C960 Task 2 Molina

May 23, 2024

Adam Silver, CTO

SNC Inc.

123 Easy Drive

St. Louis, MO

Dear Mr. Silver,

I am pleased to submit this project proposal for your consideration. As a Sports News Company™, it is understandably your goal to provide your customers with the most accurate and up to date sports news and stats. With AI and machine learning in the news, it would be a great time for your company to leverage the power that they hold.

The proposal outlines a machine-learning-based data product aimed at enhancing the analysis of basketball players' statistics for future performance predictions. For instance, if a player like T. Haliburton from the Indiana Pacers averages 18.5 points per game, the algorithm will provide a likelihood of the player scoring above or below his average in future games.

The clients will benefit from having the top-tier information regarding player performance, game outcomes and likelihoods, allowing SNC. Inc. to create better content for its viewers and have increased credibility as a news platform. The sportscasters will have their decision-making process improved by being given additional insight into future games/players.

The funding requirements will be $15,000 for the initial upfront cost to cover the compensation for the chosen ML engineer. The ongoing costs for maintenance and upkeep of the infrastructure that the program runs on will be $2000 a year.

Sincerely,

Abraham Molina

Senior Director

Molina Software Solutions

A. Project Proposal: Predictive Analysis for Basketball Player Performance

**Summary of the Problem**

Sports News Company (SNC Inc.) seeks to enhance its analytical capabilities to provide deeper insights into basketball player performance. The current system lacks the advanced analytical tools needed to meet client demands for detailed performance predictions.

**Benefits of the Data Product**

The proposed data product will:

* **Improve Accuracy**: Utilize advanced machine learning algorithms for precise performance predictions.
* **Support Decision-Making**: Offer detailed insights to aid strategic decisions.
* **Increase Credibility**: Position SNC Inc. as a leader in sports analytics.

**Outline of the Data Product**

The data product will:

* Analyze historical player performance data.
* Predict future points scored utilizing feature analysis using stats like assists, steals, rebounds, etc.
* Feature a simple interface for result interpretation.

**Description of the Data**

The data product will use:

* **Historical Player Data**: Stats from past games, including shooting percentages, rebounds, assists, points, blocks, steals, etc. In addition, the expected average of the players points will be added alongside the data so that the learning algorithm can have even more insight into why the player beat or missed their average
* **External Factors**: Data such as game location (home/away) and minutes played.

**Objectives and Hypotheses**

* **Objective**: Develop a model that accurately predicts if a player will score above or below their average points.
* **Hypothesis**: Advanced features and machine learning will improve prediction accuracy over traditional methods like simple averages and fan sentiment.

**Outline of the Project Methodology**

**Understanding -** First, I'll get a clear idea of what SNC Inc. wants. They need an algorithm to predict if a basketball player will score above or below their average points per game. This helps them in making better predictions and providing more insightful content for their viewers.

**Gathering and Cleaning -** Next, I'll gather all the data we can get on the player's stats. This includes points, assists, rebounds, minutes played, shooting percentages, and details about the games (like whether they were home or away), and I’ll clean this data to make sure there are no errors or missing values.

**Exploring the Data/Features –** I will create some sample visualizations so that I can see patterns and trends. Then, I will choose features to account for like minutes played, shooting percentages, etc.

**Building/Testing –** I will use a logistic regression for this project that will predict the probability of the player scoring above or below their average points. The features mentioned previously will serve as additional feature data that will affect the outcome of the test. For example, if player T. Haliburton is a consistently high scorer, but his rebounds + assists are climbing and they have a negative correlation, that can impact his predicted score.

**Evaluation/Delivery –** Once the model is trained, I will evaluate its performance by performing back-tests. Finally, I will deliver the algorithm to SNC. Inc. along with the associated documentation and reporting as necessary.

**Funding Requirements**

* **Initial Cost**: $15,000 for development.
* **Annual Maintenance**: $2,000.

**Impact on Stakeholders**

* **SNC Inc.**: Better decision-making and competitive advantage. Opportunity to lease access to key insights and analytics.
* **Players and Teams**: Insightful performance trends and improvement areas.
* **Fans and Analysts**: Better viewing experience with more accurate insights from sportscasters.

