Oren Shapira
Michael DeMers
CS699

Predicting the Performance of Quarterbacks in the NFL

Contents

Data Mining Goal: 2

Data Set Descriptions: 3

Data Preprocessing: 6

Data Mining Process and Analysis 11

Tools and Methodologies Used: 11

Attribution Selection: 11

Classifier Algorithms: 13

Selecting our Model of Choice: 13

Additional Model Performance Analysis: 15

Confusion Matrix 15

Additional Comparison Metrics: F-Measure 16

Additional Comparison Metrics: ROC/AUC 17

Conclusion 18

Data Mining Goal:

Our goal for this project was to use a combination of different variables to predict the performance of Quarterbacks (QBs) in NFL games. The metric we focused on for QB performance was classifying how much a QB's passing yards in a given game would deviate from their season average of passing yards per game. The predictor variables we were most interested in related to weather (temperature, wind, and humidity), and the strength of the team defense a QB was facing. In online Fantasy Football leagues, these variables are often taken in consideration when deciding whether to start a specific Quarterback

on your Fantasy Team roster for a weekly matchup. Discovering the extent of weather and a defense's impact on QB performance would help Fantasy Football participants gauge how much weight these variables should play into their roster decisions. For the project's purposes of including many variables for analysis and testing, we also evaluated the significance of other attributes such as game date, game location (home or away), passing completion percentage, passes attempted, and game score.

Data Set Descriptions:

We used four different datasets for this project, which were merged together into one master data set for our data mining analysis. Each of these datasets are described in detail below:

1. **Weather_2013_1231.csv: (Weather dataset)**
    a. Summary: This dataset contains weather information for every NFL game between 1960-2013. It has 11 attributes, and 11,192 tuples.
    b. Source: http://www.nflsavant.com/about.php
    c. Additional detail has been provided on attribute values that are not intuitive based on column name, or were particularly relevant to our project.
        i. Id
        ii. Home_team:
        iii. Home_score
        iv. Away_team
        v. Away_score
        vi. Temperature
        vii. Wind_chill: Likely won't use this, since it's only available for some games
        viii. Humidity
        ix. Wind_mph
        x. Weather: A concatenation of the other weather columns
        xi. Date
2. **Game_Logs_Quarterback.csv (QB Stats for every game)**
    a. Summary: This dataset contains passing statistics for every Quarterback, for NFL game between 2000 – 2016. This includes total passing yards, pass attempts, Touchdowns, Interceptions, etc. This will be joined with the weather dataset above on overlapping years. This dataset has 29 attributes and 40,247 rows.
    b. Source: https://www.kaggle.com/kendallgillies/nflstatistics
    c. Additional detail has been provided on attribute values that are not intuitive based on column name, or were particularly relevant to our project.
        i. Player Id: Unique Identifier
        ii. Name: Name of Quarterback
        iii. Position: Player position
        iv. Year: Year of game. Will be joined with weather dataset for overlapping years
        v. Season: Distinguishes preseason vs. regular vs. postseason
        vi. Week: Season week of game
        vii. Game Date: Will be joined with weather dataset
        viii. Home or Away: This and Opponent will determine where the game was played

ix. Opponent:
x. Outcome: Win or Loss
xi. Score
xii. Games Played
xiii. Games Started
xiv. Passes Completed
xv. Passes Attempted
xvi. Completion Percentage
xvii. Passing Yards: This is the attribute we would like to predict
xviii. Passing Yards Per Attempt: We may look at this if time allots
xix. TD Passes
xx. Ints
xxi. Sacks
xxii. Sacked Yards Lost
xxiii. Passer Rating: Numeric rating of QB's performance in the game
xxiv. Rushing Attempts
xxv. Rushing Yards
xxvi. Yards Per Carry
xxvii. Rushing TDs
xxviii. Fumbles
xxix. Fumbles Lost

