وزارة التعليم العالي والبحث العلمي

جامعة بابل

كلية العلوم للبنات

قسم علوم الحاسبات

نظام لتشخيص فايروس كورونا باستخدام خوارزميات التعلم الالي

مشروع تخرج مقدم لنيل شهادة البكالوريوس في علوم الحاسبات

من قبل الطالبة:

تبارك عباس رياح

بإشراف

د. علي كاظم محمد

2023

بسم الله الرحمن الرحيم

((هُوَ الَّذِي جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا

وَقَدَّرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابَ مَا

خَلَقَ اللَّهُ ذَلِكَ إِلَّا بِالْحَقِّ يُفَصِّلُ الْآيَاتِ لِقَوْمٍ

يَعْلَمُونَ)) .

صدق الله العظيم

سورة يونس- الآية 5 .

الاهداء

الى اعز الناس واقربهم الى قلبي الى والدتي العزيزة ووالدي العزيز اللذان كانا عونا وسندا لي والى اخوتي واخواتي والى اساتذتي واهل الفضل علي الذين غمروني بالحب والتقدير والنصيحة والتوجيه والارشاد .

الى كل هؤلاء اهدي هذا العمل المتواضع .

الشـــكـــر والتقـديـر

نحمد الله عز وجل على اتمام هذا البحث العلمي ألهمنا الصحة والعافية

والعزيمة فالحمد لله حمداً كثيراً

اتقدم بجزيل الشكر والتقدير الى جميع من ساعدني واعانني في اكمال هذا البحث وخاصة استاذي ومشرف البحث العلمي د. علي كاظم محمد على كل ما قدمه لي من توجيهات ومعلومات قيمة ساهمت في اثراء موضوع دراستي في مختلف جوانبها .

واقدم شكري وتقدير الى الكادر التدريسي واساتذتي في قسم علوم الحاسوب .

جدول المحتويات

ت	المحتويات	الصفحات
.1	الآية القرآنية	2
.2	الاهداء	3
.3	الشكر والتقدير	4
.4	خلاصة	6
	الفصل الاول	
.5	المقدمة	8
.6	الهدف من البحث	9
.7	منهجية البحث	9
	الفصل الثاني	
.8	لغة بايثون	12-11
.9	التعلم الالي	14-13
.10	الخوارزميات	17 -15
.11	المكتبات	18
	الفصل الثالث : الجانب العملي	
.12	data collection	20
.13	feature extraction	21-20
.14	training and testing	23-22
.15	مقاييس الاداء	24
	الفصل الرابع	
.16	النتائج	32-26
.17	المصادر	33

خلاصة

يُعد كوفيد -19 فيروسًا جديدًا يسبب عدوى في كل من الجهاز التنفسي العلوي والرئتين. ازداد عدد الحالات والوفيات على أساس يومي على نطاق جائحة عالمي. أثبتت صور الأشعة السينية للصدر أنها مفيدة في مراقبة أمراض الرئة المختلفة ، وقد تم استخدامها مؤخرًا لمراقبة مرض COVID-19.

في هذه البحث ، باستخدام مناهج خوارزميات التعلم الآلي ،

تم استخدام تقنية HOG لاستخراج الميزات من بيانات الصورة ، وتم استخدامها لتصنيف COVID-19 وصور الصدر العادية (الصحية) بالأشعة السينية. تم استخدام الصور في تجربة الدراسة. تم استخدام دقة التصنيف كمقياس لأداء الدراسة.

الفصل الاول

1-1 المقدمة :-

تُعد أتمتة تشخيص الأمراض قدرة بارزة للتعلم الآلي وأنظمة الذكاء الاصطناعي. يمكن لنماذج تصنيف التعلم الآلي والتعرف على العلامة تحديد الاتجاهات والعلاقات في مجموعات البيانات ، مما يجعلها مناسبة تمامًا للعديد من التطبيقات لأتمته القرارات أو التطبيقات التي تتطلب تصنيف عينات غير معروفة ، بما في ذلك التطبيقات الطبية. لذلك ، نهدف إلى البحث التجريبي في عدة طرق للعثور على أفضل نظام يمكنه اكتشاف COVID-19. يخضع المرضى المشتبه بهم الذين يعانون من أعراض COVID-19 للفحص باستخدام الأشعة السينية للصدر أو التصوير المقطعي المحوسب للصدر ، وكلاهما يظهر أنماط مرض الرئة COVID-19. يعد فحص الصدر بالأشعة السينية أحد أكثر فحوصات التصوير الطبي تكرارًا وفعالية من حيث التكلفة. ومع ذلك ، على عكس صور التصوير المقطعي المحوسب ، وُجد أن التشخيص السريري للأشعة السينية للصدر أكثر صعوبة وتحديًا . كما ادعى الباحثون مرارًا وتكرارًا حول تقسيم الأشعة السينية ، فإن صور الأشعة السينية أكثر توفرًا ولكن يصعب أيضًا اكتشافها وتشخيصها. تم تقييم Covid-19 Chest X Rays أو CXR بواسطة لوجود توزيع محدد للمرض. ووجدوا أن السمة المميزة لـ COVID-19 pneu- monia هي تورط حقول الرئة الطرفية الخارجية التي تظهر عتامات ثنائية البؤرة أو غير مكتملة أو متكدسة. ومع ذلك ، فإن فيروسات أخرى مثل SARS-Cov-1 و MERS-Cov قد تسبب أيضًا الالتهاب الرئوي . لذلك ، يجب على أطباء الأشعة فحص صور الأشعة السينية لتحديد سبب عدوى الرئة. حفزنا هذا على التحقيق في إمكانية تصميم نظام CAD لاكتشاف COVID-19 من الأشعة السينية. بالإضافة إلى ذلك ، في مرحلة ما قبل التحليل ، لا يتم ضمان التشخيص الجزيئي الدقيق لـ COVID-19 دائمًا ويتأثر بجمع عينة الجهاز التنفسي المناسبة في الوقت المناسب من الموقع التشريحي الصحيح . هذا يؤكد الحاجة إلى طرق تشخيص أخرى غير PCR ، مثل الأشعة السينية. أيضًا ، على عكس مجموعات اختبار COVID-19 ، لا تتطلب أجهزة الأشعة السينية نقل العينات المصابة وهي متوفرة على نطاق واسع في جميع المستشفيات. يمكن استخدام الكشف بمساعدة الكمبيوتر أو التشخيص بمساعدة الكمبيوتر ، CAD ، كخط أول من الفحص وكطريقة تشخيص موازية.

1- 2 الهدف من البحث:-

من خلال تطبيق خوارزميات التعلم الآلي جنبًا إلى جنب مع صور الأشعة السينية والأشعة المقطعية للرئتين من أجل الكشف الدقيق والسريع عن المرض ، مما يساعد في :

1- التخفيف من مشكلة ندرة مجموعات الاختبار.
2- سرعة ودقة التشخيص للأمراض .
3- تسهيل تشخيص امراض الرئة .
4- التفريق بين اشعة الاشخاص المصابين والاشخاص السليمين بشكل اسرع.