**Ethical and Legal Considerations**

* **Data Privacy**: Ensure data is anonymized and securely stored.
* **Transparency**: Clearly communicate how predictions are made.
* **Compliance**: Adhere to data protection regulations.

**Relevant Expertise**

Molina Software Solutions (MSS) has a proven track record in data science and machine learning. Our team has delivered successful projects across various industries, and we are confident in our ability to deliver this project effectively.

We look forward to working with SNC Inc. to develop this innovative data product and enhance our analytical capabilities.

B. Executive Summary for IT Professionals

**Decision Support Problem or Opportunity**

Sports News Company (SNC Inc.) aims to enhance its analytical capabilities by developing a machine-learning-based data product that predicts basketball players' future performance. This product will address the need for advanced statistical analysis, enabling more accurate and insightful predictions and enhancing customer engagement.

**Description of the Customers and Their Needs**

The primary customers for this data product are sports analysts, team managers, and dedicated sports fans. These customers require detailed and accurate player performance predictions to make informed decisions, improve game strategies, and provide engaging content for their audience. This product fulfills their needs by offering precise insights derived from historical data.

**Existing Gaps in Data Products**

Currently, SNC Inc. lacks the advanced analytical tools required to provide deep insights into player performance. The existing system focuses on basic statistics and does not leverage machine learning for predictive analysis. This gap limits the ability to offer detailed predictions and strategic recommendations, which the proposed data product aims to address.

**Data Available and Required**

* **Available Data**: Historical player performance statistics, including points, assists, rebounds, minutes played, shooting percentages, and game details (home/away).
* **Required Data**: Additional external factors such as estimated averages and player injuries to enhance predictive accuracy. This can be gathered from public facing sources.

**Methodology for Data Product Design and Development**

1. **Data Collection and Preprocessing**: Gather and clean historical and external data to ensure accuracy and completeness. The actual main data will be pulled from a [reputable stats website](https://basketball-reference.com), which will definitely require cleanup as some of the columns are extraneous (like game number, age in days, etc.)
2. **Feature Engineering**: Create new features that capture important patterns and relationships in the data. Some of the chosen ones include the points to rebounds + assists ratio and 2-point field goal to 3-point field goal ratio. These data fields provide crucial insight as the model will attempt to classify the data using all of the fields. Coupling home/away status, the team played against, and these additional ratios (typically not present on traditional sports statistics sites) our model will provide a cutting-edge estimate of a player’s performance.
3. **Model Development**: Train and validate the chosen machine learning model to identify the best-performing features. As mentioned previously, the model development will consist of building the model in python utilizing Scikit-learn, and feature engineering by choosing specific fields that will provide the model additional insight into classifying data.
4. **Implementation**: Develop a user-friendly application to present the predictions and insights. As of now, the plan is to have the interface be primarily command line, with popout windows for the various graphs generated by matplotlib.
5. **Data Validation and Testing**: Continuously evaluate model performance and make necessary adjustments. The model may need to be tweaked to increase accuracy or prevent errors.

**Deliverables**

1. **Data Collection and Preprocessing Scripts**: Tools to gather and clean data will be developed. These tools will encompass manual collection methods, such as gathering and collating data from various sources that cannot be easily or quickly scripted. The data will be extracted from the referenced website and saved into a .csv file, which the program will subsequently read. Additionally, methods will be developed within the program to preprocess this pre-formatted data, ensuring it is accurately loaded into the tables. This will include steps to clean the data, handle missing values, and format it appropriately for analysis, ensuring a seamless integration into the data pipeline.
2. **Feature Engineering Documentation**: Detailed explanation of the features used. The features will provide the model with additional insight so that it can better classify data. For example, say a basketball player tends to go over his average much more often when he attempts more 3 pointers vs 2-point field goals. The model will pick up on this metric due to the engineer specifically adding it in, along with many other features that can factor into the model’s accuracy. All features will have their purpose and design fully documented so that stakeholders can better grasp why they are being used and in what manner.
3. **Trained Machine Learning Model**: The predictive model with performance metrics. Classification reports that include accuracy scores and a confusion matrix will help the data engineer better understand and tune the model for accuracy.
4. **User Interface Design**: An interface for displaying predictions and insights. The developer will provide a simple command line interface.
5. **Final Report**: Comprehensive documentation of the project, including methodology, results, and future recommendations.

