June 9, 2024

Max Sealey (SID: [redacted]

C964: Computer Science Capstone Template

Task 2 parts A, B, C and D

Part A: Letter of Transmittal

June 4, 2024

Roger Goodell

Commissioner

National Football League

345 Park Ave, 5th Floor

New York, NY 10154

To Whoever This May Concern,

I am submitting herewith a project proposal for an application that will use a machine learning model to predict whether or not an NFL team is likely to make the playoffs.

Predicting team success in the past, both by oddsmakers and other entities, has generally been negatively affected by bias and media noise. This machine learning-based solution will utilize the salary cap allocation data of every NFL team to generate a predictive model, the results of which would assist your organization in creating the season schedule, negotiating with streaming services, focusing weekly advertising efforts, and likely several other purposes that I haven't foreseen. This model will ignore all the noise, focusing on and learning from one critical, under-utilized piece of data: salary cap statistics. I have found a correlation between the success of every NFL football team and how they structure player salaries. Specifically on how much they spend on the quarterback (the most important position), and the percentage spent on offense (as a unit) vs. the percentage spent on the defense (as a unit). My goal is to analyze that data with a machine learning-based application to predict yearly football team success with accuracy consistently.

I've estimated the total budget for this project to be approximately $351,361.77, a minute fraction of what the NFL stands to gain from investing in my application. Besides the funding, all I require is the salary cap data already available to the public. This project should be finished in approximately five months, well before the 2025 NFL season. Ten years of experience as a Data Scientist and Software Engineer inform my hypothesis and qualify me to develop this solution. I ask that you strongly consider this proposal and embrace the future of data analytics. If you have any questions, please contact Max Sealey at [redacted]

Sincerely,

Max Sealey, Software & Data Engineer

Part B: Project Proposal Plan

Project Summary

In 2023, the National Football League (NFL) failed to create a schedule that maximized viewership and revenue. An example of this would be the number of primetime slots (six) given to the New York Jets, who fell victim to off-season overhype. Some of those primetime games were shown on streaming services that the NFL has begun working on partnerships with. The NFL needs a reliable model to predict which teams are playoff contenders to maximize the revenue from streaming contracts and give primetime slots to the better teams. They need a model based on hard data, one that won't be swayed by media bias and off-season hype. That is why they should invest in my machine learning-based application to predict whether a team is or isn't a playoff team. This application and its success-predicting capabilities are the best way for the NFL to maximize profits and give fans the viewing experience they deserve. My solution will analyze the three most important factors in the salary cap statistics: quarterback, offense (as a unit), and defense (as a unit). Then, based on that model, the NFL will be able to enter those statistics for any team in the upcoming season and learn whether or not they are likely to make the playoffs. Not only that, but they will be able to see the data visualized in at least three different ways, which will help in the decision-making process. If the NFL greenlights this proposal, they will be getting this game-changing application and an extremely informative user guide.

Data Summary

- The required data is all open source, so the dataset will be found on Kaggle (link will be under References). It will contain a complete record of how much each NFL team spent on each position every year from 2013-2022.
- The data will be put into a structured database, where it can then be queried and modified using a series of Structured Query Language (SQL) commands. Each piece of the application's core functionality will create a table of information from that database using only the subsets of data that they need. For example, let's say that the application wants to make a pie chart and needs only a team name, a year, and the percentage of cap space spent on offense and defense. The resulting table with cleaned, parsed data may look like this:

TEAM	SEASON	OFFENSE_P	DEFENSE_P
Broncos	2015	0.3467891	0.5632678

The use of a structured database like this also makes it easy to add new data, modify data, and improve the machine learning product as a whole.

- Since the goal of the project is to make a prediction based on salary cap data, a database with only that data is enough to meet the needs of this project. The dataset should not have any incomplete data and there are no outliers that need to be removed.
- All of the information used for our machine learning application is publicly available and no personally identifiable information is used. Therefore, there shouldn't be any ethical or legal concerns.

