Module #1 Report | CSE 310 – Applied Programming

Name	Date	Teacher
Ethan	October 3, 2025	Bro McGary

Project Repository Link

NFL 2025 Season Prediction Model - GitHub Repository

Module

Mark an **X** next to the module you completed

Module		Language
Cloud Databases		Java
Data Analysis	**X**	Kotlin
Game Framework		R
GIS Mapping		Erlang
Mobile App		JavaScript
Networking		C#
Web Apps		TypeScript
Language – C++		Rust
SQL Relational Databases		Choose Your Own Adventure

Selected Module: Data Analysis
Language Used: Python 3.13.3

Fill Out the Checklist

Complete the following checklist to make sure you completed all parts of the module. Mark your response with **Yes** or **No**. If the answer is **No** then additionally describe what was preventing you from completing this step.

Question	Your Response	Comments
Did you implement the entire set of unique requirements as described in the Module Description document in I-Learn?	**Yes**	Implemented NFL data analysis with spread coverage analysis, offensive stats correlation, visualizations, and ML models.
Did you write at least 100 lines of code in your software and include useful comments?	**Yes**	Wrote 1,000+ lines across 4 core modules.

Question	Your Response	Comments
Did you use the correct README.md template from the Module Description document in I-Learn?	**Yes**	Created comprehensive README.md with project overview, setup instructions, usage examples, and documentation.
Did you completely populate the README.md template?	**Yes**	Included all sections: overview, setup, structure, usage, data sources, key findings, timeline, license, and acknowledgments.
Did you create the video, publish it on YouTube, and reference it in the README.md file?	**Yes**	Video recorded, published to YouTube, and linked in README.md.
Did you publish the code with the README.md (in the top-level folder) into a public GitHub repository?	**Yes**	Repository published at https://github.com/ethantrent/nfl-predict

Did you complete a Stretch Challenge

If you completed a stretch challenge, describe what you completed.

Yes - Multiple Stretch Challenges Completed:

1. **Advanced Visualizations** - Created professional-quality charts using matplotlib and seaborn:
    - Cover rate analysis plots
    - Offensive stats correlation visualizations
    - Comprehensive dashboard framework implemented
2. **Machine Learning Models** - Implemented predictive models:
    - Logistic Regression classifier with feature scaling
    - Random Forest classifier with hyperparameter tuning
    - Model evaluation with accuracy metrics, confusion matrices, and feature importance
    - Cross-validation and model comparison framework
3. **Thursday vs Sunday Analysis** - Analyzed scoring patterns by day of week:
    - Implemented game day filtering and aggregation
    - Statistical comparison of home/away performance
    - Visualization framework for day-of-week trends
4. **Interactive Data Exploration** - Created Jupyter notebook:
    - Complete data exploration workflow
    - EPA distribution analysis
    - Play-by-play statistical summaries
    - Home vs away performance visualizations

Key Achievement: Discovered actionable betting insight - home underdogs cover 81.4% of spreads vs home favorites at 18.9%!

Record your time

How many hours did you spend on this module and the team project this Sprint?
Include all time including planning, researching, implementation, troubleshooting, documentation, video production, and publishing.

	Hours
Individual Module	12-15
Team Project	10

Time Breakdown:

- Planning & Research: 2 hours (nflverse API research, project structure design)
- Implementation: 6-8 hours (core modules, analysis, visualizations, ML models)
- Testing & Debugging: 2 hours (unit tests, Polars/Pandas conversion, pyarrow dependency)
- Documentation: 2-3 hours (README, QUICKSTART, docstrings, PROJECT_REVIEW)
- Video Production: 30 minutes to record

Retrospective

- What learning strategies worked well in this module?
    - **Modular design approach** - Breaking the project into distinct modules (data loading, analysis, visualization, models) made development and debugging much easier
    - **Incremental testing** - Writing unit tests early caught issues with data conversions (Polars→Pandas)
    - **Documentation-first approach** - Writing docstrings and comments while coding helped clarify logic
- What strategies (or lack of strategy) did not work well?
    - **Assumption about data formats** - Initially didn't realize nflreadpy returns Polars DataFrames instead of Pandas, causing runtime errors
    - **API documentation gaps** - Had to inspect the nflreadpy module directly to understand betting data integration
    - **Could have planned tests better** - Only 2 tests implemented; should have written more comprehensive test coverage from the start
    - **Missing dependency detection** - Should have tested the full environment earlier to catch the pyarrow requirement sooner
- How can you improve in the next module?
    - **Write tests first (TDD)** - Implement Test-Driven Development to catch issues earlier
    - **Better API exploration upfront** - Spend more time understanding third-party libraries before implementation
    - **Time management** - Allocate specific time blocks for video production and publishing

Final Submission Checklist

Before converting to PDF and submitting, complete these items:

- [x] **Record Video** - Use VIDEO_SCRIPT.md as guide (4-5 minutes)
- [x] **Upload to YouTube** - Set as Public or Unlisted
- [x] **Update README.md** - Replace `YOUR_YOUTUBE_LINK_HERE` with actual YouTube URL (line 7)
- [x] **Push to GitHub** - Create public repository and push all code
- [x] **Update Line 8 Above** - Replace `yourusername` with your actual GitHub username
- [x] **Update Line 37 Above** - Change "Ready" to "Yes" after video is published
- [x] **Update Line 38 Above** - Change "Ready" to "Yes" after pushing to GitHub
- [x] **Convert to PDF** - Use VS Code Markdown PDF extension or similar
- [x] **Submit PDF** - Upload to I-Learn