# CS 165A – Artificial Intelligence, Fall 2024
## Machine Problem 1

Responsible TA: Xuan Luo, xuan_luo@ucsb.edu

Due: Nov 12, 2024 11:59pm

## 1 Notes

■ Make sure to read the "Policy on Academic Integrity" in the course syllabus.
■ Any updates or corrections will be posted on Canvas.
■ You must work individually on this assignment.
■ Each student must submit their report and code electronically.
■ If you have any questions about MP1, please post them on the Canvas MP1 Q&A.
■ Plagiarism Warning: We will use software to check for plagiarism. You are expected to complete the project independently without using any code from others.

## 2 Background: Basketball Win/Loss Prediction

Sports analytics is a rapidly growing field that merges the excitement of sports with the analytical power of data science. By analyzing vast amounts of statistical data—including team performance metrics, player statistics, and historical trends—organizations can gain valuable insights into how and why games are won or lost. This data-driven approach has become an essential tool in professional sports, with organizations frequently hiring data analysts to examine matches and adjust team strategies accordingly, providing a significant competitive advantage.

In this context, our project aims to develop a Naive Bayes classifier to predict whether the home team will win or lose a basketball game based on various game statistics. The dataset includes information for both home and away teams, such as recent performance, field goal percentages, rebounds, assists, and more. This project offers newcomers to artificial intelligence a hands-on opportunity to apply machine learning techniques to real-world data.

## 3 Data Format

The dataset used for this project contains various statistics from past basketball games. It includes both numerical and categorical features. The table below provides an example of the dataset format:

### 3.1 Categorical and Numerical Variables

The following columns are categorical variables:

- team_abbreviation_home
- team_abbreviation_away
- season_type
- home_wl_pre5 (W/L strings)
- away_wl_pre5 (W/L strings)

The following columns are numerical variables:

| Column Name | Description | Example |
|---|---|---|
| team_abbreviation_home | Abbreviation of the home team's name | NYK |
| team_abbreviation_away | Abbreviation of the away team's name | BKN |
| season_type | Type of the season (e.g., Regular Season) | Regular Season |
| label | Game outcome (1 for home win, 0 for home loss) | 1 |
| min_avg5 | Average minutes played in the last 5 games | 245 |
| fg_pct_home_avg5 | Home team's average field goal percentage over the last 5 games | 0.4786 |
| fg3_pct_home_avg5 | Home team's average three-point percentage over the last 5 games | 0.378 |
| ft_pct_home_avg5 | Home team's average free throw percentage over the last 5 games | 0.7466 |
| oreb_home_avg5 | Home team's average offensive rebounds over the last 5 games | 12.2 |
| dreb_home_avg5 | Home team's average defensive rebounds over the last 5 games | 33.6 |
| reb_home_avg5 | Home team's average rebounds over the last 5 games | 45.8 |
| ast_home_avg5 | Home team's average assists over the last 5 games | 22.4 |
| stl_home_avg5 | Home team's average steals over the last 5 games | 5.8 |
| blk_home_avg5 | Home team's average blocks over the last 5 games | 2.2 |
| tov_home_avg5 | Home team's average turnovers over the last 5 games | 10.8 |
| pf_home_avg5 | Home team's average personal fouls over the last 5 games | 24.6 |
| pts_home_avg5 | Home team's average points over the last 5 games | 118.4 |
| home_wl_pre5 | Home team's win/loss record in the last 5 games | WWLWW |
| fg_pct_away_avg5 | Away team's average field goal percentage over the last 5 games | 0.4872 |
| fg3_pct_away_avg5 | Away team's average three-point percentage over the last 5 games | 0.299 |
| ft_pct_away_avg5 | Away team's average free throw percentage over the last 5 games | 0.8062 |
| oreb_away_avg5 | Away team's average offensive rebounds over the last 5 games | 8 |
| dreb_away_avg5 | Away team's average defensive rebounds over the last 5 games | 31.6 |
| reb_away_avg5 | Away team's average rebounds over the last 5 games | 39.6 |
| ast_away_avg5 | Away team's average assists over the last 5 games | 24.6 |
| stl_away_avg5 | Away team's average steals over the last 5 games | 11.2 |
| blk_away_avg5 | Away team's average blocks over the last 5 games | 4.4 |
| tov_away_avg5 | Away team's average turnovers over the last 5 games | 17.6 |
| pf_away_avg5 | Away team's average personal fouls over the last 5 games | 22.6 |
| pts_away_avg5 | Away team's average points over the last 5 games | 105.6 |
| away_wl_pre5 | Away team's win/loss record in the last 5 games | LWLWW |