**Implementation Plan and Anticipated Outcomes**

* **Phase 1: Data Collection and Preprocessing** (3 days/24 working hours)
  + **Outcome**: Clean and structured dataset ready for analysis.
* **Phase 2: Feature Engineering and Model Development** (10 days/80 working hours)
  + **Outcome**: A trained machine learning model with high predictive accuracy.
* **Phase 3: Implementation and Integration** (2 days/16 working hours)
  + **Outcome**: A functional interface for end-users.
* **Phase 4: Testing and Validation** (1 day/8 working hours)
  + **Outcome**: Verified and validated data product ready for deployment.

**Validation and Verification Methods**

* **Cross-Validation**: Ensure model accuracy and robustness through cross-validation techniques.
* **Back-Testing**: Evaluate model performance using historical data to simulate real-world scenarios.
* **User Testing**: Gather feedback from end-users to ensure the product meets their needs and expectations.

**Programming Environments, Costs, and Human Resources**

* **Programming Environments**: Python (with libraries such as pandas, scikit-learn, and matplotlib) and VS Code.
* **Costs**: Initial development cost of $15,000, annual maintenance cost of $2,000.
* **Human Resources**:
  + Data Scientist (1): Responsible for data collection, preprocessing, and model development.
  + Software Developer (1): Responsible for interface development and integration with model.
  + Project Manager (1): Responsible for overseeing the project timeline and deliverables.

**Projected Timeline and Milestones**

* **Milestone 1: Data Collection and Preprocessing** (Start: 05/27/2024, End: 05/28/2024, Duration: 3 business days)
  + **Dependencies**: None
  + **Resources**: Data Scientist
* **Milestone 2: Feature Engineering and Model Development** (Start: 05/29/2024, End: 06/11/2024, Duration: 10 business days)
  + **Dependencies**: Completion of Milestone 1, dataset normalized
  + **Resources**: Data Scientist
* **Milestone 3: Implementation and Integration** (Start: 06/12/2024, End: 06/14/2024, Duration: 2 business days)
  + **Dependencies**: Completion of Milestone 2, model completed
  + **Resources**: Software Developer, Project Manager
* **Milestone 4: Testing and Validation** (Start: 06/15/2024, End: 06/15/2024, Duration: 1 day)
  + **Dependencies**: Completion of Milestone 3, interface created
  + **Resources**: Data Scientist, Software Developer, Project Manager

By adhering to this structured plan, Molina Software Solutions will deliver a high-quality data product that enhances SNC Inc.'s analytical capabilities and meets the needs of its customers effectively.  
  
  
C. Design and Develop

1. **Descriptive and Nondescriptive Methods**:
   * **Descriptive**: The project includes descriptive statistics (like mean calculations) and visualizations (histograms, scatter plots).
   * **Nondescriptive**: The project uses a logistic regression model to predict whether a player's performance will be above or below their estimated average based on past performance metrics.
2. **Datasets**:
   * The project handles datasets from CSV files concerning regular season and playoff performances. The actual dataset used, while named regular season, is actually a combination of both reg season and playoff game stats.
3. **Decision Support Functionality**:
   * The ML tool makes decisions based on past performance data. It categorizes the data and attempts to classify a future game based on the implied numbers.
4. **Data Processing (Featurizing, Parsing, Cleaning)**:
   * The code parses dates, handles missing values, and transforms categorical data into numerical formats. Features like minutes played and Win/Loss are extracted and converted.
5. **Methods and Algorithms for Data Exploration and Preparation**:
   * The preprocessing function systematically cleans and prepares the dataset, which includes handling data types, extracting features, and creating dummy variables.
6. **Data Visualization**:
   * The application implements histograms and scatter plots, which are fundamental for exploring distributions and relationships within the data. Users can view visualizations separately prior to preforming model training.
7. **Interactive Queries**:
   * The application includes a command-line interface that allows users to interact with the model by entering commands and making selections.
8. **Machine-Learning Implementation**:
   * A logistic regression classification model is employed to predict outcomes based on the processed features. It does not utilize a continuous model that predicts specific scores. It instead categorizes all of the data and features are generated that allow the algorithm to make a prediction of whether or not the specified player beats his estimated average for the game. Specifically, it uses features specified in the code that allow the algorithm to have a better insight into the players performance. For example, the algorithm will not just have access to the points and assists, but combinations like the ratio of points to rebounds and assists.
9. **Accuracy Evaluation**:
   * The program provides functionality to evaluate the model using accuracy scores, classification reports, and confusion matrices.
10. **Security Features**:
    * The only real security feature is input validation is performed at the preprocessing stage to ensure data cohesiveness and program functionality. Otherwise, no further security precautions are needed at this moment as the code is both 1) stored and executed locally and 2) no PII is present in the dataset.
11. **Monitoring and Maintenance Tools**:
    * The script features debugging tools that offer detailed insights into the most demanding sections of the program and generate logs. Additionally, there is a settings menu that allows for further customization of the program.
12. **User-Friendly, Functional Dashboard**:
    * The command-line interface provides a direct way for users to interact with the system. In addition, several interactive menus are provided for the user to interact with the program in depth.