1- 3 منهجية البحث:-

اولا: مشكلة البحث

بسبب ظهور مرض COVID-19 وصعوبة تشخيصه عن امراض الرئة الاخر ، ظهرت الحاجة لإيجاد الوسائل من أجل تشخيص المرض بصوره صحيحه . لذا فقد اصبحت الاشعة السينية احدى التطبيقات المهمة لتشخيص هذا المرض لإيجاد افضل الوسائل في المستشفيات .

ثانيا: أهمية البحث

لأجل التخلص من مشاكل من التشخيص الخاطئ واكتشاف المرض شكل صحيح .

ثالثا: هدف البحث

يهدف هذا البحث لبناء برنامج لتشخيص المرض , ويتم ذلك من خلال استلام مجموعه من الصور بالأشعة السينية للأشخاص المصابين بمرض COVID-19 وامراض رئة اخرى واشخاص سليمين , ومن خلال ذلك نستطيع تحليل البيانات لأجل اتخاذ الاجراءات المناسبة

الفصل الثاني

1-2 لغة بايثون :-

مقدمة الى بايثون

لغة البرمجة بايثون لغة برمجية للأغراض العامة, تفاعلية, موجهة الغرض وعالية المستوى. تم إنشاء هذه اللغة على يد العالم جودي فان وزم بين عامي 1985 – 1990 . وكما هو الحال في لغة البرمجة بيرل فإن مصدر شيفرة لغة بايثون متوفر لدى حكومة الاتحاد الوطني (GNU) كشهادة عامة.

مفتوحة المصدر وقابلة للتطوير. تعتبر لغة بايثون لغة تفسيرية متعددة الأغراض وتستخدم بشكل واسع في العديد من المجالات كبناء البرامج المستقلة باستخدام الواجهات الرسومية المعروفة وفي عمل برامج الويب، بالإضافة إلى استخدامها كلغة برمجة نصية للتحكم في أداء بعض من أشهر البرامج المعروفة أو في بناء برامج ملحقة لها. وبشكل عام يمكن استخدام بايثون لبرمجة البرامج البسيطة للمبتدئين ولإنجاز المشاريع الضخمة كأي لغة برمجية أخرى في نفس الوقت. غالباً ما يُنصح المبتدئين في ميدان البرمجة بتعلم هذه اللغة لأنها من بين أسرع اللغات البرمجية تعلماً.

نشأت بايثون في مركز CWI (مركز العلوم والحاسب الآلي) بأمستردام على يد جويدو فان رُزوم في أواخر الثمانينات من القرن المنصرم، وكان أول إعلان عنها في عام 1991. تم كتابة نواة اللغة بلغة C. أطلق فان رُزوم اسم "بايثون" على لغته تعبيرًا عن إعجابه بفِرقَة مسرحية هزلية شهيرة من بريطانيا، كانت تطلق على نفسها اسم مونتي بايثون (بالإنجليزية : Monty Python).

تتميز بايثون بمجتمعها النشط، كما أن لها الكثير من المكتبات البرمجية ذات الأغراض الخاصة والتي برمجها أشخاص من مجتمع هذه اللغة، مثلاً مكتبة PyGame التي توفر مجموعه من الوظائف من أجل برمجة الألعاب. ويمكن لبايثون التعامل مع العديد من أنواع قواعد البيانات مثل MySQLوغيره.[1]

استخدام لغة بايثون

أكثر ما يميز لغة Python هو بساطتها والقدرة على تعلمها بسرعة، لأنها تتيح للمستخدمين تطوير جميع أنواع التطبيقات، ومجالات استخدام لغة Python التالية [2]

- يتم استخدامه لإنشاء برامج وصفحات ويب مختلفة باستخدام أنظمة Python مثل Django و Flask.
- إنشاء تطبيقات الهاتف المحمول وتعديلها بما يتناسب مع تجربة المستخدم.
- من خلالها يمكن إنشاء البرامج بسهولة وسرعة أيضًا.
- يمكنك من خلاله تحديث البيانات والملفات المخزنة في قواعد البيانات.
- إنشاء المواقع الإلكترونية وإضافة الوظائف والخصائص المختلفة إليها وتعديلها .

خصائص لغة بايثون

تتمتع لغة برمجة Python بالعديد من المزايا، من أبرزها ما يلي [2]

- لغة عالية المستوى أي أنها برنامج أقرب ما يكون إلى لغة الإنسان، حيث يسهل فهمه وإتقانه.
- لغة مفتوحة المصدر أي أنها لغة متاحة للجميع بحيث يمكن للجميع استخدامها وتعديلها وفقًا لمتطلبات البرنامج الذي يتم إنشاؤه.
- لغة البرمجة الشيئية لأنها تجعل البرنامج يعكس سيناريوهات مشابهة للسيناريوهات التي تستهدف العالم الحقيقي، مما يساعد على إعداد تطبيقات جيدة الصنع.
- التضمين هذه لغة تضمن لمستخدميها إدراج البرامج الداخلية المكتوبة بلغات برمجة مختلفة مثل C أو C ++.
- اللغة المحمولة أي أنه يمكن نقل نفس البرامج إلى عدة أجهزة مختلفة دون الحاجة إلى إجراء أي تغييرات وإضافات، أي أنها لغة مرنة.
- هيكل الجمل بسيط يمكن حفظ أوامر الكود الخاصة به وإتقانها بسهولة وبسرعة دون تعقيد.
- متعدد الأغراض يمكن استخدامه في العديد من التطبيقات المختلفة، من تطبيقات الويب وتطبيقات الهواتف الذكية والمزيد.

2-2 machine learning :-

ما المقصود بتعلم الآلة؟

التعلم الآلي هو عِلم تطوير الخوارزميات والنماذج الإحصائية التي تستخدمها أنظمة الحاسوب لأداء المهام بدون تعليمات واضحة، اعتمادًا على الأنماط والاستدلال بدلاً من ذلك. وتستخدم أنظمة الحاسوب لوغاريتمات التعلم الآلي لمعالجة كميات كبيرة من البيانات السابقة والتعرّف على أنماط البيانات. وهذا يسمح لها بتوقع النتائج بصورة أكثر دقة من مجموعة بيانات مدخلة معينة. على سبيل المثال، يمكن لعلماء البيانات تدريب تطبيق طبي على تشخيص مرض السرطان من صور الأشعة السينية عن طريق تخزين ملايين الصور المفحوصة والتشخيصات المقابلة.[3]