Table 1: Data Format and Example

- min_avg5, fg_pct_home_avg5, fg3_pct_home_avg5, ft_pct_home_avg5, oreb_home_avg5, dreb_home_avg5, reb_home_avg5, ast_home_avg5, stl_home_avg5, blk_home_avg5, tov_home_avg5, pf_home_avg5, pts_home_avg5
- fg_pct_away_avg5, fg3_pct_away_avg5, ft_pct_away_avg5, oreb_away_avg5, dreb_away_avg5, reb_away_avg5, ast_away_avg5, stl_away_avg5, blk_away_avg5, tov_away_avg5, pf_away_avg5, pts_away_avg5

For the labels in `home_wl_pre5` and `away_wl_pre5`, a win (W) is encoded as 1 and a loss (L) is encoded as 0. For example, `WWLWW` would be represented as {1, 1, 0, 1, 1}.

## 3.2 Dataset Details

The dataset is split into two parts:

- **train_data.csv**: This contains 10,000 rows of training data.
- **validation.csv**: This contains 1,000 rows of validation data.

Students are expected to use the training data to compute the Naive Bayes statistics (e.g., mean, variance for numerical variables, and probabilities for categorical variables), and evaluate the accuracy of their Naive Bayes model on the validation data.

It is important to note that using pre-existing Naïve Bayes libraries, such as those in Scikit-learn, is not allowed. You must implement the Naïve Bayes algorithm from scratch. For this programming assignment, you may use Python 3 along with any built-in modules, as well as numpy, pandas, or scipy.

# 4 Program Input, Output, and Submission Guidelines

Your program must handle two file paths provided as command-line arguments. The first path should lead to a training dataset named `./train_data.csv`, and the second path should point to the testing dataset. The tasks for your program are to: (1) train the Naive Bayes classifier using `train_data.csv`, and (2) evaluate the classifier on a provided test dataset (such as `validation_data.csv`) and output the predictions to the terminal. For each data point in the test dataset, the output should be a single integer, either 0 (for loss) or 1 (for win).

The accuracy of your program will be evaluated based on its performance on a secret test set, `test_data.csv`, which will be used for grading. So don't use the public testing dataset `validation_data.csv` to train your classifier hoping to achieve greater accuracy. This is called overfitting and it would make your classifier very accurate on this specific dataset only, but not on unseen one. Also, there will be a time limit of 10 minutes and your program must finish execution within this time. If it runs for more than 10 minutes, the grader will terminate your program and your accuracy score will be 0.

## 4.1 Program Execution

Your program should be executable from the command line using Python3, and it should read the two required file paths as command-line arguments. The first argument should point to the training data, and the second should point to the test data. The program should print the predictions (0 or 1) directly to the terminal, with each prediction on a new line, corresponding to each test example in the dataset. **The output should not be written to a file.**

**Example Input and Output** Below is an example of how your program should be executed and the format of the expected output:

**Example Input:**

```
python3 NaiveBayesClassifier.py train_data.csv validation_data.csv
```

**Example Output:**

```
0
1
0
1
...
(1000 rows in total)
```

## 4.2 Gradescope Submission

We will create two distinct assignments on Gradescope (`https://www.gradescope.com/courses/880446`) for this module:

**MP1_report:** Submit only your PDF report to this assignment.

**MP1_code:** Submit all code-related documents here, including:

- A shell script named `NaiveBayesClassifier.sh`
- Your source code, e.g., `NaiveBayesClassifier.py`
- Any additional scripts required for execution

**Important: Do not submit any data files**. Also, do not place your scripts or code inside a directory or zip file. The autograder should be able to run your code and provide accuracy results within a few minutes. While there is no limit to the number of submissions, only the last submission will be considered for grading.

### 4.3 Execution and Environment

- **Executable File:** Ensure your program's executable is named `NaiveBayesClassifier`.
- **Wrapper Script:** Include a shell script named `NaiveBayesClassifier.sh` that executes your Python code. Example usage:

  ```
  ./NaiveBayesClassifier.sh train_data.csv test_data.csv
  ```

  This script should allow command line arguments to be passed directly to your Python script.
- **Python Version:** Clearly state the version of Python used in your project within your report. Ensure all code is compatible with this version. We will use python 3 on gradescope.

**Note:** If the Gradescope autograder does not support a package you require, please make a post on Canvas. We aim to accommodate all necessary tools to ensure your code runs smoothly.

## 5 Grading Criteria

### 5.1 Grading Breakdown

The final grade for this project will be determined based on the following criteria:

- **Complete Report:** 20%
- **Execution Specifications:** 10%
- **Correct Program Specifications:** 10%
  - Reads file names from command-line arguments
  - Produces correct standard output results
  - No segfaults or unhandled exceptions
- **Test Accuracy and Runtime:** 60%

### 5.2 Leaderboard and Bonuses

- The top scorer in classifier testing accuracy will receive a 50% bonus on their project grade.
- The top three scorers will each receive a 25% bonus.
- A real-time leaderboard will be available on Gradescope to track these rankings.

### 5.3 Accuracy and Time

The majority of the grade (60%) is based on the testing accuracy and runtime:

- **Testing Accuracy:** Assessed on a private dataset.
- **Runtime:** Your code must complete execution within 10 minutes; otherwise, it will be terminated, and the accuracy score will be zero.

#### Score Calculation

- A naive implementation that achieves less than 0.5 accuracy will receive a 0% score for the accuracy component.
- Scores for accuracy levels:
  - $0.50 \leq$ Accuracy $< 0.60$: Scores will be linearly interpolated between 0% and 80%.
  - $0.60 \leq$ Accuracy $< 0.70$: Scores will be linearly interpolated between 80% and 100%.
  - $0.70 \leq$ Accuracy $\leq 1.00$: Scores will be linearly interpolated between 100% and 120%.
- A correct implementation of Naive Bayes should achieve above 0.65 accuracy.

This scoring is designed to reward improvements beyond the basic requirement and encourage innovative solutions that enhance the model's accuracy.

# 6 Report Guidelines

The project report should adhere to the following specifications:

- **Length:** The report must be between 1 and 3 pages, with no exceptions exceeding this limit.
- **Font Size:** The text must be at least 11pt in size.
- **Content Requirements:** Your report should comprehensively cover the following sections:
    1. **Architecture:** Provide a concise explanation of your code's architecture, including a description of classes and their basic functionalities.
    2. **Preprocessing:** Describe how you represent an instance within the dataset.
    3. **Model Building:** Explain the process of training your Naive Bayes Classifier.
    4. **Results:** Discuss your results on the provided datasets, including accuracy and running time.
    5. **Challenges:** Detail the challenges you encountered during the project and how you addressed them.
- **Submission Format:** Submit only one PDF report containing all the sections listed above. Do not include additional README files.
- **Identification:** Your report must include your name, email, and perm number at the top of the first page.

Please ensure all sections are clearly defined and thoroughly addressed to meet the project's evaluation criteria.