Implementation

- The project will be implemented using the AGILE methodology, which will allow us to continually iterate upon our work until the machine learning model has been adequately trained and the application is complete. The phases include Plan, Design, Develop, Test, Deploy, and Review.
- After setting up and acquiring resources, we'll jump right into the planning stage, which involves hiring and acquiring resources. The design stage involves designing the AI framework, breaking down development into smaller tasks, and delegating responsibilities. Then we begin development on the AI framework, the UI, and we start introducing training data to the model. Testing will be happening simultaneously as we finish new features, but there will be another testing phase after development. Once defects have been addressed, we can begin to let the NFL use it to make predictions for the 2025 season.

Timeline

Milestone or deliverable	**Duration (hours or days)**	**Projected start date**	**Anticipated end date**
Set up workspaces and meeting area	2 days	10/1/24	10/2/24

Financial planning, acquiring resources, hiring	28 days	10/3/24	10/31/24
System design & task delegation	30 days	11/1/24	12/1/24
Creating AI Framework and begin initial data training	20 days	12/2/24	12/21/24
Develop User Interface	12 days	12/22/24	1/4/25
Training dataset, build data ingest pipeline, training pipeline	17 days	1/5/25	1/20/25
Performance testing, legal compliance review	10 days	1/21/25	1/31/25
Fix defects, perform further statistical analysis	15 days	2/1/25	2/15/25
Deployment and user training	12 days	2/16/25	2/28/25

Evaluation Plan

- A consultation with another ML expert will be done at the end of the Design phase to ensure the system design is feasible and of good quality. Regular checks will done on the ML model for overfitting and underfitting during the development and testing phases. Code reviews will be done regularly to ensure quality standards are met and to prevent the introduction of bugs into the code. Unit testing and regression testing will be performed throughout the development process to make sure that the application is functioning properly.
- The solution will be checked for accuracy by dividing the number of correct predictions by the total of incorrect predictions. We would like the initial model to be performing at greater than or equal to 80% accuracy.

Resources and Costs

Hardware and Software Costs

Laptops	~ $ 4,000
Software (PyCharm, Git, Python libraries, Google Drive)	FREE

Labor Costs/Salaries (over a five-month projected timeframe)

Project Manager & Lead Developer (me)	$ 95,069.69
Senior Developer Team (1)	$ 60,821.70
Junior Developers (2)	$ 80,070.26
Intern (1)	$ 35,104.67
SQA Engineers & Testers (2)	$ 42.005.89
IT Specialist (1)	$ 22,289.56

PROJECTED TOTAL:	**$ 351,361.77**

Estimated environment costs (yearly)

Maintenance (developer salary)	$ 130,000.00
Hosting and Deployment (AWS)	$ 2,000.00

Part C: Application

Sealey_Capstone_Project.zip should be included in this submission. Unzip the folder, open in PyCharm IDE, and follow the user guide located in Part D. The User Interface is through the CLI. This project was developed in PyCharm 2023.1.2 (Community Edition) on a 2020 M1 Macbook Pro. Runtime version 17.0.6+10-b829.9 aarch64 (that information may or may not be useful to you running the program).

Part D: Post-implementation Report

Solution Summary

- The NFL had a problem with being unable to consistently predict with accuracy which teams would end up being successful in the upcoming season. By using poor predictions to decide which games to highlight and which games to bury, they missed out on a lot of potential revenue putting bad teams in the primetime slots.
- This application ignores the factors that cause the NFL to make those mistakes (media bias, hype, etc.) and focuses on finding a pattern within hard data. This application also displays an accuracy score and three different ways to visualize the data used in these calculations.

