Phase	Task	Date	Decisions & Rationale	Challenges & Solutions	Implementation
Prediction & Hypothesis Testing Phase	**Load and Prepare Cleaned Datasets**	**17 November 2025**	The team decided to use the cleaned versions of the Reddit, Twitter, and YouTube datasets to ensure the modeling process relies on high-quality data after removing noise. Merging the platforms simplifies comparing metrics across different environments.	**Challenge:** Some rows contained missing or non-numeric values in the *Engagement* column. **Solution:** We used to_numeric() with coerce to remove invalid values, then dropped incomplete rows.	Using pandas to load the three datasets, select only the needed columns (Engagement, toxicity_score, word_count, negative_word_count, Platform), and unify them into a single DataFrame.
	Create Scatterplot (Engagement vs Toxicity)	**17 November 2025**	Using a scatterplot helps visually detect trends between toxicity and engagement, and whether the relationship differs across platforms.	**Challenge:** Platforms differ and require visual distinction. **Solution:** Using the hue parameter in sns.scatterplot to color points by platform.	Generated a seaborn scatterplot (6×4) showing the relationship between Toxicity and Engagement with 0.7 transparency for clarity.
	Encode Platform Using Dummy Variables	**18 November 2025**	Dummy encoding was selected because linear models and decision trees cannot process text. Using drop_first=True prevents multicollinearity.	**Challenge:** Multiple platforms require clear numerical encoding. **Solution:** Created Platform_Twitter and Platform_YouTube.	Applied pd.get_dummies() to add the new encoded columns to df_model while keeping numeric fields unchanged.
	Train–Test Split	**18 November 2025**	Using 20% for the test set is a common standard that maintains enough data for training. random_state = 42 was chosen to ensure reproducible results.	**Challenge:** Engagement values vary widely and may affect split quality. **Solution:** Allowed natural random distribution because sample size was large enough to maintain balance.	Performed the split using train_test_split() for X and y.
	Build Baseline Model (Mean Predictor)	**19 November 2025**	Creating a baseline is essential to evaluate whether advanced models provide meaningful improvement. The mean predictor is the simplest possible numeric prediction.	**Challenge:** Baseline always produces negative R². **Solution:** Compare all models primarily using **MSE** instead of R².	Computed the mean of the training Engagement and repeated it across test samples.
	Train & Evaluate Machine Learning Models	**19–20 November 2025**	Three different models were selected to capture different relationship types: • Linear Regression • Ridge Regression	**Challenge:** Understanding how each feature impacts predictions. **Solution:** Printed model coefficients for linear models. **Challenge:** Decision trees may	A loop was used to train each model, calculate MSE and R², and print coefficients, intercepts, and metrics..

			• Decision Tree (max_depth=5) This diversity allows testing both linear and non-linear patterns.	overfit. **Solution:** Limited depth to 5. .	
	Select Best-Performing Model	**20 November 2025**	Model selection was based on lowest MSE, as it is the most reliable measure of prediction error. This helps determine which model captures the Engagement–Toxicity relationship best.	**Challenge:** Sometimes a model may show better R² but worse MSE. **Solution:** Rely on MSE as the primary metric..	Used min(results, key=lambda k: results[k]["MSE"]) to determine the best model.
	Visualize Toxicity Distribution by Platform	**20 November 2025**	A boxplot was chosen to highlight differences in toxicity levels across Reddit, Twitter, and YouTube. It reveals medians, ranges, and outliers clearly.	**Challenge:** Platforms had different dataset sizes. **Solution:** Combined all toxicity values into a single DataFrame before plotting.	Generated a seaborn boxplot (6×4) comparing Platform vs Toxicity Score.
	Conduct Pairwise Hypothesis Testing (t-test + Bonferroni)	**21 November 2025**	A Welch t-test was used because platforms do not have equal variance. Bonferroni correction was required due to multiple comparisons (3 platform pairs).	**Challenge**: Three platforms = three pairwise tests → increases Type I error. **Solution**: Adjust α by dividing it by number of comparisons.	Created the run_hypothesis_tests function to print: t-statistic • p-value • Statistical decision after Bonferroni correction •