ورقة بحثية مجلة مفتوحة الوصول

خطوتك الأولى نحو التعلم الآلي

د. بشرى معلا * م. محمد عبد الحميد **

* (كلية الهندسة ، جامعة المنارة، البريد الإلكتروني: boushra.maala@manara.edu.sy)

** (كلية الهندسة الميكانيكية والكهربائية، جامعة تشرين، البريد الإلكتروني: mohammedabdlhamed589@gmail.com)

الملخص

يعد التعلم الآلي مجالاً مثيراً وحيوياً يهدف إلى تمكين الأنظمة الحاسوبية من تعلم الأنماط واتخاذ القرارات بشكل ذاتي. يشمل هذا المجال العديد من التقنيات والمفاهيم التي تساهم في تحقيق ذلك مثل الشبكات العصبونية والتعلم العميق. يمتاز التعلم الآلي بتطبيقاته الواسعة في مختلف المجالات. تعد لغة البرمجة Python من أهم اللغات المستخدمة في مجالات مختلفة ومنها مجال التعلم الآلي. يهدف هذا المقال إلى إعطاء مقدمة إلى مجال التعلم الآلي وأهم تطبيقاته كما يتناول المراحل اللازمة لتطبيق خوارزميات التعلم الآلي ويساعد في معرفة المصادر التي يمكن منها تحميل قواعد البيانات اللازمة لعمل الخوارزميات، ويشرح عمليات المعالجة المسبقة على البيانات باستخدام لغة Python قبل تطبيقها على خوارزمية التعلم الآلي.

كلمات مفتاحية _ التعلم الآلي، Python، قاعدة بيانات.

ABSTRACT

Machine learning is an exciting and dynamic field aimed at enabling computer systems to learn patterns and make decisions autonomously. This field includes various technologies and concepts that contribute to achieving this goal, such as neural networks and deep learning. Machine learning is characterized by its wide range of applications in various fields. The Python programming language is one of the most important languages used in various areas, including machine learning. This article aims to provide an introduction to the field of machine learning and its key applications. It also covers the necessary stages for implementing machine learning algorithms and helps in understanding the sources from which the necessary Datasets can be downloaded. It explains the preprocessing steps on the data using the Python language before applying them to machine learning algorithms.

Keywords_ Machine Learning, Python, Dataset

I. مقدمة

يقوم التعلم الآلي على فكرة استخراج المعرفة من البيانات. إنه مجال بحثي عنده تتقاطع الإحصاءات والذكاء الاصطناعي وعلوم الكمبيوتر، ويعرف أيضاً باسم التحليلات التنبؤية أو التعلم الإحصائي. أصبح تطبيق أساليب التعلم الآلي في السنوات الأخيرة في كل مكان في الحياة اليومية، من التوصيات التلقائية للأفلام التي يجب مشاهدتها، إلى الطعام الذي يجب طلبه أو المنتجات التي يجب شراؤها، إلى الراديو المخصص عبر الإنترنت والتعرف على أصدقائك في صورك، تحتوي العديد من مواقع الويب والأجهزة الحديثة على خوارزميات التعلم الآلي في جوهرها. عندما تنظر إلى موقع ويب معقد مثل Facebook أو Amazon أو Netflix، فمن المحتمل جداً أن يحتوي كل جزء من الموقع على نماذج متعددة للتعلم الآلي [1].

خارج التطبيقات التجارية، كان للتعلم الآلي تأثير هائل على الطريقة التي تجرى بها الأبحاث القائمة على البيانات اليوم. يمكن تطبيقها على مشاكل علمية متنوعة مثل فهم النجوم، والعثور على كواكب بعيدة، واكتشاف جزيئات جديدة، وتحليل تسلسل الحمض النووي، وتوفير علاجات السرطان الشخصية.

II. أهمية التعلم الآلي:

في الأيام الأولى للتطبيقات "الذكية"، كان هناك استخدام واسع لنظم قواعد القرار المشفرة يدوياً من أجل معالجة البيانات أو التكيف مع إدخالات المستخدم. على سبيل المثال، يمكن التفكير في نظام تصفية البريد العشوائي الذي يحول الرسائل الإلكترونية الواردة المناسبة إلى مجلد البريد العشوائي. يمكن إنشاء قائمة سوداء تحتوي على الكلمات التي قد تؤدي إلى وضع علامة على رسالة بريد إلكتروني كبريد عشوائي. هذا يمثل مثالاً على استخدام نظام قواعد صمم بواسطة خبراء لتطبيق "ذكي". ومع ذلك، يمكن صياغة قواعد القرار يدوياً في بعض التطبيقات، خاصةً تلك التي يكون فيها لدى البشر فهم جيد لعملية النمذجة. لكن رغم ذلك، يوجد عيبان رئيسيان في استخدام قواعد القرار المشفرة يدوياً لاتخاذ القرارات [2]:

- المنطق المطلوب لاتخاذ قرار خاص بمجال واحد ومهمة واحدة. إن تغيير المهمة ولو قليلاً قد يتطلب إعادة كتابة النظام بأكمله.
- يتطلب تصميم القواعد فهماً عميقاً لكيفية اتخاذ القرار من قبل خبير بشري.

أحد الأمثلة على المكان الذي سيفشل فيه هذا النهج المشفر يدوياً هو اكتشاف الوجوه في الصور. اليوم، يمكن لكل هاتف ذكي اكتشاف وجه في صورة. ومع ذلك، كان اكتشاف الوجه مشكلة لم تحل حتى وقت قريب من عام 2001. المشكلة الرئيسية هي أن الطريقة التي يتم بها "إدراك" وحدات البكسل (التي تشكل صورة في جهاز كمبيوتر) بواسطة الكمبيوتر تختلف اختلافاً كبيراً عن كيفية إدراك البشر للوجه. هذا الاختلاف في التمثيل يجعل من المستحيل بشكل أساسي على الإنسان التوصل إلى مجموعة جيدة من القواعد لوصف ما يشكل وجهاً في صورة رقمية. ومع ذلك، فإن استخدام التعلم الآلي سيجعل الأمر بسيطاً، فإن مجرد تقديم برنامج يحتوي على مجموعة كبيرة من صور الوجوه يكفي الخوارزمية لتحديد الخصائص اللازمة لتحديد الوجه.

III. المشاكل المحلولة من قبل التعلم الآلي

أنجح أنواع خوارزميات التعلم الآلي هي تلك التي تعمل على أتمتة عمليات صنع القرار عن طريق التعميم من الأمثلة المعروفة. في هذا الإعداد، والذي يعرف باسم التعلم تحت الإشراف supervised learning، يزود المستخدم الخوارزمية بأزواج من المدخلات والمخرجات المطلوبة، وتجد الخوارزمية طريقة لإنتاج المخرجات المطلوبة بالنظر إلى المدخلات. على وجه الخصوص، الخوارزمية قادرة على إنشاء مخرج لمدخلات لم ترها من قبل دون أية مساعدة من الإنسان. بالعودة إلى مثالنا عن تصنيف الرسائل الالكترونية غير المرغوب فيها، باستخدام التعلم الآلي، يزود المستخدم الخوارزمية بعدد كبير من رسائل البريد الإلكتروني (المدخلات)، إلى جانب معلومات حول ما إذا كان أيّ من رسائل البريد الإلكتروني هذه غير مرغوب فيها (الإخراج المطلوب). بالنظر إلى رسالة بريد إلكتروني جديدة، ستنتج الخوارزمية بعد ذلك تنبؤاً فيما إذا كان البريد الإلكتروني الجديد غير مرغوب فيه [2].

تسمى خوارزميات التعلم الآلي التي تتعلم من أزواج المدخلات/المخرجات خوارزميات التعلم الخاضعة للإشراف لأن "المعلم" يوفر الإشراف على الخوارزميات في شكل المخرجات المطلوبة لكل مثال يتعلمون منه. في حين أن إنشاء مجموعة بيانات من المدخلات والمخرجات غالباً ما يكون عملية يدوية شاقة، فإن خوارزميات التعلم الخاضعة للإشراف مفهومة جيداً ومن السهل قياس أدائها. إذا كان من الممكن صياغة طلبك كمشكلة تعليمية خاضعة للإشراف، وكنت قادراً على إنشاء مجموعة بيانات تتضمن النتيجة المرجوة، فمن المحتمل أن يكون التعلم الآلي قادراً على حل مشكلتك.