ما أهمية التعلم الآلي؟

يساعد التعلم الآلي الأعمال عن طريق دفع عجلة النمو وفتح سبل إيرادات جديدة وحل مشكلات صعبة. فالبيانات هي القوة المحركة الحاسمة وراء صنع القرار التجاري، ولكن في الأحوال التقليدية، كانت تستخدم الشركات البيانات من مصادر مختلفة، مثل ملاحظات العملاء والموظفين والوضع المالي. وتعمل أبحاث التعلم الآلي على أتمتة هذه العملية وتحسينها. عن طريق استخدام البرمجيات التي تحلل كميات هائلة من البيانات بسرعات عالية، تستطيع الأعمال تحقيق النتائج بصورة أسرع. [3]

ما طريقة عمل التعلم الآلي؟

الفكرة الرئيسة وراء التعلم الآلي تتمثل في علاقة رياضية قائمة بين أي مزيج من بيانات المدخلات والمخرجات. لا يعرف نموذج التعلم الآلي هذه العلاقة بشكل مسبق، ولكنه يستطيع أن يخمن إذا تم تزويده بمجموعات بيانات كافية. وهذا يعني أن كل لوغاريتم للتعلم الآلي يُبنى على أساس دالة رياضية قابلة للتعديل. من خلال هذا الفهم الأساسي، يركز التعلم الآلي في المبدأ الذي ينص على أن كل نقاط البيانات المعقدة يمكن ربطها رياضيًا عن طريق أنظمة الحاسوب ما دامت مزودة بالقدر الكافي من البيانات وقدرة الحوسبة على معالجة هذه البيانات. لذلك، دقة المخرج تربطه علاقة طردية بحجم المدخل المُعطى. [3]

ما طرق التعلم الآلي؟

يمكن تصنيف الطرق إلى أربعة أساليب تعلم فريدة اعتمادًا على المخرج المتوقع ونوع المدخل [3].

1. التعلم الآلي تحت الإشراف
2. التعلم الآلي بدون إشراف
3. التعلم تحت الإشراف الجزئي
4. التعلم الآلي المعزز

ما المزايا والعيوب المرتبطة بالتعلم الآلي؟

ألقِ نظرة إلى بعض الأمور التي يستطيع التعلم الآلي فعلها والأمور التي يتعذر عليه فعلها [3] :

مزايا نماذج التعلم الآلي:

- تستطيع التعرف على اتجاهات البيانات وأنماطها التي ربما لن يستطيع البشر اكتشافها.
- تستطيع العمل بدون تدخل بشري بعد الإعداد. على سبيل المثال، التعلم الآلي في برمجيات الأمن السيبراني يستطيع مراقبة الحالات غير الطبيعية والتعرف عليها في حركة مرور الشبكة بدون أي مدخلات من جانب المسؤول.
- يمكن أن تصبح النتائج أكثر دقة بمرور الوقت.
- تستطيع التعامل مع مجموعة متنوعة من تنسيقات البيانات في بيئات ديناميكية وذات أحجام عالية ومعقدة.

عيوب نماذج التعلم الآلي:

- يُعد التدريب الأولي عملية مكلفة ومستغرقة للوقت. قد يصعب تنفيذها في حالة عدم توافر بيانات كافية.
- هي عملية كثيفة الحوسبة تتطلب استثمارًا مبدئيًا كبيرًا إذا كانت الأجهزة مُعدة في المقر.
- قد يكون من الصعب تفسير النتائج بشكل صحيح وإزالة عدم اليقين بدون مساعدة الخبراء .

3-2 الخوارزميات:-

1-3-2 خوارزمية KNN

تندرج خوارزمية K ضمن فئة التعلم بالإشراف وتستخدم للتصنيف والانحدار. إنها خوارزمية متعددة الاستخدامات، تُستخدم أيضًا لحساب المقادير المفقودة وإعادة تشكيل مجموعات البيانات. كما يوحي الاسم (K nearest neighbors) بأنه يعتبر (K nearest neighbors) (نقاط البيانات) للتنبؤ بالفئة أو القيمة المستمرة لنقطة البيانات الجديدة [4].

تعلم الخوارزمية:

التعلم القائم على المثيل (Instance-based learning) : لا نتعلم هنا الأوزان من بيانات التدريب للتنبؤ بالمخرجات (كما هو الحال في الخوارزميات المستندة إلى النموذج) ولكننا نستخدم حالات تدريب كاملة للتنبؤ بمخرجات البيانات غير المرئية. [4]

التعلم الكسول (Lazy Learning) : لا يتم تعلم النموذج باستخدام بيانات التدريب مسبقًا ويتم تأجيل عملية التعلم إلى وقت يتم فيه طلب التنبؤ في المثيل الجديد.

غير البارامترى (Non-Parametric) : فيKNN ، لا يوجد شكل محدد مسبقًا لوظيفة التعيين.

2-3-2 خوارزمية SVM

هي احدى خوارزميات التعلم المراقب Supervised Learning والتي يمكن استخدامها في مسائل التصنيف Classification والانحدار Regression وعادة تستخدم في مسائل التصنيف لفعاليتها ولحصولها على دقة ممتازة في اغلب البيانات المستعملة .
وهي على نوعين التصنيف الثنائي Binary Classification حيث تقوم بالتصنيف لمجموعتين فقط ,
و Multi-class Classification حيث تقوم بالتصنيف لأكثر من مجموعة .[5]

تعلم الخوارزمية:

الفكرة الرئيسية لعمل خوارزمية SVM هي ايجاد افضل مستوى فاصل a separating hyperplane الذي يفصل بين المجاميع classes وذلك عن طريق المحاولة لجعل الحاشية marginاكبر ما يمكن.
و الحاشية margin هي المسافة بين بيانات التدريب وبين المستوى الفاصل التي تعتبر محور عملية التدريب في خوارزمية SVM.[5]

3-3-2 خوارزمية DT45

ان خوارزمية شجرة القرار هي نموذج استكشافي يظهر على شكل شجرة , كما يعتبر اسمها , وبشكل دقيق يمثل كل فرع من فروع الشجرة سؤالا تصنيفيا, وتمثل اوراق الشجرة اجزاء من قاعدة البيانات تنتمي للتصنيفات التي تم بنائها.
ويكثر استخدام شجار القرار في الذكاء الصناعي , نظرا لما توفره من مزايا متعددة في هذا المجال.[6]

تعلم الخوارزمية :

تعتبر من خوارزميات التصنيف والتنبؤ فهي تقسم البيانات في كل فرع بدون انقاص اي منها , اي ان عدد السجلات الكلي في الفرع الام يساوي مجموع السجلات في الفروع المنبثقة منه.[6]

4-3-2 خوارزمية NB

Never Bayes هي طريقة تعلم آلي تعتمد على الاحتمالات تم اختراعها من قبل العالم توماس بيز . تستخدم لعمل تصنيف للبيانات مثلا تصنيف البريد الالكتروني الوارد الى مرغوب وغير مرغوب او تصنيف المشاعر للمستخدمين في وسائل التواصل الاجتماعي عن منتج معين او التنبؤ بالطقس.[7]

تعلم الخوارزمية :