3. **Career_Stats_Passing.csv (QB stats for season averages)**
    a. Summary: This dataset contains every NFL player's overall passing stats for the entire season, and spans between 1924-2016. It will be used primarily to normalize across Quarterbacks that have different averages of passing yards per game. There are 23 attributes and 8,525 rows.
    b. Source: https://www.kaggle.com/kendallgillies/nflstatistics
    c. Additional detail has been provided on attribute values that are not intuitive based on column name, or were particularly relevant to our project.
        i. Player Id
        ii. Name
        iii. Position: We would only filter on the position of Quarterbacks
        iv. Year: NFL season year. Will be merged with other datasets on overlapping years
        v. Team: The team the Quarterback plays for
        vi. Games Played
        vii. Passes Attempted
        viii. Passes Completed
        ix. Completion Percentage
        x. Pass Attempts Per Game
        xi. Passing Yards
        xii. Passing Yards Per Attempt
        xiii. Passing Yards Per Game: This will be the key metric used to normalize for the impact of weather on passing ability
        xiv. TD Passes
        xv. Percentage of TDs per Attempts

xvi. Ints
xvii. Int Rate
xviii. Longest Pass
xix. Passes Longer than 20 Yards
xx. Passes Longer than 40 Yards
xxi. Sacks
xxii. Sacked Yards Lost
xxiii. Passer Rating

4. **Sportsref_download_ [2000-2013].xls (Team Defense Stats)**
    a. Summary: This series of datasets contains team defensive stats for each NFL season between 2000 - 2013. These source data sets were combined into a "Defense_combined_2.xlsx" dataset, with some additional columns we derived (see in italics below). We initially explored using min_max normalization for a team's total yards in a given season, which would subsequently be used to normalize how many yards a QB throws per game. However, we were not able to come up with good min or max values that wouldn't potentially overinflate the impact of defense. We instead binned defensive strength based on Z-score of yards allowed, and used this as a predictor variable in our model. There are 28 attributes and 32 tuples.
    b. Source: https://www.pro-football-reference.com/years/2013/opp.htm
    c. Additional detail has been provided on attribute values that are not intuitive based on column name, or were particularly relevant to our project.
        i. Rk: Defensive rank
        ii. Tm: Team name
        iii. G: Games
        iv. PF
        v. Yds
        vi. Ply
        vii. Y/P
        viii. TO
        ix. FL
        x. 1stD
        xi. Cmp: Completed passes against defense
        xii. Att: Attempted passes against defense
        xiii. Yds: Passing yards against defense
        xiv. TD: Passing Touchdowns against Defense
        xv. Int: Interceptions
        xvi. NY/A
        xvii. 1stD
        xviii. Att
        xix. Yds: Rushing yards against defense
        xx. TD: Rushing Touchdowns against defense
        xxi. Y/A
        xxii. 1stD
        xxiii. Pen
        xxiv. Yds

xxv. 1stPy
xxvi. Sc%
xxvii. TO%
xxviii. EXP
xxix. Year
xxx. Mean: Mean average passing yards allowed (Column G) per year
xxxi. Standard_Deviation: Calculated Standard deviation per given year
xxxii. Z-Score: Z-score for a defense's strength based on yards allowed compared to other teams within the same season year
xxxiii. Min_yards: Min value of yards allowed by any team per given season
xxxiv. Max_yards: Max value of yards allowed by any team per given season
xxxv. Min_Max_norm: Normalized min-max value, with a range of 0.5 – 1.5. We later on chose to abandon this approach to normalization, and used binning instead based on Z-score.

Data Preprocessing:

There was a significant amount of preprocessing that had to be done to prepare these datasets for data mining analysis. At a high level, this included filtering the datasets to rows/columns of interest, joining the datasets together, and discretizing numeric variables into categorical values. Preprocessing details for each dataset are described below, which loosely follow the workflow order of operations within the R programming script we had provided.

