Test Plan

Test Case ID	Component	Objective	Expected Output	Success Criteria	Notes/Reflection
TC1	Data Ingestion - Audio	Verify that audio files are loaded correctly from Google Drive	List of ‘.wav’ file paths (e.g., 'premierleague.wav', 'seriaa.wav')	All files are listed without errors	Reflecting on this, we underestimated how tricky Drive integration could be—permissions were a hurdle.
TC2	Data Ingestion - Reddit	Ensure Reddit posts are scraped successfully	List of 10 post objects with comments	At least 50 posts retrieved, no API errors	The API was very easy to use compared to twitter (X). Could use it for more projects.
TC3	Data Ingestion - Web	Confirm web articles are scraped and text extracted	Raw text string from the article	Text is non-empty and contains football-related content	Web scraping felt chaotic—filtering irrelevant text was tougher than expected.
TC4	Preprocessing - Audio	Test audio transcription and sentence splitting	List of sentences (e.g., "VAR has got to go.", "It delayed the game.")	Transcription matches audio, sentences are split correctly	Whisper’s accuracy was impressive, but noisy audio tripped it up sometimes.
TC5	Preprocessing - Reddit	Validate language detection and translation	Translated text (e.g., "VAR is bad", “Jota should have been yellow carded.”)	Language detected as non-English; translation is accurate	Translation glitches emphasized double-checking with multiple libraries.
TC6	Preprocessing - Web	Check text cleaning and sentence filtering	Cleaned sentences (e.g., "VAR rules!", "Visit our website on....")	Noise removed; only relevant sentences kept	Regex tweaking was tedious but satisfying once it worked.
TC7	Feature Engineering	Ensure text is tokenized and chunked for model input	List of chunks (e.g., chunk1 & chunk2, each ≤512 tokens)	Chunks are correctly sized and tokenized	Chunking long audio transcripts felt like a breakthrough for model compatibility.
TC8	Sentiment Analysis	Verify sentiment prediction accuracy	Sentiment label "POSITIVE" with score >0.8, polarity >0	Matches expected sentiment and threshold	Fine-tuning was tricky. Had to negate “NEGATIVE” values.
TC9	Evaluation	Confirm evaluation metrics are calculated correctly	Metrics (e.g., accuracy >0.75, F1 >0.70)	Metrics align with manual validation	Seeing the confusion matrix come together was rewarding but showed model limits.
TC10	Output - Storage	Ensure results are saved correctly	CSV file (e.g., `var_sentiment_audio_2024.csv`) with correct data and is saved in Google Drive folder.	File is readable, data matches predictions	Saving to CSV felt like a small victory after all the processing.
TC11	Output - Visualization	Validate visualization accuracy	Bar plot showing sentiment is accurate (eg. Negative) and data count matches.	Plot matches data counts	Visuals brought the data to life.

Quality plan

Quality Characteristic	Assessment Criteria	Rating (1-5)	Comments	Mitigation	Reflection
Functional Suitability	The pipeline meets the intended functions and fulfils the requirements	4	Pipeline processes audio, Reddit, and web data, but "NEUTRAL" labels were initially missing.	Revisit input data to ensure mid-range scores exist; adjust thresholds if needed.	We learned how critical input data quality is—missing "NEUTRAL" labels threw off our analysis.
Reliability	The pipeline is stable and performs without failure	5	No crashes or errors in running the pipeline; all cells execute successfully.	N/A—continue monitoring for edge cases in larger datasets.	We were relieved it ran smoothly after all the debugging.
Usability	The pipeline is relatively easy to understand for users	3	Code is well-documented, but visualisations need better labels; "NEUTRAL" issue confused interpretation.	Add clearer plot titles and legends; document the "NEUTRAL" fix in the report.	Struggled to understand other code files from peers.
Performance Efficiency	The pipeline handles the workload in a reasonable time	4	Takes 15 - 40 seconds to process 851 rows on Colab, but the translation and chunking step is slow.	Optimize translation with batch processing; cache translated results for reuse.	The translation bottleneck from deepgram. Used google trans library.
Compatibility	The pipeline works seamlessly on other systems	5	Compatible with Python 3.2 on Colab; dependencies (pandas, seaborn) are standard.	N/A—ensure dependency versions are pinned for reproducibility.	Compatibility on Colab made it easy to share and edit on different systems.
Maintainability	The pipeline is easy to analyze, modify, test, and change	4	Debugging "NEUTRAL" was time-consuming because of thresholds.	Add unit tests for each pipeline step; modularize translation and sentiment logic further.	Debugging without tests was a slog—We'll prioritize testing in future projects.
Portability	The pipeline is accessible on varied environments	4	Runs on Google Colab with minor setup (e.g., Drive mount); no OS issues.	Document setup steps (e.g., Drive mounting) in a README for non-Colab users.	All code is Colab specific, we could have made it more portable to other IDE like Kaggle or locally.

