ML 1- Yilmaz

2024-2025

Name of the group members (max 2 people): **Aidan Lee, Max Weinstein**

If you are working alone, please state it clearly.

If you are working in groups, only one group member should submit the proposal.

Must haves:

* Link to dataset: **https://data.scorenetwork.org/hockey/nhl-shots.html**
* Information about your data
  + Meaning of attributes:
    - **Descriptions of the distance of the puck from the goal, current score, teams, whether or not the goal is empty at the time of the shot**
  + Dimension
    - **21 attributes**
  + Number of instances
    - **160,573**
  + How many missing values

event\_goalie\_name - Missing Values: 71509

empty\_net - Missing Values: 151861

event\_player\_2\_name - Missing Values: 374

event\_player\_2\_type - Missing Values: 374

strength\_code - Missing Values: 306

x\_fixed - Missing Values: 13

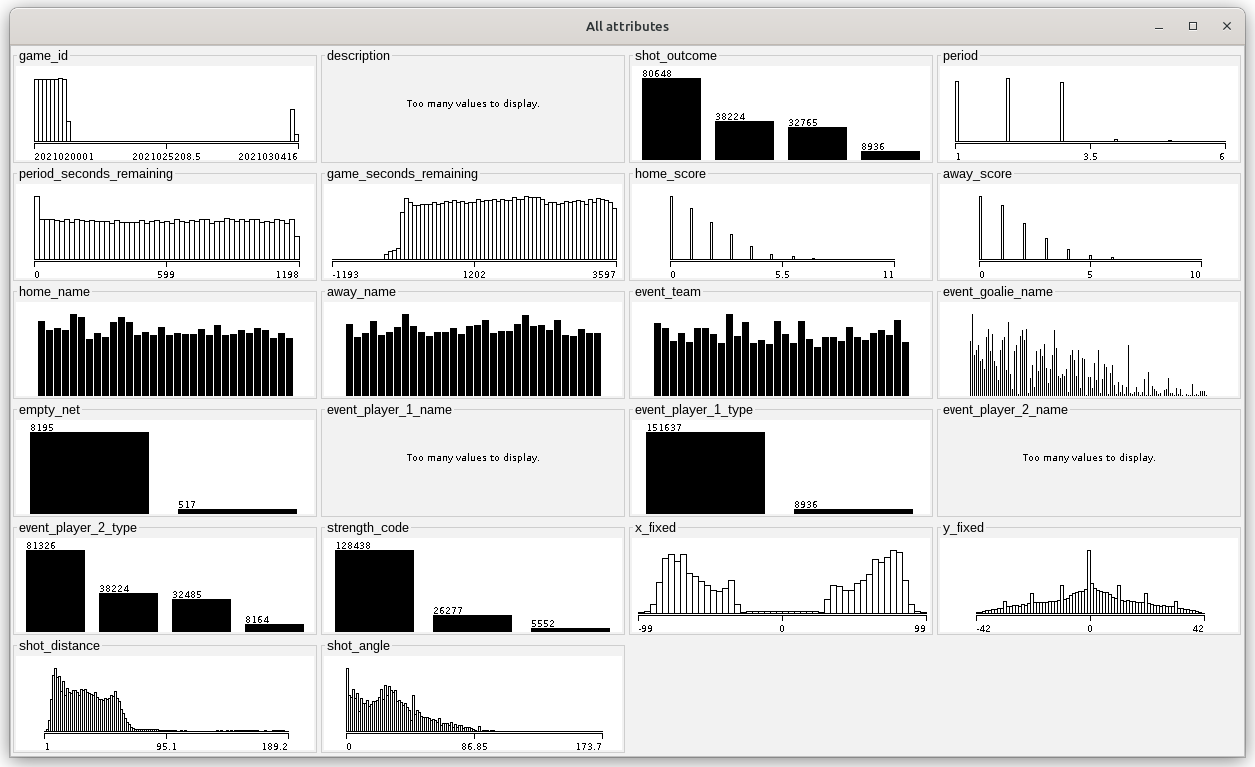
y\_fixed - Missing Values: 13

shot\_distance - Missing Values: 38224

shot\_angle - Missing Values: 38224



* + Is it uniform? Skewed?
    - **Data is mostly uniform, with some notable skews @ the shots distance and angle. The class variable is skewed towards being blocked by the goalie, with very little shots going in.**

****

* + What are the class distributions?