Weather dataset:

1. Convert humidity variable from string to numeric
2. Convert game date from string into date format
3. Impute a value of 1 mph for all rows in the dataset that had a Wind mph value of 0. We felt there was a high likelihood the source data was inaccurate for these rows
4. Remove teams from dataset that play in a dome. These teams/game locations are not subject to weather conditions, which was our primary data mining focus
5. Convert team names to their respective acronyms (i.e. Buffalo Bills -> "BUF") so that this dataset could be joined to others that used team acronyms
6. Drop rows with NA values for humidity

Game Logs dataset:

1. Filter on rows where QBs started. We wanted to exclude game log entries for QBs that likely didn't play many minutes and had very few pass attempts
2. Convert game date from to proper date format
3. Convert game week from integer to character
4. Convert game completion percentage to a numeric variable
5. Drop rows with NA values for game completion percentage

6. Filtered only on regular season games. We excluded preseason games because QBs do not typically play their hardest in these games, and excluded post-season games because we thought this would be skewed to high-performing quarterbacks that aren't as susceptible to weather. Post-season games also had game dates that spilled over into subsequent years (such as early January), which would have complicated our ability to join datasets based on season-year.
7. Drop rows with NA values for passing yards

Career Stats dataset:

1. Convert team names to their respective acronyms (i.e. Buffalo Bills -> "BUF") so that this dataset could be joined to others that used team acronyms
2. Drop rows with NA values for season completion percentage

Defense dataset:

1. Team defense stats for 2000-2013 were 13 distinct datasets that had to be manually stacked together in an Excel file. In this Excel file, some columns were calculated:
    a. Mean passing yards allowed across all teams for a given season
    b. Standard deviation of passing yards allowed for all teams in a given season
    c. Z-score for passing yards allowed in a given season. This field was later used to categorize team defensive strength
        i. There are additional min/max, min/max_norm fields that were initially used as an attempt to use Min-Max normalization that would be used as a multiplier to normalize the passing yards thrown in a game. We later determined that this normalization method heavily skewed the results for QB passing yards in a game
2. Drop irrelevant columns. We were only interested in measuring team opponents' defensive strength based on how many passing yards a team allowed in a given season
3. Convert team names to their respective acronyms for purposes of joining with other datasets

Joining datasets:

1. Inner join Career Stats dataset with the Game Logs dataset based on matching season Year and QB name. Rename this "CareerGameLogs"
    a. Create additional column for game location using an if/else statement. If a game log was marked as "away" for a QB, then game location = Opponent. If it was a "home" game, then game location = the QB's team.
        i. This added field was used to join with the weather dataset. Without this field, the weather dataset would have only joined with the merged GameLogs/CareerStats dataset on games where the QB was playing a home game.
    b. Create a "Yards deviation" column that showed how a QB's passing yards in a game deviate from their season average (game log passing yards – season average passing

yards). This was the primary “performance” metric which was later discretized to be the class variable we tried to predict.

c. Create a “Game completion % deviation” column that showed how a QB’s passing completion rate in a game deviated from their season average (game log passing completion rate (%)– season average completion rate (%)).

2. Join the merged “CareerGameLogs” with the weather dataset, based on matching game date and game location. Rename this “weather_gamelogs_inner”.
3. Join the merged “weather_gamelogs_inner” with the Defense dataset by joining on matching season year and team acronym.

Binning/Discretizing variables of interest (Equal-With Binning):

1. Bin “Yards deviation” (our primary class variable that we are predicting) into 8 equal width bins. We chose these bin ranges, as seen in column 1 of the table below, based on our analysis of a quartiles and histogram visualization of this data. A high positive deviation implied an “Excellent game” while a low negative deviation implied a “Horrendous game”.
2. Bin “team defense strength” into 5 equal width bins. We chose these bin ranges, as seen in column 2 of the table below, based on our calculated Z-score for a team’s yards allowed in a given season compared to all other teams. A high Z-score implied an “top tier defense” while a low Z-score implied a “poor defense”.
3. Bin “temperature” into 5 equal width bins. We chose these bin ranges, as seen in column 3 of the table below, based on our analysis of a quartiles and histogram visualization of this data. A high temperature value >70 degrees was categorized as “hot”, and < 10 degrees was labeled “really cold”.
4. Bin “humidity” into 5 equal width bins. We chose these bin ranges, as seen in column 4 of the table below, based on our analysis of a quartiles and histogram visualization of this data. A high humidity percentage value of >80% degrees was categorized as “very humid”, and < 20% was labeled “dry”.
5. Bin “Wind (mph)” into 4 equal width bins. We chose these bin ranges, as seen in column 5 of the table below, based on our analysis of a quartiles and histogram visualization of this data. A strong wind of > 22.5 mph was categorized as “very windy”, and < 7.5 mph was labeled “low wind”.