تتضمن أمثلة مهام التعلم الآلي الخاضعة للإشراف ما يلي:

- تحديد الرمز البريدي من الأرقام المكتوبة بخط اليد على مظروف هنا الإدخال هو مسح ضوئي للخط اليدوي، والإخراج المطلوب هو الأرقام الفعلية في الرمز البريدي. لإنشاء مجموعة بيانات لإنشاء نموذج تعلم آلي، تحتاج إلى جمع العديد من المظاريف. ثم يمكنك قراءة الرموز البريدية بنفسك وتخزين الأرقام كنتائج مرغوبة.

– تحديد ما إذا كان الورم حميداً بناء على صورة طبية هنا الإدخال هو الصورة، والإخراج هو ما إذا كان الورم حميداً. لإنشاء مجموعة بيانات لبناء نموذج، تحتاج إلى قاعدة بيانات للصور الطبية. تحتاج أيضاً إلى رأي خبير، لذلك يحتاج الطبيب إلى إلقاء نظرة على جميع الصور وتحديد الأورام الحميدة وأيها ليست كذلك. قد يكون من الضروري إجراء تشخيص إضافي يتجاوز محتوى الصورة لتحديد ما إذا كان الورم الموجود في الصورة سرطانياً أم لا.

– الكشف عن النشاط الاحتيالي في معاملات بطاقات الائتمان هنا الإدخال هو سجل لمعاملة بطاقة الائتمان، والإخراج هو ما إذا كان من المحتمل أن يكون احتيالياً أم لا. على افتراض أنك الكيان الذي يوزع بطاقات الائتمان، فإن جمع مجموعة بيانات يعني تخزين جميع المعاملات وتسجيلها إذا أبلغ المستخدم عن أية معاملة على أنها احتيالية.

شيء مثير للاهتمام يجب ملاحظته حول هذه الأمثلة هو أنه على الرغم من أن المدخلات والمخرجات تبدو واضحة إلى حد ما، إلا أن عملية جمع البيانات لهذه المهام الثلاث تختلف اختلافاً كبيراً. في حين أن قراءة المغلفات شاقة، إلا أنها سهلة ورخيصة. من ناحية أخرى، لا يتطلب الحصول على التصوير الطبي والتشخيصات آلات باهظة الثمن فحسب، بل يتطلب أيضاً معرفة الخبراء النادرة والمكلفة، ناهيك عن المخاوف الأخلاقية وقضايا الخصوصية. في مثال الكشف عن الاحتيال على بطاقات الائتمان، يكون جمع البيانات أبسط بكثير. سيوفر لك عملاؤك الإخراج المطلوب، حيث سيبلغون عن الاحتيال. كل ما عليك فعله للحصول على أزواج الإدخال / الإخراج للنشاط الاحتيالي وغير الاحتيالي هو الانتظار.

الخوارزميات غير الخاضعة للإشراف Unsupervised هي النوع الآخر من الخوارزميات. في التعلم غير الخاضع للإشراف، لا يعرف سوى بيانات الإدخال، ولا تُعطى بيانات إخراج معروفة للخوارزمية. في حين أن هناك العديد من التطبيقات الناجحة لهذه الأساليب، إلا أنه عادة ما يكون من الصعب فهمها وتقييمها.

تتضمن أمثلة التعلم غير الخاضع للإشراف ما يلي:

– تحديد الموضوعات في مجموعة من مشاركات المدونة إذا كان لديك مجموعة كبيرة من البيانات النصية، فقد ترغب في تلخيصها والعثور على السمات السائدة فيها. قد لا تعرف مسبقاً ما هي هذه الموضوعات، أو عدد الموضوعات التي قد تكون هناك. لذلك، لا توجد مخرجات معروفة.

– تقسيم العملاء إلى مجموعات ذات تفضيلات متشابهة نظراً لمجموعة من سجلات العملاء، قد ترغب في تحديد العملاء المتشابهين، وما إذا كانت هناك مجموعات من العملاء لديهم تفضيلات مماثلة. بالنسبة لموقع التسوق، قد يكون هؤلاء "آباء" أو "لاعبين". نظرا لأنك لا تعرف مسبقاً ما هي هذه المجموعات، أو حتى عددها، فليس لديك مخرجات معروفة.

اكتشاف أنماط الوصول غير الطبيعية إلى موقع ويب لتحديد إساءة الاستخدام أو الأخطاء، غالبا ما يكون من المفيد العثور على أنماط وصول مختلفة عن القاعدة. قد يكون كل نمط غير طبيعي مختلفاً تماماً، وقد لا يكون لديك أي حالات مسجلة من السلوك غير الطبيعي. نظراً لأنك في هذا المثال لا تلاحظ سوى الحركية، ولا تعرف ما الذي يشكل سلوكاً طبيعياً وغير طبيعي، فهذه مشكلة غير خاضعة للإشراف.

بالنسبة لكل من مهام التعلم الخاضعة للإشراف وغير الخاضعة للإشراف، من المهم أن يكون لديك تكرار لبيانات الإدخال الخاصة بك التي يمكن للكمبيوتر فهمها. غالبا ما يكون من المفيد التفكير في بياناتك كجدول. كل نقطة بيانات تريد التفكير فيها (كل بريد إلكتروني، كل عميل، كل معاملة) هي صف، وكل خاصية تصف نقطة البيانات هذه (على سبيل المثال، عمر العميل أو مبلغ المعاملة أو موقعها) هي عمود. يمكنك وصف المستخدمين حسب عمرهم وجنسهم ومتى أنشأوا حساباً وعدد المرات التي اشتروا فيها من متجرك عبر الإنترنت. يمكنك وصف صورة الورم من خلال قيم التدرج الرمادي لكل بكسل، أو ربما باستخدام حجم الورم وشكله ولونه.

يعرف كل كيان أو **صف** هنا باسم **عينة** (أو نقطة بيانات) في التعلم الآلي، في حين تسمى **الأعمدة** أي الخصائص التي تصف هذه الكيانات **بالميزات**.

IV. تطبيقات التعلم الآلي ML:

ان ML هي الكلمة الأكثر شيوعاً في تكنولوجيا اليوم وتنمو بشكل متسارع يوماً بعد يوم. ونحن نستخدم ML بشكل يومي مثل خرائط غوغل Google Maps ومساعد غوغل Google Assistant.

وفيما يلي أكثر تطبيقات ML شيوعاً [1]:

1. تمييز الصور Image Recognition

من أشهر تطبيقات الذكاء الصنعي. تستخدم لتعريف الأشياء، الأشخاص والأماكن والصور الرقمية. الاستخدام الأشهر لتمييز الصور واكتشاف الوجوه هو اقتراحات الإشارة الآلية للأصدقاء Automatic friend tagging suggestion.

يؤمن Facebook لنا خاصية المشاركة مع الأصدقاء بمجرد تحميل صورة وعندها تلقائيا يوجد اقتراح مشاركة مع الاسم والتقنية وراء ذلك هي خوارزمية التعلم الآلي لكشف وتمييز الوجوه machine learning's face detection and recognition algorithm. وهي مرتكزة على DeepFace والتي تعد مسؤولة عن تمييز الوجه وهوية الشخص في الصورة.

2. تمييز الكلام Speech Recognition:

عند استخدام Google فإنه يوجد خيار البحث عن طريق الصوت Search by voice والتي تأتي تحت مجال تمييز الكلام speech recognition وهي إحدى تطبيقات ML. تمييز الكلام هو عملية تحويل التعليمات الصوتية إلى نص، ومعروفة أيضاً باسم Speech to text أو Computer speech recognition. إن خوارزميات ML لتمييز الصوت مستخدمة في العديد من التطبيقات مثل Google Assistant و Siri وغيرها.

3. التنبؤ بالحركية Traffic prediction:

في حال كنا نريد زيارة مكان جديد، نلجأ لمساعدة خرائط غوغل Google Maps، والتي تظهر لنا المسار الصحيح مع المسار الأقصر وتتنبأ بشروط الحركية Traffic.

تتنبأ بالشروط Traffic condition مثل الطقس والحركة البطيئة أو الازدحام العالي بمساعدة طريقتين:

- Real Time location وهو الموقع الخاص بالعربة في الزمن الحقيقي بمساعدة تطبيق Google Map والحساسات.
- Average time وتمثل الوقت الوسطي الذي احتاجه في الأيام السابقة بنفس الوقت.

