



الجمهورية العربية السورية
جامعة دمشق
كلية الهندسة المعلوماتية
قسم الذكاء الصناعي

ML Project

التنبؤ بأداء طلاب جامعة دمشق
في حل المسائل البرمجية بلغة ++c
عن طريق التنبؤ بصعوبة الحل
واعتماداً على بيانات طلاب دمشق على موقع ال codeforces

تسليم عجاج
رغد الحلبي

ساره الغدير
راما ريحاوي

إشراف
م. زينة الدلال

1 المقدمة

لدينا مجموعة من المشاريع البرمجية العملية التي يتم اعتماد أداء الطالب فيها لقياس درجة إتقان الطلاب للمفاهيم المطلوبة لحل هذه المشاريع واعتماداً على درجة الإتقان هذه يتم التوصية بمشاريع تناسب مهارة الطالب.

نريد نظام تقييم لقياس درجة الإتقان آنفة الذكر، ولنحقق ذلك نريد طريقة للتنبؤ بصعوبة حل الطالب من أجل المسائل المضافة حديثاً على النظام.

2 الأعمال المشابهة

2.1 Elo-rating المطور في الأصل لتصنيف لاعبي الشطرنج

تعتمد الطريقة على توقع أداء اللاعب تبعاً لمهارته ومهارة اللاعب الخصم، ثم تعديل نقاط اللاعبين تبعاً للأداء الحقيقي، حيث يحصل اللاعب على نقاط إذا كان أداؤه أعلى من المتوقع، ويخسر نقاطاً إذا كان أداؤه أقل من المتوقع.

يمكن استخدام نظام تصنيف Elo في الأنظمة التعليمية إذا فسرنا محاولة الحل على أنها تطابق بين الطالب والعنصر. فإن تكييفها في مجال التعليم يفترض أن المتعلم هو اللاعب، والعنصر هو الخصم. علاوة على ذلك، يتم تمثيل تصنيف Elo للمتعلّم والمحتوى برقم يزيد أو ينقص اعتماداً على المحاولات الناجحة أو الفاشلة لحل تمرين البرمجة.

يوفر استخدام نظام التقييم Elo العديد من المزايا: فهو نظام بسيط يسهل تنفيذه في الأنظمة التعليمية؛ يحتاج عدد قليل من السمات ليتم تعيينه، يمكن استخدامه بسهولة في بيئة عبر الإنترنت؛ كما أنه يوفر أداءً مشابهاً للنماذج الأكثر تعقيداً.

2.2 نظام تقييم قائم على المنطق الترجيحي Fuzzy logic:

يتم توفير علاقة التبعية بين المفاهيم في لغة البرمجة باستخدام خريطة معرفية ضبابية (Fuzzy set). حيث تمثل المجموعات الضبابية والقواعد مستوى معرفة المتعلمين الإلكترونيين وتساعد في تقديم التوصيات المناسبة للمفاهيم السابقة واللاحقة ذات الصلة في الخريطة المعرفية.

2.3 نظام تقييم قائم على نظام خبير expert system:

نموذج طالب مثالي يتم تحديده من قبل خبير ويتم مقارنة أداء الطالب معه، ويتم مقارنة أداء الطالب بشكل عام مع أدائه المعتاد.

3 جمع البيانات

تم الاعتماد على موقع codeforces بشكل أساسي لجمع بيانات حلول طلاب من جامعة دمشق وبيانات المسائل التي تم حلها.

3.1 problemset_df

مجموعة المسائل تم جمعها اعتماداً على ترتيبها التنازلي وفقاً لعدد الحلول (أي تم أخذ الـ 100 مسألة الأكثر حلاً).

3.2 friends_problemset_df

مجموعة المسائل التي قام بحلها أصدقاؤني من مجموعة المسائل السابقة.

3.3 Problem_submissions_source_code_50_df

مجموعة حلول أصدقاؤني على 50 مسألة من مجموعة المسائل السابقة، تم تقسيم هذه الداتا لأربعة أقسام وجمعها على 4 حواسيب لكبر حجمها.