The table below provides additional detail into how our numeric variables were discretized.

Table A: Binning Categories

Yards thrown deviation from season average-passing yards/game. Units: passing yards	Opponent Defense Strength based on Z-Score for average passing yards allowed	Temperature	Humidity	Wind (mph)	Completion pct (no predefined levels; system chose 5 bins based on range) Units: accuracy
excellent game	**top tier defense**	**really cold**	**dry**	**low wind**	**Excellent game**
>= 150	2.05	below 10	0-20	0-7.5	5th bin floor ->
great game	**very good defense**	**cold**	**below normal humidity**	**breezy**	**Good game**
>= 100 to < 150	0.83 to 2.05	10-30	20-40	7.5-15	4th bin floor - 5th bin
good game	**good defense**	**cool**	**normal humidity**	**windy**	**Average game**
>= 50 to < 100	-0.39 to 0.83	30-50	40-60	15-22.5	3rd bin floor - 4th bin
slightly above average game	**average defense**	**warm**	**humid**	**very windy**	**Below average game**
>= 0 to < 50	-1.62 to -0.39	50-70	60-80	22.5+	2nd bin floor - 3rd bin
slightly below average game	**poor defense**	**hot**	**very humid**		**poor game**
<= 0 to > -50	-2.85 to -1.62	70+	80-100		1st bin floor- 2nd bin
bad game					
<= -50 to > -100					
awful game					
<= -100 to > -150					
horrendous game					
<= -150					

Cleaning up dataset for model training and testing:

1. After all dataset merging and formatting, the final dataset used for our data mining algorithms was cleaned up to a subset of 23 columns. These columns were: opponent_acr, game_completion_pct, qb_name, qb_team, game_date_year, game_date_month, game_date_day, week, home.or.away, away_team, home_score, away_score, outcome, game_passes_attempted, game_completion_pct.1, season_completion_pct, season_passer_rating, temperature_cat, humidity_pct_cat, wind_mph_cat, defense_quality, game_completion_pct_cat, **yards_dev_cat**.
    a. Yards_dev_cat was our class variable that was used to predict QB performance.

**********Supplementary preprocessing**********

Binning/Discretizing variables of interest (Equal-Depth Binning):

After learning that our initial model performed poorly (additional detail below), we decided to explore using equal-depth binning. The preprocessing for this was as follows:

1. "Yards deviation" (our primary class variable that we are predicting): We chose to use the same number of bins as used in equal-width binning (8). We had to determine the total number of rows in the dataset, and divide by 8, to determine the number of tuples in each bin. A column was then added for categorization purposes. We then sorted the data frame in ascending order of yards deviation and assigned categorization values based on the index position of each tuple. We applied the same categorization labels as used for equal-width binning.
2. Defense strength: We chose to use the same number of bins as used in equal-width binning (5). We had to determine the total number of rows in the dataset, and divide by 5, to determine the number of tuples in each bin. A column was then added for categorization purposes. We then sorted the data frame in ascending order of defense "Z-score" and assigned categorization values based on the index position of each tuple. We applied the same categorization labels as used for equal-width binning.
3. Temperature: We chose to use the same number of bins as used in equal-width binning (5). We had to determine the total number of rows in the dataset, and divide by 5, to determine the number of tuples in each bin. A column was then added for categorization purposes. We then sorted the data frame in ascending order of temperature and assigned categorization values based on the index position of each tuple. We applied the same categorization labels as used for equal-width binning.
4. Humidity: We chose to use the same number of bins as used in equal-width binning (5). We had to determine the total number of rows in the dataset, and divide by 5, to determine the number of tuples in each bin. A column was then added for categorization purposes. We then sorted the data frame in ascending order of humidity (%) and assigned categorization values based on the index position of each tuple. We applied the same categorization labels as used for equal-width binning.
5. Wind: We chose to use the same number of bins as used in equal-width binning (4). We had to determine the total number of rows in the dataset, and divide by 4, to determine the number of tuples in each bin. A column was then added for categorization purposes. We then sorted the

data frame in ascending order of wind strength (mph) and assigned categorization values based on the index position of each tuple. We applied the same categorization labels as used for equal-width binning.