**Almost 50% of the shots were blocked by a goalie, 22% blocked by a non-goalie, 20% missed the frame of the goal, and the remaining went in**

* What are you classifying/predicting?

**The result of the shot (goal / blocked / missed), class variable: shot\_outcome**

* How will this be useful?

**We can predict whether or not shots will go in from position, maybe for a training tool for the team, or when watching back film they can tell when they should’ve shot**

* Plans for preprocessing

**Correct the empty values, might have to drop the values of missing distance because that would probably be heavily related to going in or not, but we would run a 70-15-15 split**

Don’t just insert your answers in the above, make it look like you're writing the portions of the final report. See the final report sample I posted at Schoology.

As soon as you are ready with your proposal, please see me, so that I can approve it.

Also, please add your project to the spreadsheet I will provide under Q1 Project Folder.

**Statement/Project Goal**

In hockey games, there are many shots that take place. On average, there are around 30 shots per hockey game. So, the goal of this project is to use a dataset to predict the result of a hockey shot (goal/blocked/missed) based on 21 different attributes. These attributes include things like whether the net is empty or not, the distance of the shot, and the angle of the shot. This would be used for predicting whether or not shots will go in from a specific position or situation during a game. Teams could use it as a training tool for their team. When they are watching film, they can tell when they should have actually shot the puck. Additionally, many hockey commentators mention something called expected goals. An expected goals model determines how likely a shot is going to go in, taken from the moment before the player shoots, based on various different factors. We can use this dataset to create our own expected goals model.

**Description of Dataset**

The link to our dataset is <https://data.scorenetwork.org/hockey/nhl-shots.html>. Our dataset is from the SCORE Network Sports Data Repository. This data repository considers data information from all the shots in the 2021-2022 National Hockey League season. Our dataset includes data on the result of a shot and different attributes, such as the distance and angle of the shot. The dataset has 160,573 different instances and 21 different attributes. Specifically, our 21 different attributes listed off are: game\_id, description, period, period\_seconds\_remaining, game\_seconds\_remaining, home\_score, away\_score , home\_name, away\_name, event\_team, event\_goalie\_name, empty\_net, event\_player\_1\_name, event\_player\_1\_type, event\_player\_2\_name, event\_player\_2\_type, strength\_code, x\_fixed, y\_fixed, shot\_distance, and shot\_angle. The data is mostly uniform with some notable skews in the shots distance and angle. The class variable is skewed towards being blocked by the goalie, with very little shots going in. Almost 50% of the shots were blocked by a goalie, 22% blocked by a non-goalie, 20% missed the frame of the goal, and the remaining went in.

**Preprocessing**

In order to start working on the model, we would need to preprocess the data first. In order to use a model classifier or attribute selection algorithm, we first would remove unnecessary attributes, address the missing values, and split the dataset into training/test/validation sets.

The first issue we need to address is the unnecessary attributes. All of the name attributes will not be used in the algorithm as they just state the name of the player, they aren’t really necessary in determining whether or not a shot will go in. These attributes are description, home\_name, away\_name, event\_team, event\_goalie\_name, event\_player\_1\_name, and event\_player\_2\_name.

The second issue we need to fix in the preprocessing step is the missing values. The following is the distribution of missing values in our dataset:

event\_goalie\_name - Missing Values: 71509

empty\_net - Missing Values: 151861

event\_player\_2\_name - Missing Values: 374

event\_player\_2\_type - Missing Values: 374

strength\_code - Missing Values: 306

x\_fixed - Missing Values: 13

y\_fixed - Missing Values: 13

shot\_distance - Missing Values: 38224

shot\_angle - Missing Values: 38224

We already addressed event\_goalie\_name and event\_player\_2\_name by removing the attributes entirely since they are just the names of players. In the case of empty\_net, it says NA when the goal is empty. We can replace all of the NA with TRUE and keep all of the FALSE as is. For strength\_code, x\_fixed, y\_fixed, shot\_distance, and shot\_angle, we will remove all the instances that have a missing value.

The last thing we need to do is split the dataset into training/test/validation sets. We will randomly assign 70% of the instances to the training set, we will randomly assign 15% of the instances to the test set, and the remaining 15% will be assigned to the validation set.