D. Documentation for the Developed Product

**Business Vision –**

The goal of this project is to develop a machine-learning-based data product that analyzes basketball player statistics to predict future performance. The product will provide Sports News Company (SNC Inc.) with advanced analytical capabilities, allowing them to offer content with increased insights and accuracy.

**Business Requirements -**

1. **Data Ingestion**: The system should be capable of reading and preprocessing basketball player statistics from pre-cleaned CSV files.
2. **Data Analysis**: The system must provide descriptive analytics using visualizations such as histograms and scatter plots.
3. **Predictive Modeling**: The system should implement machine learning algorithms to predict if a player will score above or below a specified line.
4. **User Interface**: The product should offer an intuitive dashboard for displaying predictions and insights.
5. **Accuracy and Validation**: The system should provide metrics to assess the accuracy of predictions, including classification reports and confusion matrices.

**Raw and Cleaned Datasets -**

**Raw Datasets:**

* Original CSV files containing raw player statistics. They are pulled from several sources (listed in references) and are collated together and cleaned. For example, the sources used are not in the same order, and several rows have to be removed (such as games where the player did not play).

**Cleaned Datasets:**

* Processed CSV files with cleaned and transformed data used for analysis and model training. This is crucial as the model expects a specific product.

**Code and Executable Files:**

* I used a chrome extension called ‘Table Capture’ to scrape the data from one of the websites.
* See below for the preprocessing code

**Code for Data Analysis and Product Construction -**

**Preprocessing Code:**

def preprocess(file\_path):

    #read csv

    df = pandas.read\_csv(file\_path, parse\_dates=['Date'], dtype={'MP': str})

    #split the result and point diff from the W/L column

    df[['Result', 'Point\_Diff']] = df['W/L'].str.extract(r'([WL])\s\*\(([-+]?\d+)\)?')

    #drop NaNs from the MP because the model freaks out if there's any NaNs

    df = df.dropna(subset=['MP'])

    #drop extraneous/old columns

    df = df.drop(columns=['Rk', 'G', 'Age', 'Tm', 'W/L'])

    # convert 'Point\_Diff' to numeric (if applicable)

    df['Point\_Diff'] = pandas.to\_numeric(df['Point\_Diff'], errors='coerce')

    #store encoded data in a dictionary for user reference

    label\_encoders = {}

    label\_encoders['Opp'] = LabelEncoder()

    label\_encoders['Result'] = LabelEncoder()

    label\_encoders['LOC'] = LabelEncoder()

    #encode data that must be in boolean format (W/L must be 0 or 1, etc.)

    df['LOC\_encoded'] = label\_encoders['LOC'].fit\_transform(df['LOC'])

    df['Opp\_encoded'] = label\_encoders['Opp'].fit\_transform(df['Opp'])

    df['Result\_enc'] = label\_encoders['Result'].fit\_transform(df['Result'])

    #convert minutes played to a number and strip colons

    df['MP'] = df['MP'].apply(convert\_mp\_to\_minutes)

    df['MP'] = df['MP'].astype(float)

    # Convert to seconds since epoch. This one is less of a requirement for the model and more of one

    # for the model just to work as it doesn't do well with datetime objects.

    if 'Date' in df.columns:

        df['Date'] = pandas.to\_datetime(df['Date']).astype('int64') / 10\*\*9

    #remove any NaNs if still exist

    df.fillna(0, inplace=True)