Beyond this, additional coding was done for data exploration, which included scatter plot visualization, histograms, group-by functions, and outer joins (used for QC'ing table join integrity) that were later stashed.

Data Mining Process and Analysis

Tools and Methodologies Used:

After preprocessing, we primarily used Weka to test and analyze different data mining algorithms. We also integrated JMP Pro on a limited basis to assist with visual representations. Our goal was to determine which data mining model had the highest accuracy in predicting our class attribute "yards deviation". Each model was comprised of two components: an attribute selection algorithm and a classifier algorithm. We ended up testing 4 classifier algorithms paired with 5 distinct sets of attributes for a combined total of 20 different classifier models tested. For each of the 20 models tested, we used the 10-fold cross validation method, which splits the entire dataset into 2/3 training data and 1/3 testing dataset.

Attribution Selection:

Our initial interest for this project was to determine how much weather variables (temperature, humidity, and wind mph) and opponent defense strength would affect the performance of Quarterback performance (yards deviation). For purposes of this project, we also expanded our pool of potential attributes to ones that were not completely independent of yards deviation but were still of interest. Some examples of this include pass completion percentage and passes attempted (both of which would likely have a positive correlation with passing yards deviation). We used several different attribution selection methods in Weka to help us decide which set of attributes would provide the highest accuracy.

The table below shows the results of 4 attribution selection algorithms we tested, as well as our own hand-picked attributes. The attributes in each column are listed in order of ranking for which ones have the highest impact in predicting QB performance (yards deviation), as indicated by the numbers indicating weighted significance. Some of these attribution selection algorithms required a pre-selected search algorithm; these cases are specified in the table. While our goal was to pick the model with highest accuracy, we also had a secondary goal of determining which (if any) weather attributes had

relevance in predicting yards deviation. In the table, we highlighted the first occurrence of any weather attribute that was selected among each algorithm. In each of the algorithms, wind was the highest-ranking attribute, which implied it was the most influential weather variable in predicting a QB's passing yards.

Table B: Attribute Selection algorithms

ONE R (Ranker search w/ranked weights)	CFS SUBSET EVAL-BEST FIRST (Greedy Stepwise search)	INFO GAIN (Ranker search w/ranked weights)	CORRELATION ATTRIBUTE EVAL (Ranker search w/ranked weights)	OUR SELECTED ATTRIBUTES
33.5582 15 game_passes_attempted	home_score	0.290774 15 game_passes_attempted	0.1728 15 game_passes_attempted	Wind
29.3634 23 game_completion_pct_cat	away_score	0.173422 4 qb_name	0.09789 23 game_completion_pct_cat	Defense quality
28.4248 13 away_score	game_passes_attempted	0.097417 3 game_completion_pct	0.09514 3 game_completion_pct	Temperature
28.3661 22 defense_quality	wind_mph_cat	0.096595 23 game_completion_pct_cat	0.08394 13 away_score	Humidity
27.6034 21 wind_mph_cat	defense_quality	0.053466 5 qb_team	0.05205 12 home_score	
27.0461 2 opponent_acr	game_completion_pct_cat	0.052243 2 opponent_acr	0.03992 22 defense_quality	
26.694 3 game_completion_pct		0.052212 11 away_team	0.03684 14 outcome	
26.43 5 qb_team		0.04141 22 defense_quality	0.01684 9 week	
26.3714 7 game_date_month		0.038307 13 away_score	0.0168 21 wind_mph_cat	
26.342 6 game_date_year		0.029048 12 home_score	0.01555 17 season_completion_pct	
26.166 10 home.or.away		0.013613 21 wind_mph_cat	0.01519 19 temperature_cat	
25.726 4 qb_name		0.009728 14 outcome	0.01515 2 opponent_acr	

For some of the algorithms, we did not include all attributes that were ranked into the table above. We chose to limit the attribute count to 12, however, the CFS Subset Eval- Best First selection method only supplied 6 attributes. Our next step in building a predictive model was testing different classifier algorithms on each set of attributes selected from the attribute algorithms shown above.