كل شخص يستخدم تطبيق Google Map يساعد التطبيق أن يصبح أفضل حيث أنه يأخذ معلومات من المستخدم ويرسل بيانات بشكل عكسي لتحسين الأداء.

4. المنتجات المطلوبة Product recommendations

تستخدم ML بشكل واسع في الشركات الاقتصادية والشركات الترفيهية مثل Amazon و Netflix من أجل عرض المنتجات للمستخدم. حيث عند البحث في Amazon فإنه يبدأ الحصول على إعلانات عن نفس المنتج بينما يتصفح النت في نفس المتصفح وذلك بسبب ML. تفهم Google اهتمامات المستخدم من خلال عدة خوارزميات ML وتقترح المنتجات بناء على اهتمامات المستخدم.

بشكل مشابه Netflix فإننا نجد نفس الاقتراحات من أجل الأفلام والمسلسلات وذلك أيضاً بمساعدة ML.

5. السيارات ذاتية القيادة Self-driving cars:

واحدة من أشهر تطبيقات ML هي السيارات ذاتية القيادة. حيث تلعب ML دوراً ملحوظاً فيها. إن شركة التصنيع الخاصة بالسيارات الشهيرة Tesla تعمل على السيارات ذاتية القيادة. وتستخدم التعلم دون إشراف لتدريب نماذج السيارات على اكتشاف الأشخاص والأغراض أثناء القيادة.

6. فرز الرسائل الالكترونية المزيفة:
Email Spam and Malware Filtering

عند استقبال رسالة الكترونية جديدة فإنه يمكن تحديد نوعها مباشرة إما إلى هامة important أو طبيعية normal أو مزيفة spam. يتم استقبال الرسائل الالكترونية الهامة ضمن inbox بينما يتم استقبال رسائل spam ضمن صندوق spam والتقنية وراء ذلك هي ML.

إن بعض خوارزميات تعلم الآلة مثل Multi-Layer perceptron و Decision Tree و Naïve Bayes classifier تستخدم في هذه الميزة.

7. التشخيص الطبي Medical Diagnosis:

تستخدم ML لتحليل الأمراض، ومع ذلك فإن التقنيات الطبية تتطور بشكل متسارع، ويمكن بناء نماذج ثلاثية الأبعاد (3D) التي يمكنها التنبؤ بالموقع الصحيح للآفات في الدماغ، وتساعد في إيجاد الأورام الدماغية والأمراض الأخرى المرتبطة بالدماغ بكل سهولة.

8. الترجمة الآلية للغات Automatic Language Translation:

في هذه الأيام، في حال زرنا مكاناً جديداً ولا نعلم اللغة التي يتحدث بها الناس فإنها ليست مشكلة لأن ML تساعدنا في تحويل النص إلى اللغة المطلوبة. Google's GNMT (Google Neural Machine Translation) يؤمن هذه الخاصية وهي آلة عصبونية تترجم النص إلى اللغة المطلوبة وتدعى بالترجمة الآلية.

التقنية وراء ذلك هي sequence to sequence learning. والتي تستخدم في تمييز الصور وترجمة النصوص من لغة إلى أخرى.

V. دورة حياة التعلم الآلي:

أعطت ML أنظمة الحاسوب القدرة على التعلم دون أن تُبرمَج بشكل صريح. ولكن كيف يعمل نظام ML؟

يمكن أن توصف دورة حياة ML بأنها عملية دائرية لبناء مشروع ML بكفاءة والغرض الرئيسي هو إيجاد الحل لمشكلة أو مشروع. إن الشيء الأهم في العملية كاملةً هي فهم المشكلة ومعرفة الغرض من المشكلة، لذلك قبل البدء بدورة الحياة يجب أن نفهم المشكلة لأن النتائج الجدية تعتمد على الفهم الأفضل للمشكلة. في كامل الدورة، لحل مشكلة يُنشَأ نموذج model من خلال التدريب Training ولكن لتدريب نموذج نحتاج بيانات Data ولذلك تبدأ العملية بتجميع البيانات [1].

1. تجميع البيانات Gathering Data:

إن عملية تجميع البيانات هي المرحلة الأولى من دورة حياة ML. والغرض من هذه المرحلة هي التعريف والحصول على البيانات المرتبطة بالمشكلة. في هذه الخطوة، نحتاج لتعريف مصادر البيانات المختلفة حيث يمكن الحصول على البيانات من مصادر مختلفة مثل internet, Database, Files أو mobile Devices. وهي واحدة من أهم الخطوات.

إن الكمية والنوعية للبيانات التي تُجمَع سوف تحدد كفاءة الخرج. وكلما كانت البيانات أكبر كلما كانت التنبؤ في الخرج أدق.

تتضمن هذه المرحلة المهام الآتية:

- تعريف مصادر البيانات المختلفة.
- تجميع البيانات.
- مكاملة البيانات التي تم الحصول عليها من مصادر مختلفة.

من خلال القيام بالمهمة السابقة نحصل على مجموعة مترابطة من البيانات تدعى قاعدة البيانات سوف تستخدم في خطوات لاحقة.

2. تحضير البيانات Data Preparation:

بعد تجميع البيانات، يجب تحضيرها لخطوات أبعد. عملية تحضير البيانات هي الخطوة التي فيها نضع بياناتنا في مكان مناسب لتحضيرها لاستخدامها في نموذج التدريب. في هذه المرحلة تُجمَع البيانات سوية، ومن ثم يُعاد ترتيبها بشكل عشوائي. يمكن تقسيم هذه المرحلة إلى عمليتين:

- Data exploration: تستخدم لفهم طبيعة البيانات التي سيتم العمل معها. حيث يجب فهم الخصائص والصيغ وجودة البيانات، إن الفهم الأفضل للبيانات يقود إلى خرج أكثر كفاءة. وفي هذه المرحلة نوجد الترابط والاتجاهات العامة.
- Processing_Data pre: تكون الخطوة التالية هي بالمعالجة المسبقة للبيانات من أجل التحليل.

3. شحن البيانات Data Wrangling

هي عملية تنظيف وتحويل البيانات السطرية إلى صيغة قابلة للاستخدام. نختار المتغيرات التي سنستخدمها وتحول البيانات إلى الصيغة الصحيحة لجعلها مناسبة للتحليل في الخطوة التالية. تعد واحدة من أهم الخطوات في كامل العملية.

ليس من الضروري أن تكون جميع البيانات المجمعة مفيدة للاستخدام فبعضها قد يكون غير مفيد. في تطبيقات العالم الحقيقي قد تمتلك البيانات المجمعة قضايا مختلفة مثل:

- قيم مفقودة Missing Values
- قيم مكررة Duplicate Value
- قيم غير صحيحة Invalid data

لذلك تستخدم تقنيات فرز لتنظيف البيانات، ويعتبر أمراً الزامياً اكتشاف وإزالة القضايا السابقة لأنها قد تؤثر سلباً على جودة الخرج.

4. تحليل البيانات Data Analysis:

تمرر البيانات المحضرة والمنظفة إلى مرحلة التحليل وتتضمن الخطوات الآتية:

1. اختيار تقنية التحليل
2. بناء النموذج
3. مراجعة النتائج

الغرض من هذه المرحلة هي بناء نموذج ML لتحليل البيانات باستخدام تقنيات تحليل ومراجعة الخرج. تبدأ بتحديد نمط المشكلة حيث نختار تقنية ML مثل التصنيف Classification أو الانحدار Regression أو تحليل الترابط والعناقيد Cluster analysis Association... ومن ثم نبني النموذج باستخدام البيانات المجهزة ونقيم النموذج.

الغاية من هذه المرحلة أخذ البيانات واستخدام خوارزميات ML لبناء النموذج.

5. تدريب النموذج Train Model

في هذه المرحلة يُدرَّب النموذج لتطوير الأداء من أجل خرج أفضل للمشكلة. تُستخدم قاعدة البيانات Dataset لتدريب النموذج باستخدام خوارزميات ML. بالنتيجة يمكنها فهم أنماط مختلفة وقواعد وخصائص.