    #convert to numeric

    df['MP'] = pandas.to\_numeric(df['MP'], errors='coerce')

    df['FG%'] = pandas.to\_numeric(df['FG%'], errors='coerce')

    df['3P%'] = pandas.to\_numeric(df['3P%'], errors='coerce')

    #feature generation

    df['above\_line'] = df['PTS'] > df['Line']

    df['above\_line'] = df['above\_line'].astype('int')

    df['rebounds\_assists\_ratio'] = df['TRB'] / df['AST']

    df['pts\_reb+ast\_ratio'] = df['PTS'] / (df['TRB'] + df['AST'])

    df['3pa\_fga\_ratio'] = df ['3PA'] / (df['FGA'] - -df['3PA'])

    df.replace([float('inf'), -float('inf')], 0, inplace=True)

    # Calculate season average stats

    season\_avg\_pts = df['PTS'].mean()

    season\_avg\_ast = df['AST'].mean()

    season\_avg\_trb = df['TRB'].mean()

    # Calculate performance deviation from average

    df['deviation\_pts'] = df['PTS'] - season\_avg\_pts

    df['deviation\_ast'] = df['AST'] - season\_avg\_ast

    df['deviation\_trb'] = df['TRB'] - season\_avg\_trb

    # Calculate rolling statistics with min\_periods=1 to avoid NaN for fewer games

    df['rolling\_std\_pts'] = df['PTS'].rolling(window=5, min\_periods=1).std()

    df['rolling\_std\_ast'] = df['AST'].rolling(window=5, min\_periods=1).std()

    df['rolling\_std\_trb'] = df['TRB'].rolling(window=5, min\_periods=1).std()

    df['rolling\_mean\_pts'] = df['PTS'].rolling(window=5, min\_periods=1).mean()

    df['rolling\_mean\_ast'] = df['AST'].rolling(window=5, min\_periods=1).mean()

    df['rolling\_mean\_trb'] = df['TRB'].rolling(window=5, min\_periods=1).mean()

    df['z\_score\_pts'] = (df['PTS'] - df['rolling\_mean\_pts']) / df['rolling\_std\_pts'].replace(0, 1)

    df['z\_score\_ast'] = (df['AST'] - df['rolling\_mean\_ast']) / df['rolling\_std\_ast'].replace(0, 1)

    df['z\_score\_trb'] = (df['TRB'] - df['rolling\_mean\_trb']) / df['rolling\_std\_trb'].replace(0, 1)

    df['rolling\_mp'] = df['MP'].rolling(window=5, min\_periods=1).mean()

    # Calculate trends

    df['trend\_mp'] = calculate\_trend(df['MP'])

    df['trend\_pts'] = calculate\_trend(df['PTS'])

    df['trend\_ast'] = calculate\_trend(df['AST'])

    df['trend\_trb'] = calculate\_trend(df['TRB'])

    df.fillna(0, inplace=True)

    return df, label\_encoders

**Model Training and Prediction Code:**

def train\_model():

    try:

        # model creation

        X = set1[['Point\_Diff', 'Result\_enc', 'LOC\_encoded', 'Opp\_encoded', 'MP', 'FG%', '3P%', 'FT%', 'TRB', 'AST', 'STL', 'BLK', 'TOV', 'rebounds\_assists\_ratio', 'pts\_reb+ast\_ratio', '3pa\_fga\_ratio', 'PTS', 'rolling\_std\_pts', 'rolling\_std\_ast', 'rolling\_std\_trb', 'z\_score\_pts', 'z\_score\_ast', 'z\_score\_trb', 'rolling\_mp', 'trend\_mp', 'trend\_pts', 'trend\_ast', 'trend\_trb']]

        #X = set1.drop(columns=['Line'])

        y = set1['above\_line'].astype(int)

        os.system('cls')

        #create train and test objects from the train\_test\_split method according to settings paramters

        #default test\_size is 0.2, meaning %20 of the data is reserved for test cases and the rest for training data

        #random state is the seed for the random number gen

        X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=settings['test\_size'], random\_state=settings['random\_state'])

        #create and fit the model according to the generated training data

        model = LogisticRegression(max\_iter=1000)

        model.fit(X\_train, y\_train)

        #predict against the test cases to generate evaluation data.

        y\_pred = model.predict(X\_test)

        return model, X\_train, X\_test, y\_train, y\_test, y\_pred, X, y

    except Exception as e:

        print(Fore.RED + f"Error during model training: {e}")

        error\_logger.error(ansi\_cleaner(f"An error occurred: {str(e)}"))

    print(Fore.LIGHTGREEN\_EX + 'Data trained!')

model, X\_train, X\_test, y\_train, y\_test, y\_pred, X, y  = train\_model()

**Assessment of Hypotheses**

**Hypothesis:** The inclusion of advanced statistical features and machine learning techniques will significantly improve the accuracy of performance predictions compared to traditional methods, such as simple averaging. Traditional methods involve calculating a player's average points per game to predict future performance.

