

# AI Agent for League of Legends Match Prediction Using Bayesian Network

Jason Cheung

February 16, 2025

## Abstract

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

## 1 INTRODUCTION

The probability we're looking for is

$$P(hasWon = 1)$$

. 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)

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

Bayesian Network for LoL Data with Separate goldDiff and expDiff Nodes

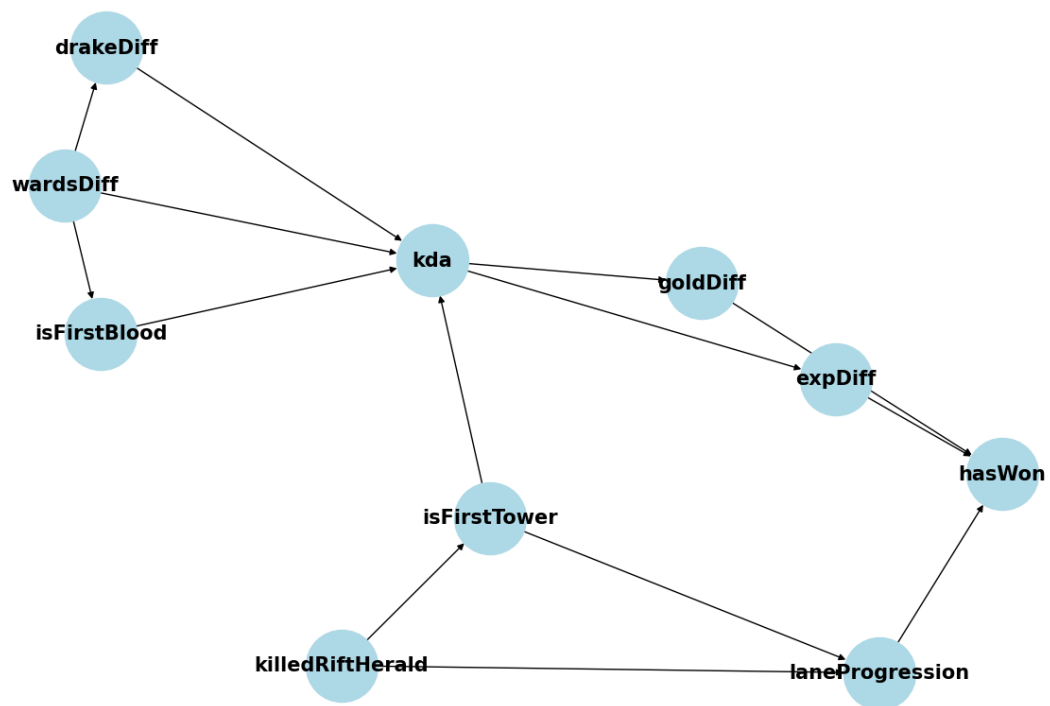


Figure 1: Bayesian Network for lol AI Agent