Data Summary

- The dataset was found on Kaggle courtesy of Luke Bukowski (link can be found under references). It contains a complete record of how much each NFL team spent on each position every year from 2013-2022.
- The data was first added to the project as a CSV file. Then I created a function (**get_df_with_cleaned_data**) in **data_helpers.py** that converted the data to a pandas dataframe, created an SQL database, and populated it with the data.
- Another file (**sql_scripts.py**) contains four functions, each returning an SQL script (as a string). Those scripts were passed into **get_df_with_cleaned_data**, which created a table containing only the requested/required/cleaned dataset, then returning it as a pandas dataframe. Each of the three data visualization methods and the ML prediction model each needed a different subset of the data, so a function to produce differently formulated tables was necessary.

Machine Learning

- I employed the Random Forest Classification model, a subsect of the Supervised Learning machine learning model, to make a prediction about which of two classifications a team would fit into: playoff team, or not a playoff team.
- The features that I used to make the model included the percentage of the cap allocated to the QB position, the percentage of the cap allocated to the offense (as a whole), and the percentage of the

cap allocated to the defense (as a whole). I then split the data into training and testing subsets (70/30 split) and fit it to a RandomForestClassifier model imported from scikit-learn. Then I had the model make predictions on random samples of the testing data, the results of which were stored and used to formulate the accuracy score, classification report, and the confusion matrix.

- My original intention was to use a regression model to predict a precise win-loss record, but I decided that that wasn't necessary to solve the client's problem. Being able to tell whether a team has a record of 8-9 or 10-7 doesn't give much information about whether they'll be worth watching. It happens fairly frequently that a team will go 8-9 and make the postseason. Even more frequently does a team go 10-7 and miss the playoffs. By using a classification model and putting teams into two categories, we were able to get right to the heart of the matter.

  I decided to narrow down the features to only the quarterback, the offense, and the defense. An emphasis was placed on the quarterback position because the QB is one of the most important positions in all of sports, and franchise QBs are usually the highest-paid players on their team. The offense and defense position groups were grouped because I found that initial models were falling victim to overfitting, getting lost in the noise of all the different data points.

Validation

- I used the accuracy score metric, classification report, and confusion matrix methods that scikit-learn provided for analysis. The accuracy score was calculated by dividing the number of correct predictions by the number of total predictions.
- The results showed that my model consistently predicts correctly over 50% of the time. Most accuracy scores fell between 0.52 and 0.65. These numbers are lower than I had hoped or anticipated, but there is reason to be optimistic since less than 50% of the teams in the NFL make the post-season. By adjusting the parameters and introducing new data points (not necessarily salary cap related), I can get the accuracy score to be much higher.

Visualizations

The three visualizations listed can be accessed and interacted with through the CLI program.

1. A pie chart showing the % of the salary cap allocated to each position given a team name and season/year
2. A stacked bar graph comparing the % of the salary cap allocated to each position for each of the four divisional rivals in the same season
3. A line graph that shows the trend in % of the salary cap allocated to either offense or defense for each team in a given division

If there are any unforeseen problems accessing them through the CLI, they can also be run through 'ice.py' if necessary. Instructions below.

User Guide

1. Open Sealey_Capstone_Project in the IDE of your choice (preferably PyCharm, as that is the only one this has been tested in).
2. If Pip (Preferred Installer Program) is installed on your machine, open the terminal and in the command line enter:
   ➔ **pip install scikit-learn matplotlib numpy seaborn pandas**
3. If that doesn't work for whatever reason, you'll need to find a way to install those libraries. This may include opening each .py file manually, right-clicking on any import statements with a red underline, and choosing the option to import
4. Inside the 'data' directory is a CSV file containing the full, unedited salary cap dataset
5. Navigate to the 'app' directory or just open 'main.py' and run the file. This program was configured to run with Python 3.9.
6. The CLI should now allow you to run all aspects of the program, including the three data visualizations.
   a. If, due to an unforeseen error, you are not able to view the three data graphs/charts using the CLI, they can be run manually from app/ice.py.

References

Bukowski, L. (2024, February 21). *NFL salary cap spending 2013-2022*. Kaggle. http://www.kaggle.com/datasets/lukebukowski/nfl-salary-cap-spending-2013-2022

No other outside sources were used.