**Assessment:** To assess this hypothesis, the logistic regression model's predictions were compared against simple averaging. The model's accuracy, precision, recall, and F1-score were evaluated using the test set. The comparison showed that the logistic regression model provided more accurate predictions than the simple average method. For instance, Tyrese Haliburton's average playoff points in the 2024 NBA Playoffs through May 21 was 19.3, while the line was set at 18.5. The algorithm predicted below the line, and in the real-world test case on May 23 he scored 10 points, confirming the prediction.

**Visualizations and Storytelling Elements**

**Key Storytelling elements -**

* **Histograms:** Distribution of points scored. Users can view the frequency that a player scores points per band in bands of 2 points.
* **Scatter Plots:** Relationship between minutes played and points scored, points vs. rebounds and assists ratio, and points vs. assists recorded.
* **Decision support:** The algorithm attempts to classify the input data into one of two buckets: above or below.

**Data Summary -**

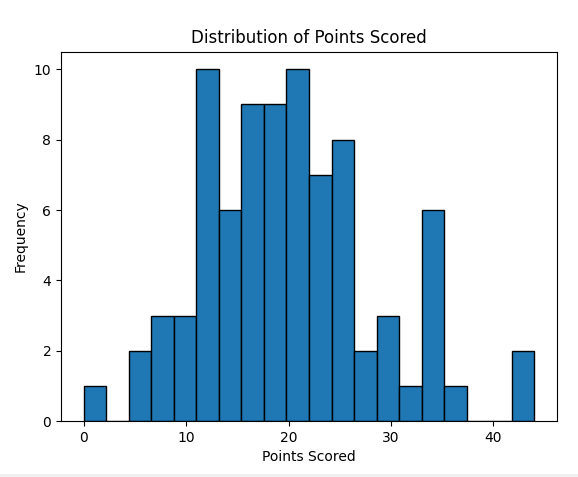
* **Tabular Data:** The program utilizes pandas to store the data efficiently in a tabular format. It allows operations on entire columns at once, which is extremely effective for generating features (e.g. rolling average of assists in last 6 games, standard deviations of game stats, trends, etc..)

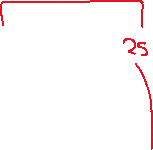
A screenshot of a computer

Description automatically generated

Tabular data shown above

* **Phenomenon Detection:** In the histogram visualization provided by the program the viewer can infer that the player Tyrese Haliburton scores either 12-13 or 20-21 points the majority of the time. While this doesn’t sound that interesting, there is a huge dropoff after 24 points scored in a game, then a huge spike in the the 34-35 bracket. Intuitive users can potentially infer that if this player scores atleast 25 in a game, he is MUCH more likely to continue on to score 34+ points.

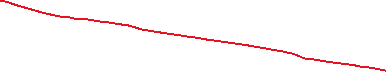




Similarly, the scatter plots provided show that the more rebounds and assists that this player gathers, the (very slightly) less points they tend to score, which isn’t always true for every NBA player. (see: Luka Doncic)

A graph with blue dots

Description automatically generated



**Assessment of Product’s Accuracy**

The model's accuracy was evaluated using the test dataset. Key metrics include:

* **Accuracy:** The proportion of correct predictions.
* **Precision, Recall, and F1-Score:** Detailed in the classification report.
* **Confusion Matrix:** Provides insights into the types of errors made by the model.

