bar chart of top 20 bigrams — place image
* Interpretations:
  + Fake: sensational words ("said", "one", "now").
  + Real: formal language, agency names, dates.

**Slide 6: Feature Space Visualization**

**Title:** PCA of TF-IDF Features

**Content to include:**

* 2D scatterplot of PCA projection with colors for fake/real — place image
* Commentary:
  + Distinct clusters with some overlap.
  + Linear separation feasible (Logistic Regression is reasonable).

**Slide 7: Model Development**

**Title:** Model Pipeline & Hyperparameter Tuning

**Content to include:**

* Pipeline steps:
  1. TF-IDF (max\_features=5000, ngram\_range=(1,2))
  2. LogisticRegression(C=10, class\_weight='balanced')
* Grid search over C and ngrams:
  1. Best found: C=10, ngram\_range=(1,2)
* (Show code snippet or text summary: "GridSearchCV → best\_params").

**Slide 8: Calibration & Threshold Selection**

**Title:** Calibrated Probabilities & Threshold

**Content to include:**

* Explain Platt scaling via CalibratedClassifierCV(method='sigmoid', cv=5).
* Show metrics:
  + ROC AUC: 0.9997
  + F1-scores after calibration
* Optimal threshold from PR curve: 0.573 — show snippet/text:
  + best\_threshold = 0.573

**Slide 9: CLI Tool & Usage**

**Title:** predict.py Script Details

**Content to include:**

* Script features:
  + Loads calibrated pipeline (TF-IDF + model).
  + Accepts one or more full texts or headlines.
  + Outputs label + P(real).
* Usage example (in PowerShell):
* python ./src/predict.py "Full article text..."
* Emphasize dynamic class indexing and ease of integration.

**Slide 10: Results & Next Steps**

**Title:** Performance & Future Work

**Content to include:**

* Final test metrics table:
  + Accuracy: 0.99
  + Precision/Recall (fake & real): 0.99
  + ROC AUC: 0.9997
* Bullet points:
  + Why design choices succeeded: speed, interpretability, reliability.
  + Suggestions:
    - Deploy as REST API
    - Monitor data drift / retrain schedule
    - Explore transformer-based models for long-form text

*End of Canvas*