نموذج بايز للتصنيف يتميز بسهولة البناء والتطوير والقدرة على معالجة البيانات الكبيرة ويتفوق في ذلك على عدد من الخوارزميات المتعقدة والمتقدمة. بحيث يتم تدريب النموذج بالبيانات وخصائصها المتاحة في قواعد البيانات ومن ثم يقوم النموذج بتحديد نوع السجلات الجديدة وتصنيفها بالاعتماد على البيانات والإحصاءات المتوفرة سابقاً لديه.
هناك عدة أساليب متفرعة من هذه الطريقة وتستخدم في كثير من أنظمة على سبيل المثال في التعرف على الرسائل المؤذية Spam .[7]

5-3-2 خوارزمية RF

الغابة العشوائية خوارزمية التعلم الآلي الشائعة تنتمي إلى تقنية التعلم بالإشراف. يمكن استخدامها لكل من مسائل التصنيف والانحدار في ML. تعتمد على مفهوم التعلم الجماعي، وهي عملية الجمع بين عدة مصنفات لحل مشكلة معقدة وتحسين أداء النموذج.[8]

تعلم الخوارزمية:

تعمل الغابة العشوائية على مرحلتين: الأولى هي إنشاء الغابة العشوائية من خلال الجمع بين N شجرة قرار، والثاني هو عمل تنبؤات لكل شجرة تم إنشاؤها في المرحلة الاولى.[8]

6-3-2 خوارزمية MLP Classifier

الشبكة العصبية ذو المستقبلات متعددة الطبقات (شبكة MLP) هي نوع من الشبكات العصبية التي تستخدم على نطاق واسع في التعلم الآلي و الذكاء الاصطناعي .إنها شبكة عصبية متجهة إلى الأمام ، مما يعني أن المعلومات تتدفق في اتجاه واحد من طبقة الإدخال إلى طبقة الإخراج .تتكون بنية ال MLPمن ثلاث طبقات أو أكثر: طبقة الإدخال Input layer ، طبقة مخفية واحدة أو أكثر Hidden layer ، و طبقة الإخراج .Output layer نقوم بتغذية طبقة الإدخال ببيانات الإدخال الخاصة بنا ونحصل على النتائج من طبقة الإخراج. يمكننا زيادة عدد الطبقة المخفية بقدر ما نشاء، لجعل النموذج أكثر دقة و تعقيدا وفقًا للمهمة التي نريد إنجازها.[9]

تعلم الخوارزمية :

هناك ثلاث خطوات أساسية يجب اتباعها عن تدريب النموذج.[9]

1- التمرير الامامي:

في هذه الخطوة من تدريب النموذج ، نقوم بتمرير المدخلات إلى النموذج وضربهن بالأوزان و إضافة قيمة التحيز في كل طبقة و الحصول على المخرجات التي تم حسابها للنموذج.

2- حساب قيمة دالة الفقد:

بمقارنة المخرجات التي تم تخمينها مع البيانات المتوقع الحصول عليها، نحسب الفقد أو نسبة الخطأ التي يتعين علينا إعادة نشرها (باستخدام خوارزمية الانتشار العكسي .(Backpropagation هناك العديد من دوال الفقد التي يمكن استخدامها بناءً على المخرجات المتطلبة حسابها.

3- الانتشار الخلفي:

بعد حساب الفقد، نقوم بإعادة نشر الفقد وتحديث أوزان النموذج باستخدام التدرج الاشتقاقي .و تعتبر هذه هي الخطوة الرئيسية في تدريب النموذج .حيث يتم في هذه الخطوة ضبط الأوزان وفقًا لتدفق التدرج في هذا الاتجاه.

بمجرد تدريب شبكة MLP ، يمكن استخدامها لعمل تنبؤات بشأن البيانات الجديدة .يتم تمرير بيانات الإدخال عبر الشبكة ، ويتم إنشاء الإخراج بواسطة طبقة الإخراج .يمكن استخدام شبكة MLP لمجموعة واسعة من المهام ، بما في ذلك التصنيف و الانحدار و التنبؤ.

4-2 المكتبات :-

1-4-2 Scikit-Learn

يمكن القول إن Scikit-Learn هي أهم مكتبة في Python للتعلم الآلي. بعد تنظيف البيانات ومعالجتها باستخدام Pandas أو NumPy . يتم استخدام Scikit-Learn لبناء نماذج التعلم الألي حيث تحتوي على الكثير من الأدوات المستخدمة في النمذجة والتحليل التنبئي .

يمكنك استخدام Scikit-Learn لبناء عدة أنواع من نماذج التعلم الآلي ، الخاضعة للإشراف وغير الخاضعة للإشراف ، والتحقق من صحة النماذج . وأهمية ميزة السلوك والكثير من الاستخدامات الأخرى .[10]

2-4-2 NumPy

NumPy هي مكتبة للغة برمجة Python , تضيف دعمًا للمصفوفات والمصفوفات الكبيرة متعددة الأبعاد، جنبا إلى جنب مع مجموعة كبيرة من الوظائف الرياضية عالية المستوى للعمل على هذه المصفوفات.

يستخدم Numpy بشكل أساسي لدعمه لمصفوفات الأبعاد . هذه المصفوفات متعددة الأبعاد أقوى 50 مرة مقارنة بقوائم بايثون . مما يجعل NumPy المفضل لعلماء البيانات.

يتم استخدام Numpy أيضًا بواسطة مكتبات أخرى مثل TensorFlow لحسابها الداخلي على الموترات، يوفر Numpy أيضًا وظائف سريعة مجمعة مسبقا للإجراءات العددية، والتي قد يكون من الصعب حلها يدويا، لتحقيق كفاءة أفضل ، يستخدم NumPy حسابات مصفوفة ، لذلك يصبح العمل مع فئات متعددة أمرا سهلا.[11]

3-4-2 Scikit-image

هي مكتبة معالجة صور مفتوحة المصدر للغة برمجة بايثون. وهو يتضمن خوارزميات للتجزئة والتحولات الهندسية ومعالجة مساحة اللون والتحليل والتصفية والتشكل واكتشاف الميزات والمزيد. تم تصميمه للتعامل مع مكتبات Python العددية والعلمية NumPy و SciPy.[12]

الفصل الثالث
الجانب العملي

العمل الخاص بنا في هذا في هذا المشروع ينقسم الى المراحل التالية :

1- data collection:

في هذه البحث ، تم انشاء واحدة من مجموعة البيانات لاكتشاف COVID-19 ، باستخدام كل من صور الأشعة. ترد مقدمة موجزة عن مجموعة البيانات في الجدول 1. يمكننا أن نرى أن مجموعة البيانات تحتوي على 128 صورة COVID-19. هناك 125 صورة طبيعية و 39 صورة ذات الرئة (الفيروسية والبكتيرية). لذلك ، يبلغ إجمالي الصور في مجموعة بيانات COVID-R 292 صورة .

