AI Agent for League of Legends Match Prediction Using Bayesian Network

Jason Cheung

February 17, 2025

Abstract

This AI agent is designed to predict match outcomes in League of Legends based on in-game statistics. Using a dataset containing high-ranked matches with detailed in-game metrics—such as minion kills, gold earned, experience gained, jungle control, and objective captures—the agent analyzes early-game conditions and predicts whether a team will win or lose. The performance measure of the agent is the accuracy of match predicting outcomes, whereas the environment is the stats of League of Legends Platinum Ranked Games (starting with minute 10 with strides of 2 minutes). Actuators are displaying match prediction output and Sensors are the Riot League of Legends Game API. The agent operates as a goal-based AI, focusing solely on maximizing the accuracy of its match outcome predictions. Given the strong conditional dependencies among in-game variables (e.g., minion kills correlating with gold and experience leads) along with the binary nature of predictions makes probabilistic agents such as Bayesian networks a good fit.

1 INTRODUCTION

This is the product rule decomposition of the calculation of the AI Agent for League of Legends Match Prediction Using Bayesian Network.

The probability we're looking for is

$$P(hasWon)$$

. With the given Bayesian network and CPTs, we can do the following:

Let $goldDiff = gd$, $expDiff = ed$, $laneProgression = lp$, $kda = k$, $wardsDiff = wd$, $isFirstBlood = fb$, $isFirstTower = ft$, $killedRiftHerald = rh$, $drakeDiff = dd$.

$$P(hasWon)$$
$$= \sum_{gd,ed,lp,k,wd,fb,ft,rh,dd} P(hasWon, gd, ed, lp, k, wd, fb, ft, rh, dd)$$

(By the chain rule of probability)

$$= \sum_{gd,ed,lp,k,wd,fb,ft,rh,dd} P(hasWon \mid gd, ed, lp, k, wd, fb, ft, rh, dd) P(gd, ed, lp, k, wd, fb, ft, rh, dd)$$

(Applying the given Bayesian network structure for conditional independence)

$$= \sum_{gd,ed,lp,k,wd,fb,ft,rh,dd} P(hasWon \mid gd, ed, lp) P(lp \mid ft, rh) P(gd \mid k) P(ed \mid k)$$
$$\times P(k \mid fb, dd, wd, ft) P(dd \mid wd) P(fb \mid wd) P(ft \mid rh) P(wd) P(rh)$$

(By summing over all possible values of the hidden variables)

$$= \sum_{gd} \sum_{ed} \sum_{lp} \sum_{k} \sum_{wd} \sum_{fb} \sum_{ft} \sum_{rh} \sum_{dd} P(hasWon \mid gd, ed, lp) P(lp \mid ft, rh) P(gd \mid k) P(ed \mid k)$$
$$\times P(k \mid fb, dd, wd, ft) P(dd \mid wd) P(fb \mid wd) P(ft \mid rh) P(wd) P(rh).$$

(Since hasWon only depends on gd, ed, lp, and we can estimate this probability using MLE)

(the whole equation narrows down to the following)

$$= \sum_{gd} \sum_{ed} \sum_{lp} P(hasWon \mid gd, ed, lp)$$

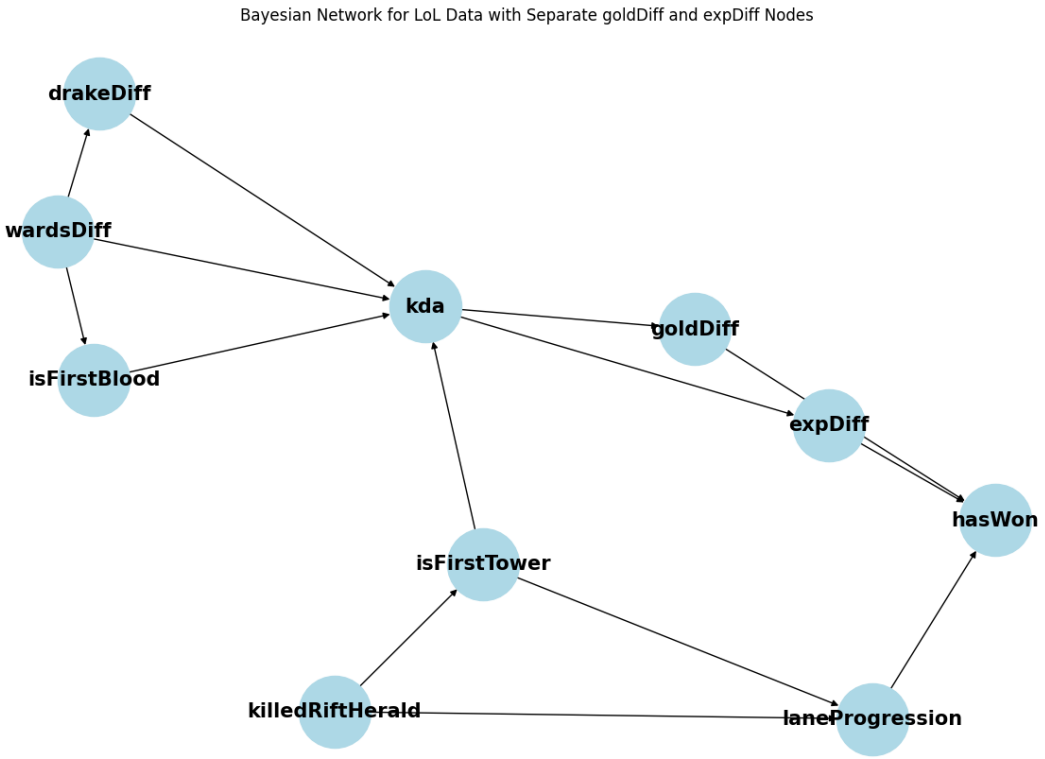

Figure 1: Bayesian Network for lol AI Agent