**Модель**

Данная задача будет решена как задача классификации. В качестве алгоритмов классификации были использованы логистическая регрессия и naïve bayes для сравнения.

В качестве инструмента был использован Python и известная ML библиотека scikit-learn (scikit-learn.org).

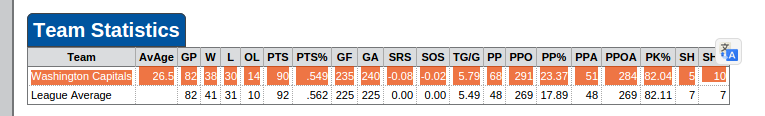
Модель данных:

Для тренировки была использована разница в статистике команд за сезон 2013-2104.

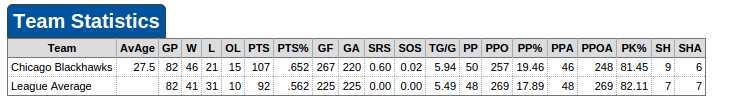
Тоесть например имеем игру Washington Capitals – Chicago Blackhawks:



Статистика Washington Capitals:



Статистика Chicago Blackhawks:



в качестве классов были использован факт от том выйграла первая комада (класс 1) или проиграла (класс 0), тоесть в данном случае класс 0. (первая команда проиграла).

тоесть в итоге поличились данные виде набора float векторов вида:

*x\_train = [ [82-82, 38-46, 30-21, 14-15, 90-107, 0.549-0.652, 235-267, 240-220,*

*-0.08-0.60, -0.02-0.02, 5.79-5.94, 63-50, 291-257, 23.37-19.46, 51-46,*

*284-248, 82.04-81.45, 5-9, 10-6]*

*.*

*.*

*.*

*]*

и вектор класов:

*y\_train = [0, ….. ]*

В качестве тестовых данных были использованы данные за сезон 2014-2015. В итоге кросс валидация на тренировочных данных показала **59%** точность классификации, а валидация на тестовых данных показала **70%** точность.

В заключение могу сказать, что задачу можно было решить через предиктивную модель с помощью линейной регрессии например, тоесть предсказать разницу в счете например. Еще можно было поиграть с нормализацией данных, если для naïve bayes это не нужно, то для логистической регрессии возможно было бы полезно.

**Описание ПО**

Данные были собраны автоматически скриптом “*data/miner.py”.* Тренировочные данные “*data/NHL\_2014\_games.pkl”,* тестовые *“data/test\_NHL\_2015\_games.pkl”,* 25 игр после 2014-12-14 *“data/NHL\_2015\_25games\_from\_2014-12-14.pkl”*.

В скрипте есть необходимиые коментарии чтобы собрать свои данные. Только важно понимать, что скрипт сделан конкретно для сайта [*http://www.hockey-reference.com*](http://www.hockey-reference.com/)*.*

Непосредственнная тренировка происходит в файлах “*src/lr.py”* для логистической регресси и “*src/nb.py”* для naïve bayes. Сначала происходит тренировка на данных сезона 2013-2104, потом кросс-валидация на этих же данных, потом тест на данных 2014-2015(с известным результатом). Затем идет рассчет вероятностей того, что первая команда выйграет для каждой из 25 игр после 2014-12-14.