Class	Number of images
COVID-19	128
Normal	125
Pneumonia (Viral + Bacteria)	39

يتم استدعائها في النظام حسب الكود التالي

```
path = "data/*.jpg"
```

2- feature extraction:

من مجموعة بيانات COVID-R سوف يتم استخلاص مجموعه من الصفات حسب الطريقة المستخدمة HOG التي **يتم استدعائها في النظام حسب الكود التالي :**

```
X,Y = im.img(path).feature_extraction_HOG()
```

2-1 Histogram of Oriented Gradient (HOG):

هو واصف ميزة يستخدم في معالجة الصور ، بشكل أساسي لاكتشاف الكائنات. واصف ميزة هو تمثيل لصورة أو تصحيح صورة يبسط الصورة عن طريق استخراج معلومات مفيدة منها.

المبدأ الكامن وراء الرسم البياني لواصف التدرجات الموجهة هو أنه يمكن وصف مظهر وشكل الكائن المحلي داخل صورة من خلال توزيع تدرجات الكثافة أو اتجاهات الحافة. تعتبر مشتقات x و y للصورة (التدرجات) مفيدة لأن حجم التدرجات كبير حول الحواف

والزوايا بسبب التغيير المفاجئ في الكثافة ونعلم أن الحواف والزوايا تحتوي على **معلومات أكثر بكثير حول شكل الكائن أكثر من المناطق المسطحة. لذلك ، يتم استخدام الرسوم البيانية لاتجاهات التدرجات كميزات في هذا الواصف.**

2-2 كيف يعمل الرسم البياني للتدرجات(HOG)

1- pre-processing

2- Computing Gradient

3- حساب الرسم البياني للتدرجات في 8 × 8 خلايا

4- Block Normalization

5- create HOG description vectors

6- classification

7- النتائج

كود العمل في المشروع :

```
def feature_extraction_HOG(self, cellx=64,celly=64):
    data_class=[]
    data=[]
    for filename in glob.glob(self.path):  # assuming gif
        im = Image.open(filename)
        size = 700, 800
        im_resized = im.resize(size, Image.ANTIALIAS)
        rgb_im = im_resized.convert('RGB')
        if "normal" in filename:
            data_class.extend([1])
        else:
            data_class.extend([0])
        fd, hog_image = hog(rgb_im, orientations = 9,
pixels_per_cell = (cellx, celly),
                            cells_per_block = (2, 2),
visualize = True, multichannel = True)
        if len(data)==0:
            data=fd
        else:
            data = np.vstack([data,fd])
    return data,data_class
```

3- training and testing :

تم تدريب البيانات حسب خوارزميات التعلم الالي:

3-1 خوارزمية KNN : كود التدريب حسب هذه الخوارزمية

```
def KNN(self,nneighbors=3):
    warnings.filterwarnings ( 'ignore' )
    knn = neighbors.KNeighborsClassifier (
n_neighbors=nneighbors )
    knn.fit ( self.traindata, self.traincalss )
    start = time.time ()
    pred_knn = knn.predict ( self.testdata )
    end = time.time ()
    retime = (end - start) / len ( self.testclass )
    print("The time of KNN is:", retime)
    return
[suppervise_ev_matrix(pred_knn,self.testclass).evaluation
(), retime,knn.classes_,len ( self.testdata[ 0 ] )]
```

3-2 خوارزمية SVM: كود التدريب حسب هذه الخوارزمية

```
def SVM(self):
    warnings.filterwarnings ( 'ignore' )
    clf = svm.SVC (gamma='auto')
    clf.fit (self.traindata, self.traincalss )
    start = time.time()
    pred_svm=clf.predict(self.testdata)
    end = time.time ()
    retime= (end - start) / len ( self.testdata )
    print("The time of SVM is:", retime)

    return
[suppervise_ev_matrix(pred_svm,self.testclass).evaluation
(),retime,clf.classes_,len ( self.testdata[ 0 ] )]
```

3-3 خوارزمية DT45 : كود التدريب حسب هذه الخوارزمية

```
def DT45(self):
    warnings.filterwarnings ( 'ignore' )
    Dtree = DecisionTreeClassifier ()
    Dtree.fit(self.traindata,self.traincalss)
    start = time.time()
    pred_Dtree=Dtree.predict(self.testdata)
    end = time.time ()
    retime = (end - start) / len ( self.testdata )
    print("The time of DT45 is:", retime)

    return
[suppervise_ev_matrix(pred_Dtree,self.testclass).evaluati
on(), retime,Dtree.classes_,len ( self.testdata[ 0 ] )]
```

4-3 خوارزمية NB : كود التدريب حسب هذه الخوارزمية

```
def NB(self):
    warnings.filterwarnings ( 'ignore' )
    NaivB = GaussianNB()
    NaivB.fit(self.traindata,self.traincalss)
    start = time.time()
    pred_NaivB = NaivB.predict ( self.testdata )
    end = time.time ()
    retime = (end - start) / len ( self.testdata )
    print("The time of NB is:", retime)

    return
[suppervise_ev_matrix(pred_NaivB,self.testclass).evaluati
on(), retime,NaivB.classes_,len ( self.testdata[0])]
```

5-3 خوارزمية RF : كود التدريب حسب هذه الخوارزمية

```
def RF(self):
    warnings.filterwarnings ( 'ignore' )
    RFC = RandomForestClassifier (
max_features="sqrt",bootstrap=False,random_state=0 )
    RFC.fit ( self.traindata, self.traincalss )
    start = time.time ()
    pred_RF = RFC.predict ( self.testdata )
    end = time.time ()
    retime = (end - start) / len ( self.testclass )
    print("The time of RF is:", retime)

    return
[suppervise_ev_matrix(pred_RF,self.testclass).evaluation(
), retime, RFC.classes_, len ( self.testdata[ 0 ] )]
```

6-3 خوارزمية MLP Classifier : كود التدريب حسب هذه الخوارزمية

```
def MLPClassifier(self):
    pre = MLPClassifier()
    pre.fit(self.traindata,self.traincalss)
    start = time.time()
    pred_final=pre.predict(self.testdata)
    end = time.time ()
    retime = (end - start) / len ( self.testdata )
    print("The time of MLPClassifier is:", retime)

    return
[suppervise_ev_matrix(pred_final,self.testclass).evaluati
on(), retime,pre.classes_,len ( self.testdata[0])]
```

4- مقاييس الاداء :

يتم تقييم أداء النموذج بناءً على Time , و Accuracy، و Precision ، و Recall ، وf1scor ، و Confusion Matrix.