6. اختبار النموذج Test Model

عند تدريب نموذج ML على قاعدة بيانات معطاة، عندها يختبر النموذج وفي هذه المرحلة نفحص دقة النموذج من خلال تزويده ببيانات. واختبار النموذج يحدد الدقة كنسبة مئوية للنموذج من أجل متطلبات المشروع أو المشكلة.

7. التطبيق Deployment

الخطوة الأخيرة من دورة ML هي التطبيق، إذ يطبق النموذج على نظام حقيقي. في حال كان النموذج المحضر يعطي نتائج دقيقة موافقة للمتطلبات بسرعة مقبولة يمكن عندها تطبيق النموذج في العالم الحقيقي. ولكن قبل تطبيق المشروع يُفحص في حال كان يطور أداءه باستخدام البيانات المتاحة أم لا. إن تطور التطبيق مشابه لعملية وضع التقرير النهائي للمشروع.

VI. كيفية الحصول على قاعدة البيانات:

إن المفتاح الرئيسي للنجاح في ML أو أن يصبح الشخص قادراً على دراسة علوم البيانات هو من خلال التدرب على قواعد البيانات مختلفة. ولكن اكتشاف قاعدة بيانات مناسبة لكل مشروع ML يعتبر مهمة صعبة.

تمثل قاعدة البيانات تجميع من البيانات والتي تكون فيها البيانات مرتبة بترتيب معين. يمكن أن تحوي معلومات كسلسلة من جدول مصفوفة لقاعدة بيانات. ويظهر الجدول 1 مثال عن قاعدة بيانات

Country	Age	Salary	Purchased
India	38	48000	No
France	43	45000	Yes
Germany	30	54000	No
France	48	65000	No
Germany	40		Yes
India	35	58000	Yes

الجدول 1 . مثال عن قاعدة بيانات

إن قاعدة البيانات المجدولة يمكن أن تفهم كمصفوفة أو جدول قاعدة البيانات، حيث أن كل عمود يمثل متغيراً محدداً وكل سطر يمثل الحقل في قاعدة بيانات. إن النمط الأكثر دعماً لقاعدة البيانات المجدولة هو CSV، ولكن للتخزين بصيغة شجرية يمكن استخدام JSON لكفاءة أعلى.

يوجد عدة أنواع للبيانات

- بيانات رقمية Numerical data: مثل سعر المنزل ودرجة الحرارة .....
- بيانات فئوية Categorical data: مثل Yes/No أو True/False أو Blue/Green ....

تكون قاعدة البيانات في العالم الحقيقي بأحجام هائلة. والتي يعتبر من الصعب ادارتها والتعامل معها. لذلك يمكن التدرب على خوارزميات ML من خلال dummy dataset.

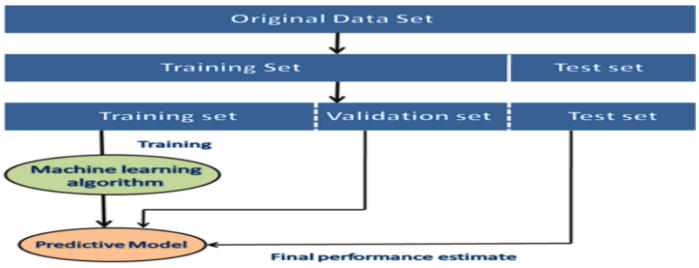
1. الحاجة لقاعدة البيانات:

للعمل مع مشاريع ML نحتاج كمية كبيرة من المعلومات لأنه دون هذه المعلومات فانه لا يمكن تدريب نموذج ML/AI. إن عملية تجميع وتحضير قاعدة البيانات هي واحدة من الأجزاء الهامة عند انشاء مشروع ML/AI.

لا يمكن للتقنية المطبقة وراء مشاريع ML أن تعمل بشكل صحيح في حال كانت قاعدة البيانات غير مجهزة ومعالجة بشكل صحيح.

خلال عملية التطوير لمشروع ML، فإن المطور يعتمد بشكل كامل على قواعد البيانات وفي بناء تطبيقات ML تقسم قواعد البيانات الى جزأين كما هو موضح في الشكل 1 :

- مجموعة التدريب Training
- مجموعة الاختبار Test

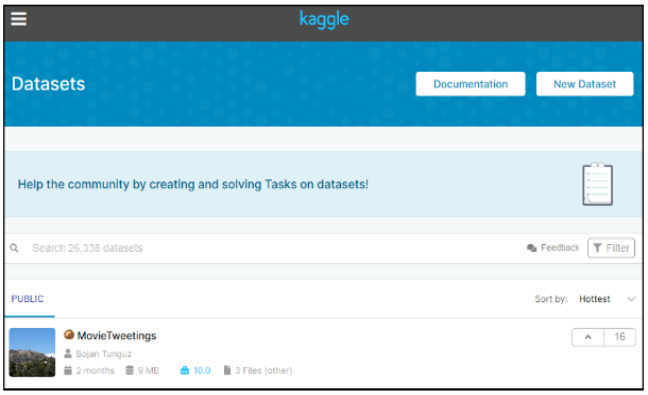


الشكل 2 . تقسيم قاعدة البيانات

يوجد عدة مصادر لقواعد البيانات أهمها:

❖ **قاعدة بيانات Kaggle:**

تعد من أهم المصادر لتأمين قواعد البيانات [3] خصوصاً لعلماء البيانات والمتدربين في ML. تسمح للمستخدم بإيجاد وتحميل ونشر قواعد البيانات بطريقة سهلة. تؤمن الفرصة للعمل مع مهندسي ML آخرين لحل مشاكل البيانات.



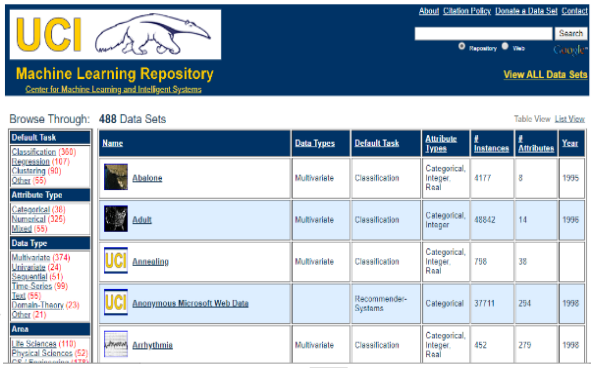
الشكل 2 . واجهة موقع Kaggle

تؤمن قاعدة البيانات بدقة عالية وبصيغ مختلفة والتي يمكن بسهولة إيجادها وتحميلها.

❖ **UCI Machine Learning Repository:**

تعتبر من أهم مصادر قواعد البيانات [4] حيث تحوي قواعد بيانات ونظريات ومولدات بيانات، إنها تستخدم بشكل واسع من

قبل مجتمع ML من أجل تحليل خوارزميات ML. استخدمت منذ عام 1987 بشكل واسع من قبل الطلاب والدكاترة والباحثين كمصدر أول لقواعد البيانات الخاصة ب ML.

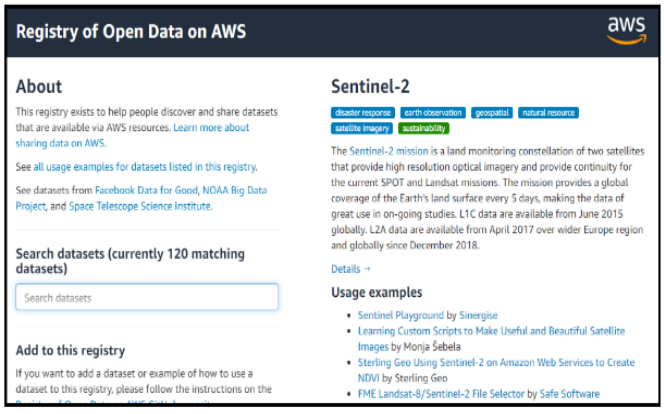


الشكل 3 . واجهة موقع UCI

تصنف قاعدة البيانات بحسب المشكلة والمهمة مثل Regression و Classification و Clustering....، تحوي أيضاً بعض قواعد البيانات الشائعة مثل Iris Datset و Car Evaluation Dataset و Poker Hand Dataset.