Classifier Algorithms:

For each of the 5 unique sets of attributes, we tested 4 distinct classifier algorithms. The different classifier algorithms we used were Naïve Bayes, J48, OneR, and Random Forest. In the Weka Screenshots document, we provided performance results for each of the 20 models tested. These results include metrics on True Positive (TP) Rate, False Positive (FP) rate, Recall, F-Measure, ROC area, and Confusion Matrices.

We tested each of these algorithms using 10-fold cross validation. We had briefly tested manually splitting the training and testing dataset to be 66% and 34%, respectively (which had slightly more accurate results), but felt that the 10-fold method was a more reliable technique for sampling. 10-folds helps ensure consistency by partitioning the data into 10 equal components (349 tuples in our case) and using the remaining 9 partitions as the training data. This process is repeated 10 times using a subsequent partition as the test data. As it cycles through, all data is eventually used for both training and test data.

Selecting our Model of Choice:

In order to determine which model was optimal for our data mining goal, we wanted to balance having a model with high accuracy that also included predictor variables that were of interest to us. After testing each of these models, we felt that the combination of the CFS Subset Evaluator- Best First selection method and Naive Bayes classifier algorithm (referred to as CFS/NB) was our best model. We've provided detailed explanation below for why we chose this model, and also compared it more closely to a model that used our hand-picked weather/defense attributes.

The CFS/NB model had a True Positive rate of 35.9636%, which was the highest among any other models. It was also one of the few models to predict better than a 50% TP rate on any of the categories of our class variable (yards deviation); We felt the overall TP rate was the best indicator of model "performance" because it aligned with our data mining goal of correctly classifying game performance (yards deviation) as accurately as possible . This rate, along with the False Positive rate, provided quick insight into the success (or lack thereof) of the selector/ classifier combination. However, we did take other model performance metrics into consideration. All other models had a TP rate within the range of 36% and 24.45%. Despite the low performance, we proceeded with the CFS/NB model because it greatly surpassed any model that used our own hand-picked attributes of interest (which we would have considered selecting if the performance was more comparable to CFS/NB). We also felt this model was a

good choice because it utilized a more succinct list of attributes than attribute selection algorithms used in other models. Therefore, any conclusions inferred regarding the impact of specific variables wouldn't be as convoluted by the inclusion of many too many covariates. The main drawback of this model was that it included some variables included that were not completely independent of QB game performance (such as game completion % and passes attempted). This was especially concerning because the model used Naïve Bayes, which is typically "naïve" to variables that have high interdependency.

We were surprised and somewhat disappointed that our best model had such as low 35% TP rate. We initially thought this low performance was due to our discretized variables (using Equal-Width binning) having a very normal distribution with small standard deviation. We discovered that the distribution of tuples in each category of certain variables was very unequal. As seen below, one example of this was for the team defense attribute. Our concern was that the minimal variance between these discretized values (80% of tuples having "average" or "good defense") would diminish this variable's effectiveness as a reliable predictor of yards deviation.

testing_data$defense_quality	Freq
average defense	1166
good defense	1535
poor defense	151
top tier defense	105
very good defense	462

To address this concern and poor model performance, we changed our method for binning numeric variables to use Equal-Depth binning instead. Unfortunately, the models we re-tested ended up performing significantly worse, and averaged a ~16% TP rate. If we had more time, we would have tested out other ways to improve our model performance. One additional tactic we could have tried would be to reduce the number of bin categories for our class variable and/or predictor variables.

Additional Model Performance Analysis:

Although we had already selected our model based on its TP rate, we analyzed additional performance metrics in comparison to other models. We also wanted to have a better understanding of why CFS/NB performed so much better than any model using our hand-picked attributes.

Confusion Matrix

We reviewed the Weka-supplied confusion matrix which was generated by the classifier. The matrix listed the predicted values generated of the test data across the horizontal lines, and the actual values on the vertical lines. It was initially difficult to interpret the confusion matrix output of the Weka models because the categories were not listed in their ordinal order. After selecting the model to analyze, a reordering of the output was required for ease of analysis and to identify a visual pattern. The CFS/ NB model we chose is shown below, as well as our initial (ultimately rejected) choice for comparison purposes (original Weka labels left intact).