**Results from Data Product Testing, Revisions, and Optimization -**

**Testing:**

* The model uses it’s own data for backtesting purposes (in below screenshot, test\_size affects how much data is used for test cases)
* Cross-validation is used when attempting to guess the category of the new data

**Revisions:**

* Improved feature engineering based on initial results. Better features could be worked (like performance vs. specific opponent).
* Adjusted hyperparameters for better model performance (like allowing the changing of minute details of the model training such as the max\_iter variable)

**Optimization:**

* The model can be refined by tweaking the settings in the program. For example, the default settings(test\_size of 0.2 and random\_state of 42) average between 0.70 ~ 0.89 accuracy score depending on the player chosen (SEE SCREENSHOT BELOW). My personal preference settings (test\_size of 0.3 and random\_state of 3) average around 0.90 – 0.92 accuracy scores. This shows that tweaking and further optimization of the model can achieve higher accuracies.

**Screenshots -**

**Visualizations:**

Screenshots of the visualizations can be viewed in earlier sections

**Model Evaluation Metrics (default settings):**

**A screenshot of a computer

Description automatically generated**

**Prediction Results (sample prediction data, successfully predicted IND@BOS on 5/23):**

**A black screen with white text

Description automatically generated**

**Source Code and Executable Files**

See attached zip file for source code/files

**Quick-Start Guide**

**Installation and Setup:**

1. **Prerequisites:** Ensure Python and necessary libraries are installed
   * The following external libraries are required:
     1. scikit-learn
     2. easygui
     3. pandas
     4. matplotlib
   * They can be installed using the command: pip install <name>
2. **Download Data:** Obtain the raw datasets
   * [Basketballreference.com - Tyrese Haliburton 23/24 Full Season/Playoffs](https://www.basketball-reference.com/players/h/halibty01/gamelog/2024)
   * [Bettingpros.com - Tyrese Haliburton Historical Lines](https://www.bettingpros.com/nba/props/tyrese-haliburton/points/)
     1. This one must be scraped utilizing the table capture extension (referenced in sources).
     2. **To Scrape:** Extensions > Tables > Choose the largest table, mine was 85 x 15 table, press blue sheet with green plus icon to copy data and paste in topleftmost cell in excel. Save for now.
3. **Clean Data:** The following cleaning actions must be preformed **IN ORDER**
   * Delete **ALL** rows with DNP/DND/Missing data fields. This step is the most crucial as NaNs must not exist in the dataset. You must select the entire numbered row on the left hand side on the window, right click, and press delete so that the rows delete and move up properly after deletion.
   * Ctrl + A to select all > Right Click > Sort > Custom Sort > Ensure ‘My data has headers’ is checked (top right) > Sort by Date, Sort on Cell Values, Order Newest to Oldest
   * From the scraped table data, select the ‘Prop Line’ column and paste it on the right side of the combined season csv file. The last three columns should look like:
   * GmSc , +/- , Line
   * Delete ALL rows where ‘NL’ is listed. Again, right click and delete whole row.
   * Change column ‘F’ to LOC
   * Change column ‘H’ to W/L
   * Change column ‘AE’ from Prop Line to just Line
   * Move csv files into a good folder for safekeeping. For simplicity, there is the ‘data’ folder in the main project directory which works just fine. The program will ask you to select a file. Once completed, you are now ready to use the program.

**Usage:**

1. **Run Program**: execute main.py (or .exe if compiled from source)
2. **Follow Prompts:** There are a few features to note
   * Option 1 – Visualizations can show you the scatter plots as well as the entire processed pandas data table
   * Option 2 – Accuracy/Classification Report shows the accuracy report for the trained model
   * Option 3 – Re-train the model (self-explanatory, but must be done after tweaking any settings)
   * Option 4 – Enter custom query allows users to input custom data to predict and classify a future match. An example query is provided which works for the t\_haliburton\_23-24\_regszn.csv dataset.
   * Option 5 – Settings shows settings that can be tweaked
   * Option 6 – Label Mappings shows the encoded labels that the preprocessing algorithm generated. It assigns teams their own ID (alphabetically), W/L as 1/0, and Home/Away as 0/1 (in the actual dataset, 1 = away because it is represented with a ‘@’ symbol and home has no symbol, hence the 0 home/1 away)

References:

<https://chromewebstore.google.com/detail/table-capture-tabular-dat/bmmhahnhjkhgephgblidkpoidfkhchnk?hl=en>

<https://www.basketball-reference.com/players/h/halibty01/gamelog/2024>

<https://www.bettingpros.com/nba/props/tyrese-haliburton/points/>