

---

# Predicting Starcraft 2 Player Skill

---

Forrest Coward

## Abstract

To be written later.

## 1 Overview of Starcraft 2

Starcraft 2 is popular a real time strategy game developed by Blizzard Entertainment. In Starcraft's primary game mode, two players compete against one another in an online match to increase their rating and gain points. A match consists of each player gathering resources and commanding units with the eventual goal of crushing the opposing force. A player wins when all enemy forces have been destroyed.

### 1.1 Leagues

In order to differentiate between the skill of different players, each player is placed into a league based on their performance. There are seven different leagues, each which contain a percent of the active player base:

League	Active Players
Grandmaster	200 players
Master	2%
Diamond	18%
Platinum	20%
Gold	32%
Silver	20%
Bronze	8%

League placement is not fixed. With good performance, a player may be promoted to a higher league. Likewise, continual poor play may warrant a demotion.

### 1.2 Replays

Every Starcraft game produces a .SC2Replay file that exactly specifies the events of a particular game. Example events are a player moving his camera, a player selecting a hotkey, a unit being created, chat messages sent etc. Replays also contain metadata information such as game length and player names. Replays are useful for analyzing strategy, sharing exciting games with others, and most importantly, data analysis.

## 2 Extracting Data From Replay Files

This section explains (1) the process of extracting data from Starcraft replays and (2) where replay files were retrieved to form the data set.

## 2.1 SC2Reader

**SC2Reader**<sup>1</sup> is Python library for parsing Starcraft 2 replay files. It abstracts away the low level details of the file format making it easy to extract game state. Example usage is shown below:

```
replay = sc2reader.parse_replay(replay_file)
players = replay.players
for event in replay.game_events:
    // process event
```

## 2.2 Retrieving League Information

A replay file does not include the leagues of the players at the time the game was played. Instead, a player's current league (as well as their highest league ever achieved) is viewable on their unique Battle.net profile page managed by Blizzard. Unfortunately, Blizzard does not expose an *API* to query for this data. To circumvent this problem, I wrote a web scraper that given a player's profile *URL*, returns the league history of the player.

## 2.3 Gathering Replays

Blizzard recently updated the replay file format to version *2.0.8.25604*. This new format includes a plethora of additional metadata to aid in statistical analysis<sup>2</sup>. Part of this metadata is a polling of certain parts of the game state every ten seconds. This makes it easy to directly compute statistics such as average unspent resources, which was notoriously difficult to determine correctly prior to this update.

Thus, using the latest replay format is practically a requirement for extracting good features from the replays. However, due to the limited time this format has been in use, the vast majority of Starcraft 2 replays on the web cannot be used for this paper.

Luckily, <http://drop.sc>, the most popular Starcraft 2 replay uploading website, adds more than two-hundred *2.0.8.25604* replays a day. Due to a lack of a bulk download option, I wrote a script to crawl [drop.sc](http://drop.sc) and download relevant replays to disk. **The final result was three-thousand replays (every available 2.0.8.25604 replay at the time) that constitutes my data set.**

# 3 Feature Selection

This section overviews which features were selected from each replay and why.

## 3.1 Feature Selection Commentary

In Starcraft 2, players choose to play as one of three races: Terran, Protoss or Zerg. Each race has their own set of structures, units and production cycles. For example, Terran may only build one unit at a time from each production structure, but may queue up to five units simultaneously. On the other hand, Zerg morph all their units from larva, which regularly spawn around a Hatchery. The existence of different units and structures makes it difficult to create features common to all races. We need features that can identify the skill of the player despite the race they play (I do not have enough time to do race by race analysis).

One possibility is to classify units into tiers. Tier 1 units have relatively few prerequisites and thus are always used in the early game. Tier 3 units have more requirements and are generally more expensive, and thus are only seen in the late game. We could hypothesize that better players use higher tier units on average, however, the tier of a unit is not correlated with its effectiveness. For

<sup>1</sup><https://github.com/GraylinKim/sc2reader>

<sup>2</sup>[http://us.battle.net/sc2/en/blog/9669862/StarCraft\\_II\\_Patch\\_208\\_Replay\\_File\\_Enhancements-5\\_7\\_2013](http://us.battle.net/sc2/en/blog/9669862/StarCraft_II_Patch_208_Replay_File_Enhancements-5_7_2013)

example, it is common for Terran players to rely on mass tier 1 and 2 units. Another possibility is to count the number of skill units built. Skill units require extensive amount of control and attention from the player to be effective, thus, we would expect higher skilled players to utilize these units more on average. Unfortunately, this is likely not a great metric. Skill units are too situational depending on the game and matchup. In the end, I decided not to use features involving specific racial units, despite writing code to query this information from a replay.

There are many game mechanics similar among the races such as worker units, resource collection and supply limits. Each race has a unique worker unit that collects the two types of resources in the game: minerals and vespene gas. The amount of workers required and resource collection rates these provide are roughly the same for all three races, making these statistics useful feature constructors. Supply is a limit on the amount of units a player can have on the battle field. Every unit has some supply cost that contributes to a player's total supply. A player must build specific structures/units in order to increase their maximum supply. The amount of time a player is supply blocked (cannot build more units because they are at maximum supply) is an indicator of skill. Good players rarely get supply capped and constantly produce units.

Game duration is another factor that must be considered in relation to certain features. For example, in shorter games, we expect lower resource collection rates and less time spent supply blocked. As the game goes on, an optimal amount of workers are created and resource collection rates reach a maximum. The amount of time a player spends supply blocked will increase as their multitasking skills are stressed. In a very long game, resources become more scarce as players fight to secure the remaining resources on the map. I do not concern myself with game duration under the assumption there will be nice averages with a large number of replays. However, I did throw out games lasting under five minutes. In the first few minutes of the game, there is little a great player can do to differentiate themselves from a lesser foe.

Finally, there are features of the player's actions. The most notable is the actions per minute (APM) metric that measures multitasking ability. Other features in this category include number of hotkeys used and number of times a player issues a command to the minimap.

### 3.2 Selected Features

Feature	Explanation	Value
League	the league the player	1, 2, 3, 4, 5, 6, 7
APM	actions completed per minute	int
Hotkey Usage	number of hotkeys used	int
Minimap	number of times minimap is clicked	int
Mineral Collection Rate	average minerals collected per minute	float
Vespene Collection Rate	average vespene gas collected per minute	float
Unspent Minerals	average unspent minerals per minute	float
Unspent Vespene	average unspent vespene gas per minute	float
Spending Quotient	$35 * (0.00137 * C - \log(U)) + 240$	float
Worker Build Rate	workers built per minute	float
Idle Workers	average % of total workers idle	float 0-100
Time Supply Capped	% of game duration spent supply capped	float 0-100
Unique Units	number of unique units created	int
Well Mannered	true if player sends 'gg' or 'gg wp' chat message	1, 0

Note: in the spending quotient equation,  $C$  is the average resource collection rate and  $U$  is the average unspent resources<sup>3</sup>.

### 3.3 Data points per replay

Each replay produces two points (one for each player), for a total of six-thousand data points.

<sup>3</sup>[http://wiki.teamliquid.net/starcraft2/Spending\\_quotient](http://wiki.teamliquid.net/starcraft2/Spending_quotient)