When analyzing the confusion matrices, we discovered that the highest concentration of TP accuracy occurred in the class variable categories that had the highest number of tuples for categories such as "slightly above average game" or "below average game". This is logical, as these align with the normal distribution expected from typical game results.

Figure C: Original Weka Confusion Matrix (CFS/NB)

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   <-- classified as
 387 330 101   0  34   3   3   6 |   a = slightly below average game (passing yards)
 274 486  36   4  95   2   9   1 |   b = slightly above average game (passing yards)
 248 115 137   0   2  10   0   6 |   c = bad game (passing yards)
   6  16   0  11  48   0  16   0 |   d = excellent game (passing yards)
  93 268   4   6 152   0  29   0 |   e = good game (passing yards)
  68  16  62   0   0  18   0  11 |   f = awful game (passing yards)
  29  90   1   9  92   0  23   0 |   g = great game (passing yards)
   7   2  15   0   0  16   0  12 |   h = horrendous game (passing yards)
```

Table C: Reordered Confusion Matrices

d	g	e	b	a	c	f	h	**CFS Subset Evaluator- Best First/ Naive Bayes**
11	16	48	16	6	0	0	0	d = excellent game (passing yards)
9	23	92	90	29	1	0	0	g = great game (passing yards)
6	29	152	268	93	4	0	0	e = good game (passing yards)
4	9	95	486	274	36	2	1	b = slightly above average game (passing yards)
0	3	34	330	387	101	3	6	a = slightly below average game (passing yards)
0	0	2	115	248	137	10	6	c = bad game (passing yards)
0	0	0	16	68	62	18	11	f = awful game (passing yards)
0	0	0	2	7	15	16	12	h = horrendous game (passing yards)

d	g	e	b	a	c	f	h	**Our selected attributes: Naive Bayes**
0	2	5	73	17	0	0	0	d = excellent game (passing yards)
2	0	13	160	62	6	0	1	g = great game (passing yards)
0	1	22	356	164	9	0	0	e = good game (passing yards)
1	3	25	546	316	16	0	0	b = slightly above average game (passing yards)
1	2	9	446	377	29	0	0	a = slightly below average game (passing yards)
0	3	4	241	240	29	0	1	c = bad game (passing yards)
0	1	2	82	81	9	0	0	f = awful game (passing yards)
0	0	0	25	22	5	0	0	h = horrendous game (passing yards)

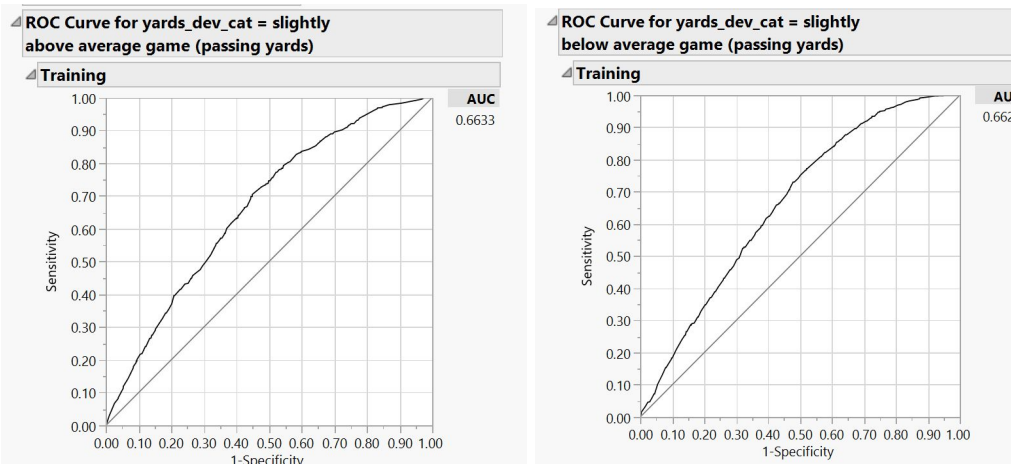
Additional Comparison Metrics: F-Measure

We additionally focused on the precision and recall of the CFS/NB approach and observed these were of similar ratios between the variants of selections/ classifiers which performed well. We used the F-measures for comparison, since it provided a balanced measure of precision and recall, and also narrowed our analysis to the middle categories that had the highest number of tuples. The F-measure for the CFS/NB model was relatively high compared to other models for the “slightly above average game” and “slightly below average” categories, with values of 0.392 and 0.436, respectively. Other models with similar F-measure performance included the One R selection/ One R classifier model, the Info Gain selection/ One R classifier model, and the Correlation Attribute selection/ One R classifier model (all with 0.403/ 0.394, respectively). Running random simulation tests using Naive Bayes methods in R on test data sets produced similar results and produced no notable anomalies or significant results out of line with our evaluations using the classifiers. Exploring the One R classification method under a more robust time frame would be an area of interest for further study, as it appears the OneR classifier has a higher accuracy overall with any attribute selection method.