كود العمل في البرنامج :

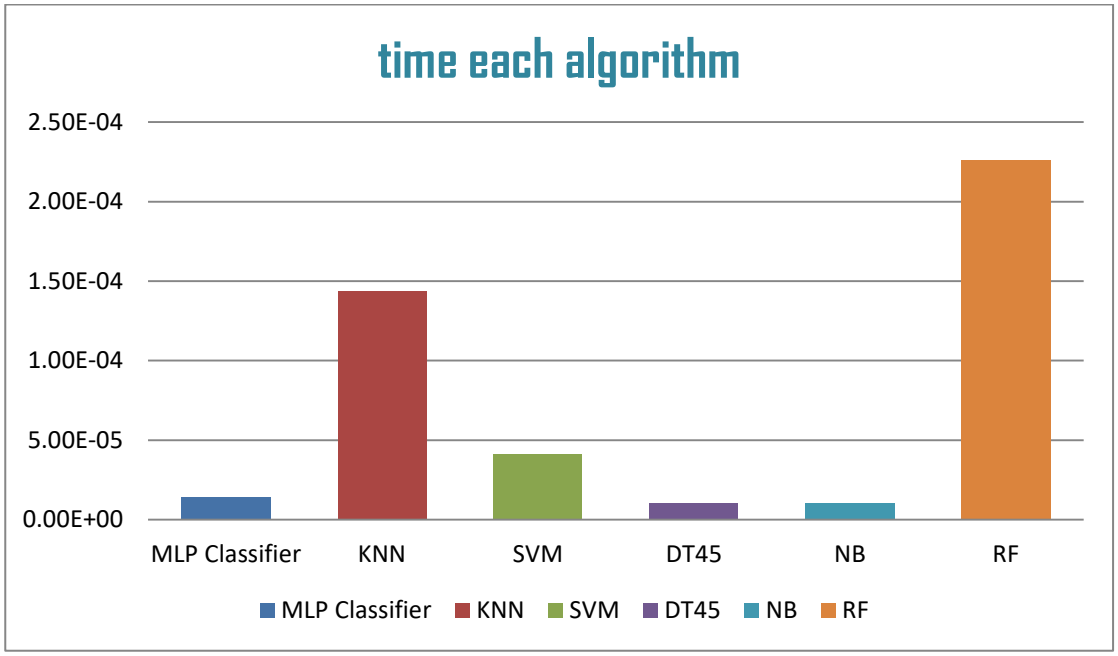
```
class suppervise_ev_matrix:
    def __init__(self, predict=None, test=None):
        self.predict=predict
        self.test=test


def evaluation(self):
    Accuracy = metrics.accuracy_score ( self.test,
self.predict ) * 100
    Precision = metrics.precision_score ( self.test,
self.predict, average='macro' ) * 100
    Recall = metrics.recall_score ( self.test,
self.predict, average='macro' ) * 100
    f1scor = 2 * (Precision * Recall) / (Precision +
Recall)
    Confusion_Matrix = metrics.confusion_matrix (
self.test, self.predict )
    print('Accuracy=', Accuracy,' \nPrecision',
Precision,
          '\nRecall',Recall, '\nf1scor',f1scor,
'\nConfusion_Matrix',Confusion_Matrix )
    return Accuracy, Precision, Recall, f1scor,
Confusion_Matrix
```

الفصل الرابع
النتائج والاعمال المستقبلية

مقدمة :-

في هذا الفصل يتم توضيح النتائج التي تم الحصول عليها من عملية التدريب لكل خوارزمية بناءً على مقايس الاداء المستخدمة (Accuracy , Time، Precision ، Recall ، f1scor ، Confusion Matrix.)

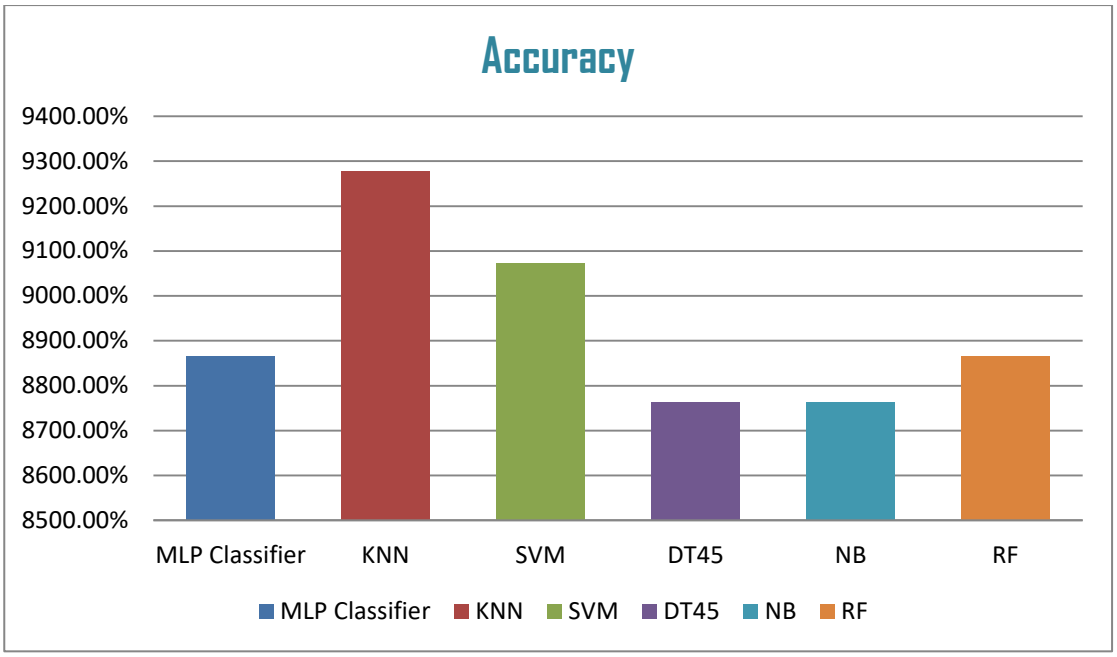
1-4 ناتج مقياس الاداء (time) لكل خوارزمية:



خوارزمية NB و DT45 استغرقت القليل من الوقت لإظهار النتائج بالنسبة للخوارزميات الاخرى .

عكس خوارزمية RF استغرقت الوقت الاكبر لإظهار النتائج بالنسبة للخوارزميات الاخرى.