❖ قاعدة البيانات AWS:

يمكن البحث والتحميل والنفاذ ومشاركة قاعدة البيانات والتي تكون متاحة بشكل عام من قبل AWS. يمكن الوصول لـ قاعدة البيانات هذه من خلال مصادر AWS ولكن تزود وتصان من قبل منظمات حكومية وباحثين وأعمال أو شخصيات [5].

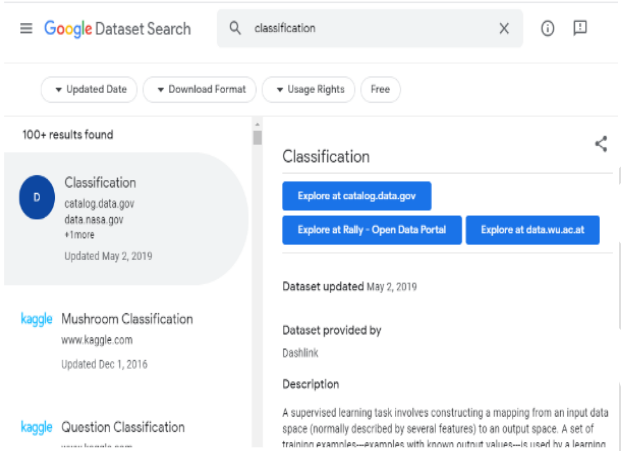


الشكل 4 . موقع AWS لقواعد البيانات

يمكن لأي شخص أن يحلل ويبني خدمات متنوعة باستخدام البيانات المشاركة من خلال AWS. قد تساعد قاعدة البيانات المشاركة على cloud المستخدمين في قضاء زمن أكبر في تحليل البيانات بدلاً من استيعاب البيانات.

❖ Google's Search Engine Datasets:

هو محرك بحث أطلق من قبل شركة Google عام 2018 [6]، إنه يساعد الباحثين للحصول على قاعدة البيانات بشكل مجاني ومتاحة للاستخدام.



الشكل 5. محرك بحث Google لقواعد البيانات

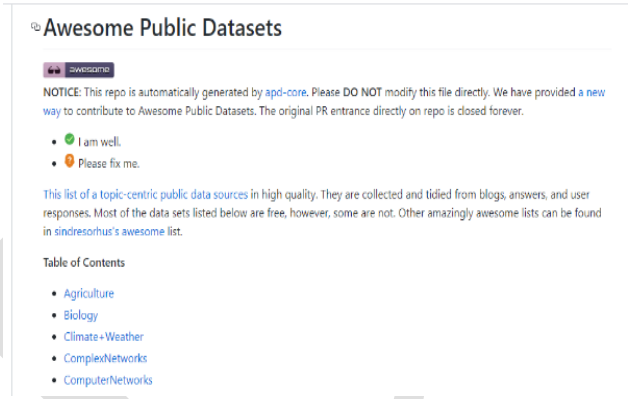
❖ Microsoft Datasets:

يحوي مجموعة من قواعد البيانات في مجالات مختلفة من NLP و Computer vision والمجال الخاص بالعلوم [7].

الشكل 6. محرك بحث Microsoft لقواعد البيانات

❖ **Awesome Public Datasets Collection:**

تؤمن قواعد البيانات بدقة عالية [8]، كما أنها مرتبة بطريقة معينة ضمن قائمة بحث عن الموضوع مثل المناخ، الشبكات المعقدة، ومعظمها متاحة بشكل مجاني والبعض الآخر ليس مجانياً.



الشكل 7. مجموعة Awesome Publication لقواعد البيانات

❖ **Government Datasets:**

يوجد مصادر مختلفة للحصول على البيانات المرتبطة بالحكومات. حيث تنشر بلدان متعددة بيانات حكومية للاستخدام العام بعد أن تجمع من مصادر مختلفة.

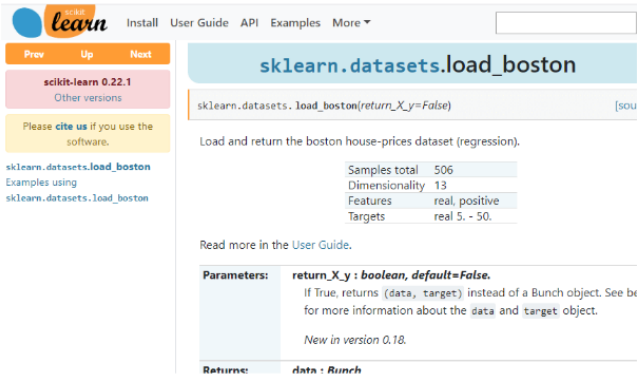
الهدف من تأمين هذه القواعد هو زيادة الشفافية في العمل بين الناس لاستخدام البيانات. وفيما يلي بعض الروابط للبيانات الخاصة بالحكومات.

❖ **Computer Vision Datasets:**

تؤمن مجموعة كبيرة من قواعد البيانات، وهي تستخدم في computer vision مثل تصنيف الصور والفيديو وتقطيع الصور. لذلك في حال أردنا بناء مشروع DL أو Image processing يمكن الاستعانة بها [9].

❖ **Scikit-learn Dataset:**

تعد مصدراً عظيماً للمتحمسين في مجال ML. ويؤمن قواعد البيانات للعالم الحقيقي والتجريب. ويمكن الحصول عليها من خلال الحزمة sklearn وباستخدام API [10].



الشكل 8. حزمة sklearn لقواعد البيانات

VII. المعالجة المسبقة للبيانات في التعلم الآلي

هي عملية تحضير البيانات السطرية وجعلها مناسبة من أجل نموذج ML. تمثل المرحلة الأولى والأهم أثناء إنشاء نموذج ML. عند إنشاء المشروع ليس من الضروري الحصول على بيانات نظيفة ومصاغة clean and formatted مباشرة. لذا أثناء القيام بأية عملية على البيانات فإنه من الضروري تنظيفها ووضعها في طريقة formatted. لذلك تستخدم مهمة المعالجة المسبقة.

تحتوي البيانات الحقيقية عادة ضجيج، قيم مفقودة وقد تحوي بيانات لا يمكن استخدامها مباشرة من قبل نموذج التعلم.

تعتبر عملية المعالجة المسبقة للبيانات مهمة لتنظيف البيانات وجعلها مناسبة للنموذج والذي بدوره يزيد الدقة والكفاءة لنموذج ML. وتتضمن المراحل الآتية:

- الحصول على البيانات Getting the Dataset
- إدراج المكتبات Importing Libraries.
- إدراج البيانات Importing Dataset
- إيجاد البيانات المفقودة Finding Missing Data
- ترميز البيانات الفئوية Encoding categorical data
- فصل البيانات إلى بيانات تدريب وبيانات اختبار Splitting Dataset into training and test set
- معايرة السمات Feature scaling

1. الحصول على البيانات

نحتاج بشكل أساسي لإنشاء نموذج ML إلى قاعدة البيانات، حيث أن نموذج ML يعمل بشكل كامل على البيانات.

قد تكون قاعدة البيانات بأشكال مختلفة لأغراض مختلفة مثلاً في حال أردنا إنشاء نموذج تعلم لغرض العمل فإن قاعدة البيانات ستكون مختلفة عن المطلوبة من أجل ملازمة مريض. لذلك فإن كل قاعدة البيانات تختلف عن الأخرى. لاستخدام قاعدة البيانات في البرنامج فإنها توضع عادة في ملف CSV وأحيانا تستخدم ملفات HTML أو xlsx.

تعد ملفات csv اختصار لـ Comma-separated Values. حيث هي صيغة ملف والتي تسمح لنا بحفظ جداول البيانات. وتعتبر مناسبة لقاعدة البيانات الضخمة واستخدامها في البرنامج.

يمكن تجميع البيانات من مصادر مختلفة وحفظها في ملف بامتداد csv. في هذا المقال سنستخدم قاعدة بيانات موجودة ضمن الموقع [11].