Additional Comparison Metrics: ROC/AUC

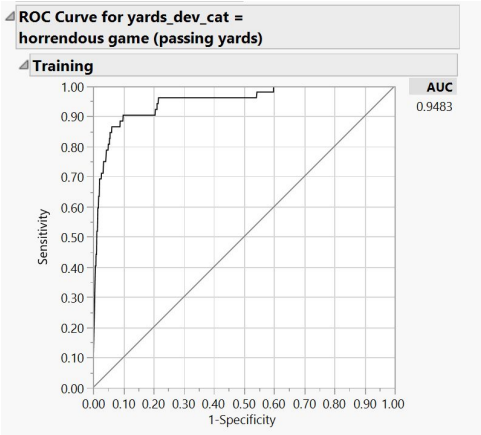
For further analysis, we assessed how ROC curves and the area under the curve (AUC) differed between our chosen CFS/NB model compared to other classifier models with similar F-measures. The ROC curves provided us a visualization of the true positive rate vs. the false positive rate. The figures below show the ROC curves for our chosen CFS/NB model for the middle categories (slightly above/below average). As seen by the concave that is above the diagonal line, this model has decent performance and would be more accurate than simply flipping a coin to classify game performance.

Figure A: AUC of populous classes (CFS/NB)



We additionally evaluated the AUC for the more sparsely populated categories/outliers and discovered that the TP Rate performed much better than in the middle categories. If we had more time, we would have explored this implication more, and perhaps focused our models on predicting outlier performance (very good or bad games).

Figure B: AUC, Outlier (CFS/NB)



In Table D, we summarized some of the additional metrics we compared for models that had similar performance. We also included comparison of a model using Naïve Bayes on our hand-picked attributes, which barely performed better than a toss of a coin! As seen below, the CFS/OneR and Correlation Attribute/NB models (one of the better performing models) had slightly lower performance across all key metrics in comparison our chosen CFS/NB model.

Table D: Comparison Samples

Selection/ Classifier	Correctly classified	F-measure;above avg game	F-measure; below avg game	AUC, slightly above avg game	AUC, below avg game
CFS/NB	35.96	0.436	0.392	0.651	0.649
CFS/OneR	33.56	0.406	0.387	0.578	0.566
Correlation Attribute/ NB	35.09	0.409	0.364	0.624	0.646
Our method/ NB	28.57	0.385	0.352	0.519	0.543

Conclusion

While we were disappointed by the poor performance of classification models using our hand-picked attributes of interest relating to weather and defense, some interesting insights were uncovered. While we initially chose our model just based on TP rate, it was reassuring to see that it also outperformed other models in F-measure, ROC/AUC, and other key metrics. If given more time, we would have tested different methods to improve the accuracy of models using our hand-picked attributes. Unfortunately, the majority of our time was spent on data preprocessing, joining disparate datasets, and binning. While we cannot conclusively say that weather and defense have minimal impact on game performance until further testing is done, we are reasonably confident that other attributes are more significant predictors. This is not surprising, since some of those attributes in our chosen model were tangentially related to game performance (e.g. game completion percentage and passes attempted). We were content that our chosen model at least included two of our attributes of interest that were completely independent of QB performance (wind and defense strength). For the time being, we do not advise Fantasy Football owners to make weekly roster decisions based on temperature or humidity.