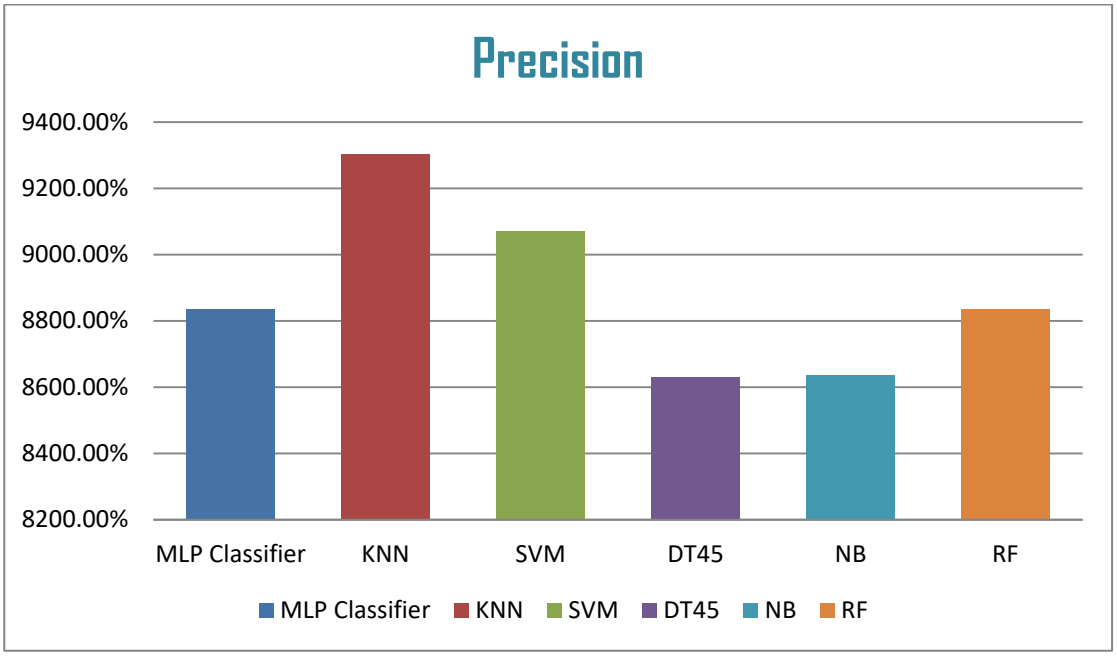
2-4 ناتج مقياس الاداء (Accuracy) لكل خوارزمية:



خوارزمية KNN الاعلى Accuracy في إظهار النتائج بالنسبة للخوارزميات الاخرى .

عكس خوارزمية NB و DT45 الاقل Accuracy في إظهار النتائج بالنسبة للخوارزميات الاخرى.

3-4 ناتج مقياس الاداء (Precision) لكل خوارزمية:

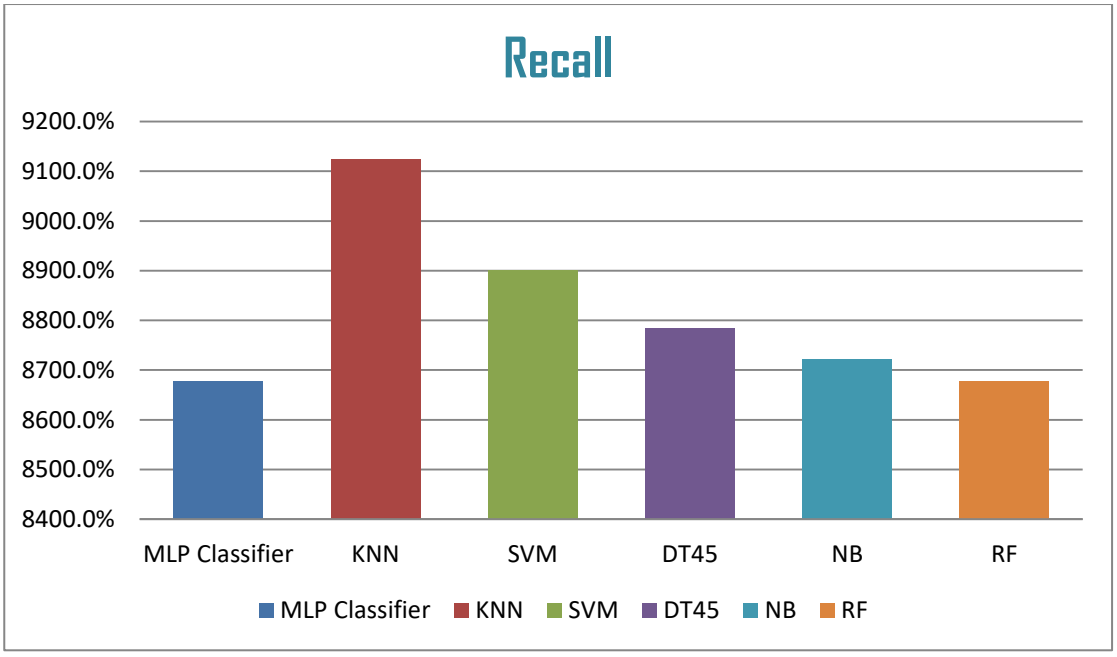


خوارزمية KNN الاعلى Precision في إظهار النتائج بالنسبة للخوارزميات الاخرى .

عكس خوارزمية NB و DT45 الاقل Precision في إظهار النتائج بالنسبة للخوارزميات الاخرى.

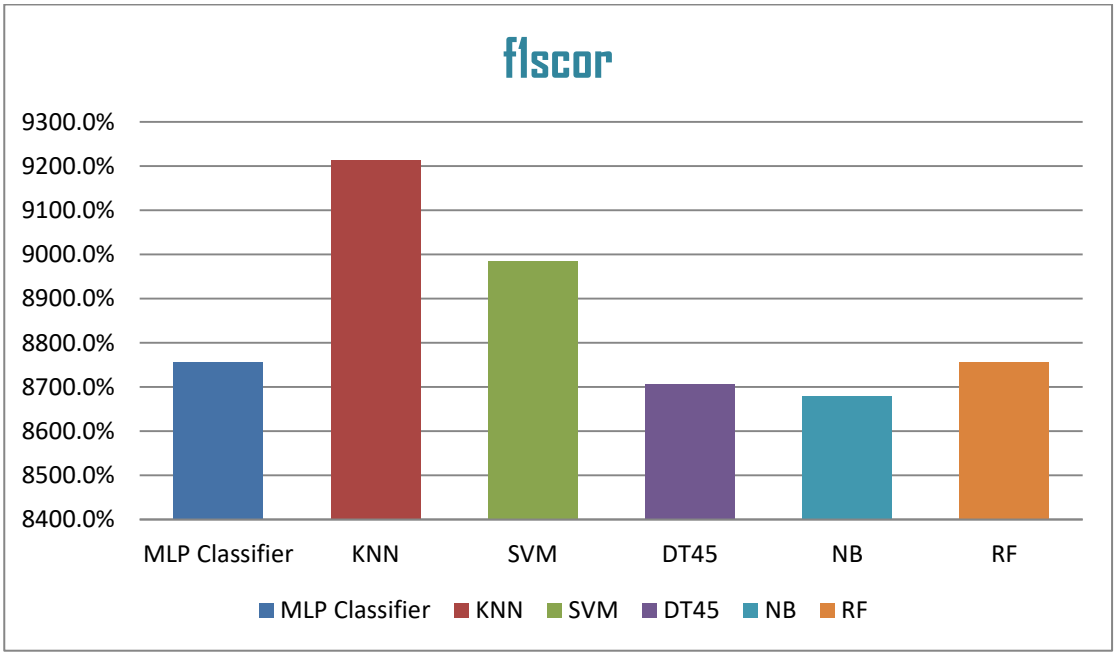
وتكون قيم Precision مقاربة الى قيم Accuracy