2. إدراج المكتبات:

من أجل القيام بالمعالجة المسبقة للبيانات باستخدام Python، نحتاج لإدراج بعض المكتبات المعرفة مسبقاً. هذه المكتبات تستخدم للقيام بمهام محددة. ويوجد 3 مكتبات أساسية سنستخدمها للمعالجة وهي

- Numpy: المكتبة المسؤولة عن التعامل مع المصفوفات
- Matplotlib: المكتبة المسؤولة عن رسم المخططات البيانية
- Pandas: المكتبة المسؤولة عن التعامل مع نماذج البيانات

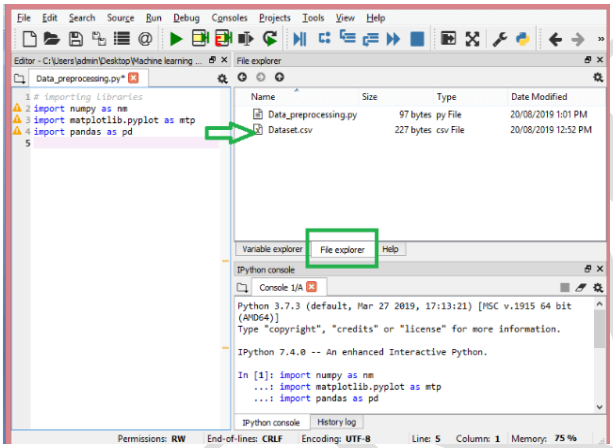
```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

الشكل 9. أدراج المكتبات المستخدمة

3. إدراج قاعدة البيانات:

نحتاج لإدراج قاعدة البيانات المجمعة من أجل المشروع الخاص بنا. ولكن قبل القيام بذلك يجب ضبط المسار الصحيح. لضبط المسار باستخدام Spyder، نقوم بالخطوات الآتية:

- حفظ ملف بايثون في المسار الذي يحوي قاعدة البيانات.
- الذهاب إلى File في Spyder واختيار المسار المطلوب.
- ضغط F5 لتنفيذ الملف.



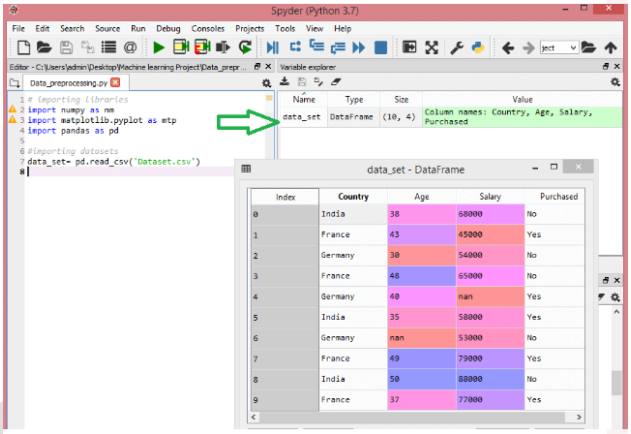
الشكل 10. آلية تشغيل الملف

لإدراج قاعدة البيانات نستخدم التابع read_csv من مكتبة pandas والذي يستخدم لقراءة ملف csv والقيام بعمليات محددة عليه. وباستخدامه يمكن قراءة الملف بشكل محلي من خلال الامر التالي:

```
data_set = pd.read_csv('Data.csv')
```

حيث data_set هي اسم متغير لتخزين قاعدة البيانات الخاصة بنا وضمن هذا التابع يجب أن نمرر اسم قاعدة البيانات. وعند تنفيذ هذا السطر من الكود فإنه سيتم إدراج قاعدة البيانات ضمن الكود ويمكن فحصه من خلال الخيار variable explorer ومن ثم الضغط على data_set كما في الشكل:

تبدأ الفهرسة من 0 ويمكن تغيير الـ format من خلال الخيار format option.



الشكل 11. قراءة البيانات باستخدام python

في ML من المهم جداً تمييز مصفوفة الخواص (المتغيرات المستقلة) والبيانات غير المستقلة من قاعدة البيانات. حيث تعبر المتغيرات المستقلة عن البيانات التي تمثل الدخل بينما تعبر البيانات غير المستقلة عن الخرج. في قاعدة البيانات المستخدمة هنا، يوجد ثلاث متغيرات مستقلة هي country وage وsalary ومتغير واحد غير مستقل هو purchased.

لاستخلاص البيانات المستقلة نستخدم الطريقة iloc[ ] من المكتبة Pandas حيث تستخدم لاستخلاص الأسطر والأعمدة من قاعدة البيانات

x = data_set.iloc[:,:−1].values

في التعليمة السابقة تستخدم: لأخذ جميع الأسطر و: الثانية لأخذ جميع الأعمدة وهنا 1−: لأننا لا نريد أخذ العمود الأخير حيث أنه يحوي بيانات غير مستقلة ومن خلال القيام بذلك سنحصل على مصفوفة الخصائص.

	0	1	2
0	France	44.0	72000.0
1	Spain	27.0	48000.0
2	Germany	30.0	54000.0
3	Spain	38.0	61000.0
4	Germany	40.0	nan
5	France	35.0	58000.0
6	Spain	nan	52000.0
7	France	48.0	79000.0
8	Germany	50.0	83000.0
9	France	37.0	67000.0

الشكل 12. البيانات المستقلة ضمن البيانات

وسيكون الخرج هو

```
[['France' 44.0 72000.0]
 ['Spain' 27.0 48000.0]
 ['Germany' 30.0 54000.0]
 ['Spain' 38.0 61000.0]
 ['Germany' 40.0 nan]
 ['France' 35.0 58000.0]
 ['Spain' nan 52000.0]
 ['France' 48.0 79000.0]
 ['Germany' 50.0 83000.0]
 ['France' 37.0 67000.0]]
```

الشكل 13. قيم الميزات في خرج البرنامج

لاستخلاص البيانات غير المستقلة نستخدم الطريقة iloc[ ] من المكتبة Pandas حيث تستخدم لاستخلاص الأسطر والأعمدة من قاعدة البيانات

y= data_set.iloc[:,3].values

هنا أُخذت جميع الأسطر مع العمود الأخير فقط وسنحصل على مصفوفة العناصر غير المستقلة

	0
0	No
1	Yes
2	No
3	No
4	Yes
5	Yes
6	No
7	Yes
8	No
9	Yes

الشكل 14. البيانات غير المستقلة ضمن البرنامج

ونحصل على

```
array(['No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes'],
      dtype=object)
```

الشكل 15. قيم الهدف في خرج البرنامج

4. معالجة البيانات المفقودة:

الخطوة التالية من المعالجة المسبقة للبيانات هي التعامل مع البيانات المفقودة في قاعدة البيانات. تحوي قاعدة البيانات الخاصة بنا بعض البيانات المفقودة، هذا قد يتسبب بمشكلة كبيرة للنموذج. لذلك من الضروري التعامل مع البيانات المفقودة. يوجد طريقتان للتعامل مع البيانات المفقودة وهي:

- من خلال حذف السطر المحدد:

تستخدم من أجل التعامل مع قيمة Null، إذ نحذف السطر أو العمود الذي يتألف من قيم Null، ولكن هذه الطريقة لا تعد ذات كفاءة لأن إزالة البيانات قد تؤدي لفقد بالمعلومات بالنتيجة لن نحصل على دقة في الخرج.

- حساب الوسيط:

في هذه الطريقة، يتم حساب الوسيط للعمود أو السطر الذي يحوي قيماً مفقودة، ويتم وضعها مكان القيمة المفقودة. تعتبر هذه الاستراتيجية مفيدة للخصائص التي تملك قيم عددية مثل age،salary، year ..... للتعامل مع القيم المفقودة، نستخدم مكتبة scikit-learn والتي تحوي عدة مكتبات لبناء نموذج ML. ونستخدم الصنف Imputer من المكتبة sklearn .preprocessing حيث يتم تعريف غرض من الصنف SimpleImputer ويأخذ بارامترين الأول هو نوع القيم المفقودة (nan) والثاني آلية التعامل معها أي mean من خلال الوسيط ثم باستخدام التابع transform يتم تطبيق العملية على القيم

```
#Taking care of missing data
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(X[:, 1:3])
X[:, 1:3] = imputer.transform(X[:, 1:3])
print(X)
```

الشكل 16. الكود اللازم لمعالجة البيانات المفقودة

ويكون الخرج في حال طباعة مصفوفة الميزات X:

```
[['France' 44.0 72000.0]
 ['Spain' 27.0 48000.0]
 ['Germany' 30.0 54000.0]
 ['Spain' 38.0 61000.0]
 ['Germany' 40.0 63777.77777777778]
 ['France' 35.0 58000.0]
 ['Spain' 38.77777777777778 52000.0]
 ['France' 48.0 79000.0]
 ['Germany' 50.0 83000.0]
 ['France' 37.0 67000.0]]
```

الشكل 17. قيم الميزات في خرج البرنامج بعد تعديل البيانات المفقودة

وتصبح مصفوفة الميزات X:

	0	1	2
0	France	44.0	72000.0
1	Spain	27.0	48000.0
2	Germany	30.0	54000.0
3	Spain	38.0	61000.0
4	Germany	40.0	63777.7777777…
5	France	35.0	58000.0
6	Spain	38.7777777777…	52000.0
7	France	48.0	79000.0
8	Germany	50.0	83000.0
9	France	37.0	67000.0

الشكل 18. قيم الميزات في برنامج Spyder بعد التعديل

حيث نلاحظ أن قيم Nan تم استبدالها بالوسيط لعناصر العمود

5. ترميز البيانات النصية:

إن البيانات الفئوية categorical data هي البيانات غير الرقمية، لدينا هنا country وpurchased. وبما أن نموذج ML يتعامل مع الرياضيات والأرقام فإنه يجب ترميز هذه البيانات بأرقام.

من أجل المتغير Country: سوف نقوم بتحويل المتغير إلى بيانات رقمية وللقيام بذلك نستخدم الصنف LabelEncoder( ) من المكتبة preprocessing.

```
# Encoding categorical data
# Encoding the Independent Variable
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
print(X)
```

الشكل 19. الكود اللازم لترميز البيانات الفئوية

في التعليمات السابقة أدرج الصنف LabelEncoder من المكتبة sklearn. والذي يقوم بترميز المتغيرات الفئوية إلى أرقام. ولكن في حالتنا يوجد ثلاث متغيرات للـ country لذلك رمزت بـ 0 و1 و2. ومن خلال هذه القيم فإن النموذج سيفرض أنه يوجد ارتباط بين هذه المتغيرات والذي بدوره سوف ينتج خرج خاطئ وللتخلص من هذه المشكلة سوف نستخدم dummy encoding.

المتغيرات dummy variables هي المتغيرات التي تأخذ قيمة 1 أو 0. حيث أن القيمة 1 تعطي وجود المتغير ضمن عمود محدد وبقية المتغيرات تكون 0، وسوف نحصل على عدد من الأعمدة مساو لعدد الفئات.

في قاعدة البيانات الخاصة بنا يوجد 3 حالات للقيمة الفئوية لذلك سوف ينتج 3 أعمدة تملك قيم 0 و1. من أجل Dummy Encoding سوف نستخدم صنف OneHotEncoder من مكتبة preprocessing.

والذي يعطي الخرج:

```
[[1.0 0.0 0.0 44.0 72000.0]
 [0.0 0.0 1.0 27.0 48000.0]
 [0.0 1.0 0.0 30.0 54000.0]
 [0.0 0.0 1.0 38.0 61000.0]
 [0.0 1.0 0.0 40.0 63777.77777777778]
 [1.0 0.0 0.0 35.0 58000.0]
 [0.0 0.0 1.0 38.77777777777778 52000.0]
 [1.0 0.0 0.0 48.0 79000.0]
 [0.0 1.0 0.0 50.0 83000.0]
 [1.0 0.0 0.0 37.0 67000.0]]
```

الشكل 20. البيانات الفئوية بعد التعديل في خرج البرنامج

كما نرى من الخرج السابق فإن جميع المتغيرات رمزت بأرقام 0 و1 وقسمت إلى 3 اعمدة ويمكن رؤيتها بوضوح من خلال variable explorer section من خلال الضغط على المتغير X

X - NumPy object array (read only)

	0	1	2	3	4
0	1.0	0.0	0.0	44.0	72000.0
1	0.0	0.0	1.0	27.0	48000.0
2	0.0	1.0	0.0	30.0	54000.0
3	0.0	0.0	1.0	38.0	61000.0
4	0.0	1.0	0.0	40.0	63777.7777777…
5	1.0	0.0	0.0	35.0	58000.0
6	0.0	0.0	1.0	38.7777777777…	52000.0
7	1.0	0.0	0.0	48.0	79000.0
8	0.0	1.0	0.0	50.0	83000.0
9	1.0	0.0	0.0	37.0	67000.0

الشكل 21. البيانات الفئوية بعد التعديل في واجهة Spyder

من أجل المتغير purchased: سوف نستخدم فقط LabelEncoder. ولن نستخدم OneHotEncder لأن المتغيرات purchased تحتوي فقط اثنين إما yes او no والتي سترمز مباشرة ب 0 و 1

```
# Encoding the Dependent Variable
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
print(y)
```

الشكل 22. الكود اللازم لترميز الخرج

y - NumPy object array

	0
0	0
1	1
2	0
3	0
4	1
5	1
6	0
7	1
8	0
9	1

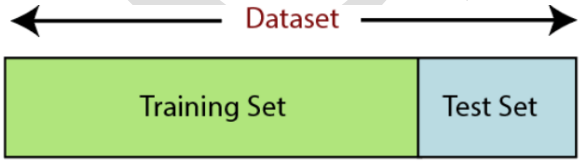
الشكل 23. الخرج بعد عملية الترميز

6. فصل البيانات إلى بيانات تدريب واختبار:

في المعالجة المسبقة للبيانات في ML. تقسم قاعدة البيانات إلى مجموعة تدريب ومجموعة اختبار. وهي واحدة من الخطوات الحرجة من معالجة البيانات data pre-processing وهذا يؤثر على تحسين أداء نموذج ML. بفرض، في حال أعطينا بيانات التدريب لنموذج من خلال قاعدة البيانات واختبرناها من خلال قاعدة البيانات مختلفة تماماً. فان ذلك سوف ينشأ صعوبات من أجل النموذج لفهم الترابط بين النماذج.

في حال دربنا النموذج بشكل جيد وكانت دقة الخرج عالية جداً، ولكن في حال قدمت له قاعدة البيانات جديدة فإن ذلك سيخفض الأداء. سنحاول دوماً إنشاء نموذج ML والذي يعمل بشكل جيد مع بيانات التدريب ومع بيانات الاختبار.

ويمكن تقسيم الـ قاعدة البيانات نفسها بالشكل:



الشكل 24. تقسيم قاعدة البيانات

مجموعة التدريب Training Set: هي مجموعة جزئية من قاعدة البيانات لتدريب نموذج ML ونكون مسبقا على معرفة بالخرج.

مجموعة الاختبار Test Set: هي مجموعة جزئية من قاعدة البيانات لاختبار نموذج ML ومن خلال استخدامها يتنبأ النموذج بالخرج.

ولفصل الـ قاعدة البيانات نستخدم الأسطر البرمجية التالية:

```
#Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 1)
print(X_train)
print(X_test)
print(y_train)
print(y_test)
```

الشكل 25. الكود اللازم لفصل البيانات الى تدريب واختبار

يستخدم السطر الأول لفصل المصفوفات من قاعدة البيانات إلى مجموعة تدريب واختبار عشوائية.

لدينا في السطر الثاني أربع متغيرات للخرج وهي:

- X_train: خواص مصفوفة التدريب
- X_test: خواص مصفوفة الاختبار
- y_train: المتغيرات غير المستقلة لبيانات التدريب
- y_test: المتغيرات غير المستقلة لبيانات الاختبار

في تابع train_test_split( ) يتم تمرير 4 بارامترات بحيث أول بارامترين هما مصفوفتا البيانات و test_size من أجل تحديد حجم مجموعة الاختبار وقد يكون 0.2، 0.3،0.5 والتي تعطي معدل التقسيم إلى مجموعات تدريب واختبار. البارامتر الأخير هو random_state ويستخدم من أجل البذرة للمولد العشوائي وإلا سنحصل على نفس النتيجة دوماً والقيمة الأكثر استخداماً لهذا المتغير هي 42، من خلال تنفيذ الكود التالي نحصل على:

X_train - NumPy object array (read only)

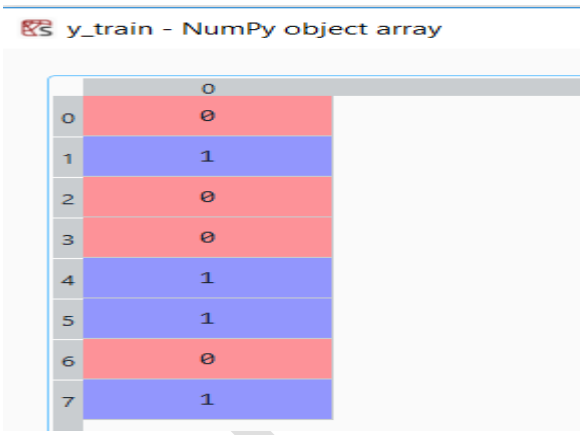
	0	1	2	3	4
0	0.0	0.0	1.0	38.7777777777…	52000.0
1	0.0	1.0	0.0	40.0	63777.7777777…
2	1.0	0.0	0.0	44.0	72000.0
3	0.0	0.0	1.0	38.0	61000.0
4	0.0	0.0	1.0	27.0	48000.0
5	1.0	0.0	0.0	48.0	79000.0
6	0.0	1.0	0.0	50.0	83000.0
7	1.0	0.0	0.0	35.0	58000.0

الشكل 26. بيانات الدخل الخاصة بالتدريب

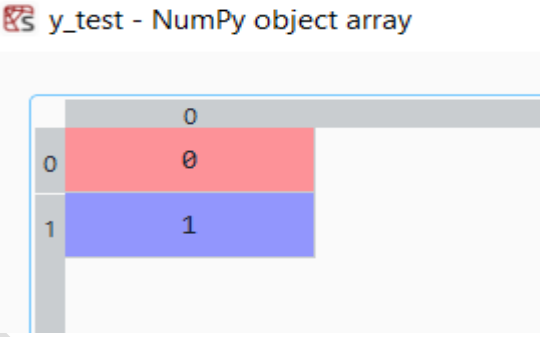
X_test - NumPy object array (read only)

	0	1	2	3	4
0	0.0	1.0	0.0	30.0	54000.0
1	1.0	0.0	0.0	37.0	67000.0

الشكل 27. بيانات الدخل الخاصة بالاختبار



الشكل 28. بيانات الخرج الخاصة بالتدريب



الشكل 29. بيانات الخرج الخاصة بالاختبار

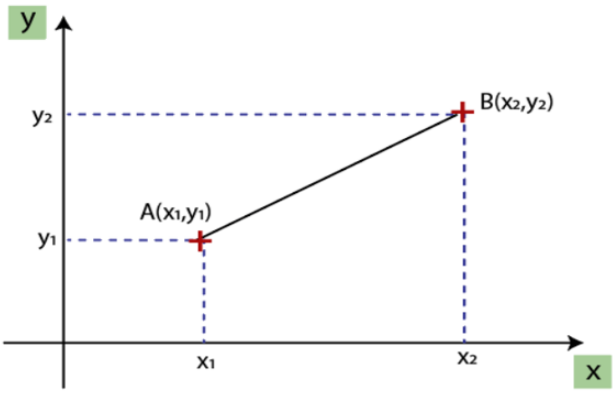
7. معايرة السمات:

الخطوة الأخيرة في عملية المعالجة المسبقة للبيانات في ML. هي تقنية لتوحيد المتغيرات المستقلة من قاعدة البيانات في مجال محدد. توضع المتغيرات في نفس المجال وفي نفس scale، وبالنتيجة لن يوجد أية متغيرات ستسيطر على الأخرى. مثلاً: لنفرض لدينا:

Age	Salary
44	72000
27	48000
30	54000
38	61000
40	nan
35	58000
nan	52000
48	79000
50	83000
37	67000

الشكل 30. قيم البيانات التي بحاجة معايرة

كما نلاحظ فإن age و Salary ليست ضمن نفس المجال وأن نموذج ML يكون مرتكزاً على المسافة الإقليدية Euclidean distance، في حال لم نوحد المتغيرات فإن ذلك سيسبب مشاكل في النموذج. تعطى المسافة الإقليدية بالشكل:



الشكل 31. المسافة الإقليدية.

وتوصف رياضيا بالصيغة:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

ففي حال حساب أي قيمتين من age و salary عندها فإن قيمة salary ستكون مسيطرة، وذلك سينتج نتائج خاطئة. لإزالة هذه المشكلة نحتاج القيام بتحجيم الخواص لـ ML.

يوجد طريقتين للقيام بذلك في ML وهما:

طريقة standardization

$$x' = \frac{x - mean(x)}{a}$$

حيث x' تمثل القيمة الجديدة و x تمثل القيمة الأصلية والتابع mean يمثل الوسيط و a تمثل الانحراف المعياري

طريقة Normalization

$$x' = \frac{x - min(x)}{\max(x) - \min(x)}$$

سنستخدم في مثالنا طريقة standardization وسندرج StandardScaler من sklearn.preprocessing بالشكل:

```python
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train[:, 3:] = sc.fit_transform(X_train[:, 3:])
X_test[:, 3:] = sc.transform(X_test[:, 3:])
print(X_train)
print(X_test)
```

الشكل 32. الكود اللازم لمعايرة بيانات الدخل

حيث يتم ادراج الصنف ومن ثم يطبق التابع transform على كل الأسطر المراد إجراء عملية المعايرة لها.

X_train - NumPy object array (read only)

	0	1	2	3	4
0	0.0	0.0	1.0	-0.1915918438…	-1.0781259408…
1	0.0	1.0	0.0	-0.0141172937…	-0.0701316764…
2	1.0	0.0	0.0	0.56670850653…	0.63356243271…
3	0.0	0.0	1.0	-0.3045301939…	-0.3078661727…
4	0.0	0.0	1.0	-1.9018011447…	-1.4204636155…
5	1.0	0.0	0.0	1.14753430682…	1.23265336345…
6	0.0	1.0	0.0	1.43794720696…	1.57499103816…
7	1.0	0.0	0.0	-0.7401495441…	-0.5646194287…

الشكل 33. بيانات الدخل الخاصة بالتدريب بعد المعايرة

X_test - NumPy object array (read only)

	0	1	2	3	4
0	0.0	1.0	0.0	-1.4661817944…	-0.9069571034…
1	1.0	0.0	0.0	-0.4497366439…	0.20564033932…

الشكل 33. بيانات الدخل الخاصة بالاختبار بعد المعايرة

لم تنفذ أي عملية معايرة على بيانات الخرج لأنه تأخذ في حالتنا قيمتين فقط هما إما 0 أو 1.

بعد القيام بعملية المعالجة المسبقة للبيانات السابقة يمكن استخدام أي نموذج من نماذج التعلم الآلي وتطبيقه على هذه البيانات وتقييم أداءه.

VIII. الخاتمة

يمكن استعمال التعلم الآلي من أجل حل مشاكل متعددة في مجالات مختلفة. تعد لغة Python من اللغات الممكن استخدامها ضمن مجال التعلم الآلي بشكل واسع بما يساعد في حل المشكلات المتعددة. تعرفنا ضمن هذا المقال على عملية المعالجة المسبقة للبيانات باستخدام لغة Python بحيث أصبحت البيانات جاهزة لإعطائها لنموذج التعلم الآلي ليتدرب عليها. ضمن المقالات المستقبلية سوف نقوم بإعطاء شرح تفصيلي لخوارزميات التعلم الآلي بأنواعها المختلفة سواء للتعلم بإشراف أو دون إشراف مع إجراء عمليات معالجة مسبقة موسعة جداً.

المراجع

[1]. https://www.javatpoint.com, last visit 3 Mar 2024
[2]. A. Muller and S. Guido, "Introduction to Machine Learning with Python" an O'REILLY book, http://oreilly.com/
[3]. https://www.kaggle.com/datasets, last visit, 7 Mar 2024.
[4]. https://archive.ics.uci.edu/ml/index.php, last visit, 4 Mar 2024.
[5]. https://registery.opendata.aws, last visit, 7 Mar 2024
[6]. https://toolbox.google.com/datasetsearch, last visit, 4 Mar 2024
[7]. https://msropendata.com, last visit, 4 Mar 2024
[8]. https://github.com/awesomedata/awesome-public-dataset, last visit, 4 Mar 2024
[9]. https://www.visualdata.io, last visit, 4 Mar 2024
[10]. https://scikit-learn.org/stable/datasets/index.html, last visit, 7 Mar 2024
[11]. https://www.superdatascience.com/pages/machine-learning, last visit 28 Feb 2024.