3.4 friends_problems_df

مجموعة البيانات النهائية تتضمن (اسم المسألة، class المسألة، حل الطالب، حل ال tutorial، صعوبة حل ال tutorial، المفاهيم البرمجية المتضمنة بكود الطالب، صعوبة المسألة، تافات المسألة (نوعها)، عدد الطلاب الذين قاموا بحلها).

4 معالجة البيانات وهندسة السمات

1. مجموعة بيانات problemset_df تحتوي على أربعة أعمدة (رمز المسألة، اسم المسألة، الصعوبة، عدد المستخدمين الذين قاموا بحلها)

عمليات التنظيف على هذه الداتا:

- فصل عمود # الذين يعبر عن رمز المسألة إلى عمودين عمود يعبر عن رقم المسألة Number والآخر يعبر عن رمزها Character.
- تنظيف عمود Solved الذي يعبر عن عدد المستخدمين بحذف 'x' منه.
- فصل عمود Name الذين يعبر عن اسم المسألة والتافات الخاصة بها إلى عمودين عمود يعبر عن اسم المسألة Problem Name والآخر يعبر عن التافات Tag.
- إعادة تصنيف عمود التاغ الناتج وتحويله إلى مجموعة أعمدة وفقاً لكل تاغ موجود في العمود الأصلي وإعطاء قيمة 0 أو 1 لكل مسألة وفقاً للتافات الموجودة فيها.
- إعادة تسمية بعض الأعمدة بأسماء جديدة لتناسب فيما بعد مع باقي الداتا فقد تم استبدال اسم عمود Character ب Class وعمود # ب Number_Class
- الاحتفاظ بأول 50 مسألة فقط problemset_50_df لتسهيل جمع باقي البيانات.

2. مجموعة البيانات friends_problemset_df تحتوي على الأعمدة الآتية (رقم الحل، تاريخ ووقت الحل، اسم الشخص الذي قام بالحل، اسم المسألة ومستواها، لغة الحل، نتيجة الحل، مدة الحل، الذاكرة) عمليات التنظيف على هذه الداتا:

- فصل عمود Problem لعمودين عمود يعبر عن مستوى المسألة Number_Class والآخر يعبر عن اسمها Problem_Name وحذف عمود Problem.
- تنظيف عمود Who الذي يعبر عن اسم الشخص الذي حل المسألة من المحارف الزائدة لتجنب التعارضات التي قد تحدث في حال اختلاف الأسماء بأكثر من مكان.
- فلتر المسائل وفقاً لعمودين Language, Verdict أي وفقاً للمسائل المقبولة والمحلوقة بلغة C++.
- دمج الداتا الحالية friends_problemset_d مع problemset_50_df وحفظها في داتا friends_problemset_50_df.

3. مجموعة البيانات Problem_submissions_source_code_50_df:

تم دمج الداتا الحالية مع friends_problemset_50_df وتم تخزينها في friends_submissions_source_code_problemset_50_df

4. final_problem_set:

تحتوي هذه الداتا على الأعمدة التالية (اسم المسألة، class المسألة، حل الطالب، حل ال tutorial، صعوبة حل ال tutorial، المفاهيم البرمجية المتضمنة بكود الطالب، صعوبة المسألة، تاغات المسألة (نوعها)، عدد الطلاب الذين قاموا بحلها، عدد المسائل التي حلها الطالب سابقاً) عمليات المعالجة:

- صعوبة حل ال tutorial تم حسابها عن طريق اكتشاف المفاهيم البرمجية الأساسية في حل الطالب وضربها بنسبة ارتباط المفهوم بالحل (عدد تكرارات المفهوم في الحل، على عدد تكرارات كافة المفاهيم فيه).
- المفاهيم البرمجية المتضمنة بحل الطالب، أيضاً تم اكتشافها باستخدام ال re ثم إعادة تصنيفها لتمثل بعدة أعمدة كل عمود يمثل مفهوم برمجي بقيمة 0 أو 1 تدل على وجود المفهوم بحل الطالب.

5 المراجع

[1] [Sci-Hub | Intelligent e-learning system based on fuzzy logic. Neural Computing and Applications | 10.1007/s00521-019-04087-y](https://doi.org/10.1007/s00521-019-04087-y)