Sprint Plan

Sprint	Scrum	Task	Task	Start Date	End Date	Assigned To	Deliverable	Notes/Reflection

	Master	ID						
Sprint 1	Ryan Kioko	SP01	Define Aims and Objectives	Feb 14, 2025	Feb 14, 2025	Ryan Kioko	Document with project goals	I realized how crucial clear aims are—kept us focused from the start.
		SP02	Define Functional & Non-Functional Requirements	Feb 14, 2025	Feb 17, 2025	Ryan Kioko	Requirements list (e.g., sentiment accuracy >70%)	Non-functional requirements like performance were tricky to define at first.
		SP03	Set Up GitHub Repository	Feb 15, 2025	Feb 16, 2025	Ryan Kioko	GitHub repo with README	Setting up GitHub early made collaboration smoother than expected.
		SP04	Create MS Project File	Feb 17, 2025	Feb 19, 2025	Ryan Kioko	MS Project file with a timeline	Learning MS Project was a steep curve, but it helped us visualize the timeline.
		SP05	Data Scraping (Audio)	Feb 18, 2025	Feb 23, 2025	Elvis Odinkor	Audio files in Google Drive	Audio scraping was slow—permissions issues taught me to double-check access rights.
		SP06	Data Scraping (Reddit & deepgram)	Feb 18, 2025	Feb 25, 2025	Chukwuebuka Tshally-Okeke	Reddit posts and Deepgram API	Reddit and Deepgram API limits were a headache; I had to adjust queries to get enough data.
Sprint 2	Elvis Odinkor	SP07	Data Preprocessing (Cleaning & Splitting)	Feb 28, 2025	Mar 4, 2025	Chukwuebuka Tshally-Okeke	Cleaned, split sentences in CSVs	Stuttering in audio transcripts was a challenge—google translation and reddit API saved the day.
		SP08	Language Standardization (Translation)	Feb 28, 2025	Mar 6, 2025	Elvis Odinkor	Translated text in English	Translation errors made me question the tool—fine-tuning thresholds helped.
		SP09	Perform Sentiment Analysis	Mar 7, 2025	Mar 12, 2025	Elvis Odinkor	Sentiment labels and scores in CSVs	The "NEUTRAL" label issue was a surprise—debugging it taught me a lot about data.
		SP10	Create Test Plan	Mar 7, 2025	Mar 9, 2025	Chukwuebuka Tshally-Okeke	Test plan document/table	Writing the test plan clarified what to focus on—I wish I'd done it earlier.
		SP11	Create Quality Assessment Plan	Mar 10, 2025	Mar 12, 2025	Elvis Odinkor	Quality plan document/table	Assessing quality made me realize gaps in usability—visuals needed more clarity.
		SP12	Result Aggregation (Combine Data)	Mar 13, 2025	Mar 15, 2025	Chukwuebuka Tshally-Okeke	`finalprojectcsv.csv` with all sources	Combining data felt rewarding, but spotting duplicates was tedious.
		SP13	Evaluation (Metrics & Validation)	Mar 16, 2025	Mar 20, 2025	Chukwuebuka Tshally-Okeke	Metrics report (accuracy, F1, etc.)	The confusion matrix showed mislabels—I should've validated more data manually.

		SP14	Output & Visualization (Plots, Word Cloud)	Mar 16, 2025	Mar 20, 2025	Elvis Odinkor	Bar plot, heatmap, word cloud	Visuals brought the data to life, but the "NEUTRAL" issue skewed early plots.
Sprint 3	Chukwuebuka Tshally-Okeke	SP15	Upload Code to GitHub	Mar 21, 2025	Mar 22, 2025	Chukwuebuka Tshally-Okeke	Code and outputs on GitHub	Uploading to GitHub was quick, but I forgot to add a detailed README at first.
		SP16	Create Presentation Slides	Mar 23, 2025	Mar 27, 2025	Ryan Kioko	PowerPoint slides for project overview	Slides took longer than expected—balancing visuals and text was tricky and using AI power slide.
		SP17	Draft Final Report	Mar 23, 2025	Mar 30, 2025	Chukwuebuka Tshally-Okeke	Draft report with all sections	Writing the report helped me reflect on our journey—wish I'd documented earlier.
		SP18	Review and Finalize Deliverables	Mar 31, 2025	Apr 1, 2025	Chukwuebuka Tshally-Okeke	Final report, slides, GitHub repo, MS Project	The final review caught small errors—team collaboration was key here.

Risk Assessment Plan

Risk Category	Description	Likelihood	Impact	Contingency & Counter method
Data Availability Risk	Twitter API changes, access restrictions, or missing data for specific leagues	Medium	High	Use alternative sources (Reddit, sports forums, football websites) if Twitter data is insufficient.
Data Quality Issues	Noisy, biased, or duplicate data leading to inaccurate sentiment analysis	High	High	Implement robust data-cleaning techniques; remove spam, duplicates, and irrelevant tweets.
Sentiment Model Accuracy	NLP models misclassify sarcasm or biased opinions	Medium	Medium	Fine-tune models with football-specific sentiment lexicons; compare results across different models.
Time Constraints	Delays in data scraping, processing, and analysis	Medium	High	Follow Agile Scrum methodology with defined sprints; prioritize key deliverables in case of time crunch.
Technical Issues	API rate limits, data storage failures, or computing power limitations	Medium	Medium	Use cloud storage solutions (Google Drive) and optimize data collection to avoid excessive API calls.
Ethical Considerations	Potential bias in data collection, violating social media policies	Low	High	Ensure compliance with ethical AI practices, anonymize user data, and adhere to platform scraping policies.
Visualization & Interpretation Challenges	Difficulty in making insights easily interpretable for stakeholders	Medium	Medium	Use intuitive graphs, dashboards, and detailed explanations to support data-driven insights.