4-4 ناتج مقياس الاداء (Recall) لكل خوارزمية:



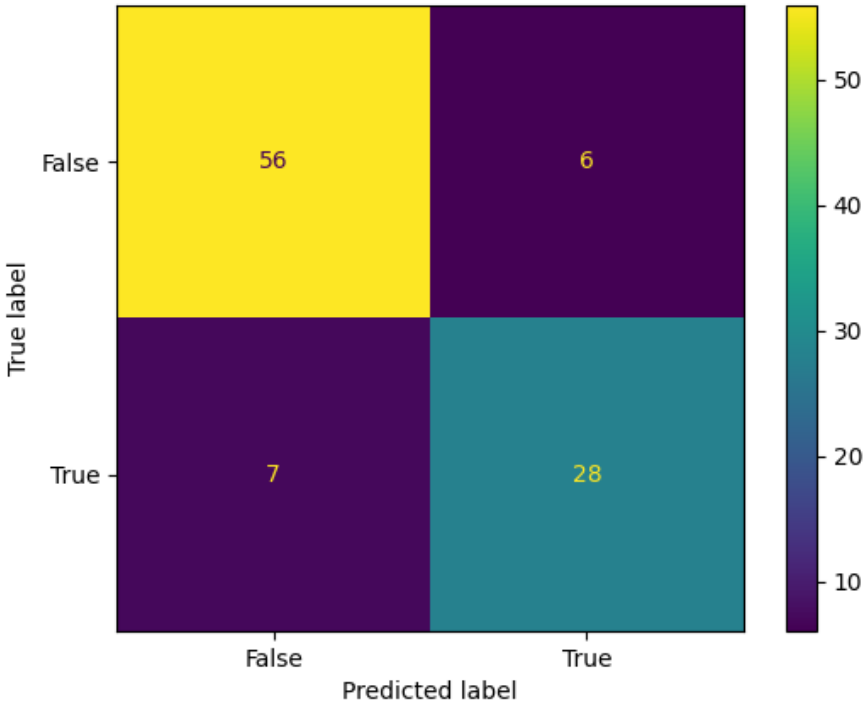
خوارزمية KNN الاعلى Recall في إظهار النتائج بالنسبة للخوارزميات الاخرى .
عكس خوارزمية RF و MLPClassifier الاقل Recall في إظهار النتائج بالنسبة للخوارزميات الاخرى.

5-4 ناتج مقياس الاداء (f1scor) لكل خوارزمية:

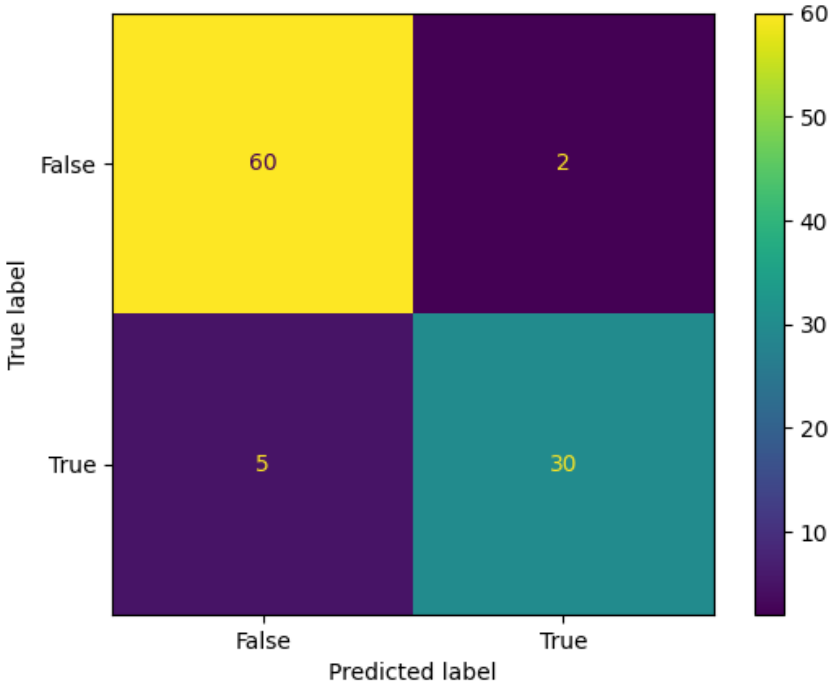


خوارزمية KNN الاعلى Precision في إظهار النتائج بالنسبة للخوارزميات الاخرى .
عكس خوارزمية NB و DT45 الاقل Precision في إظهار النتائج بالنسبة للخوارزميات الاخرى.

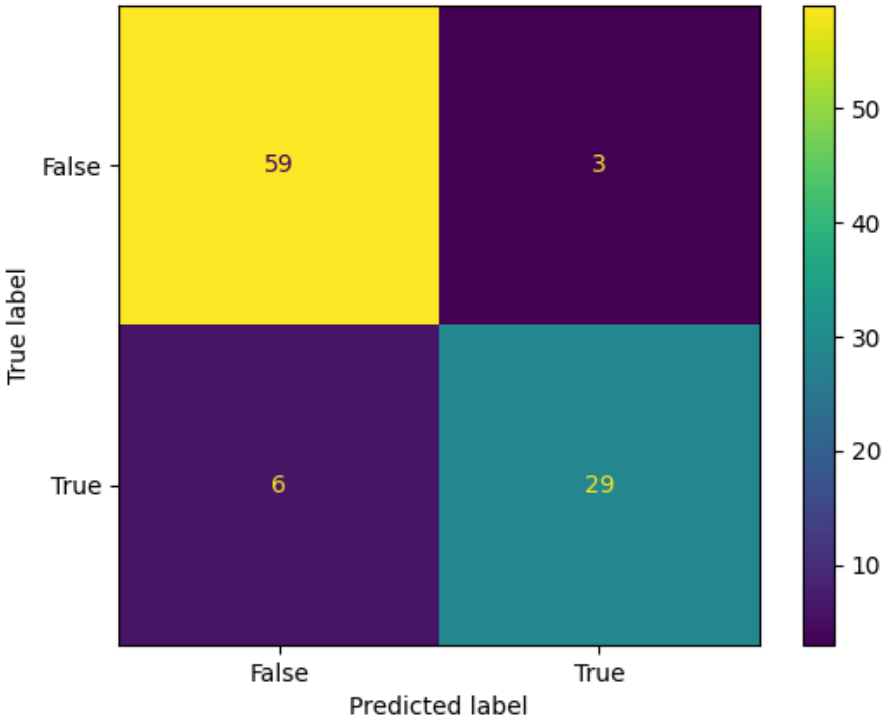
1-6-4 نتائج تصنيف البيانات لخوارزمية MLPClassifier



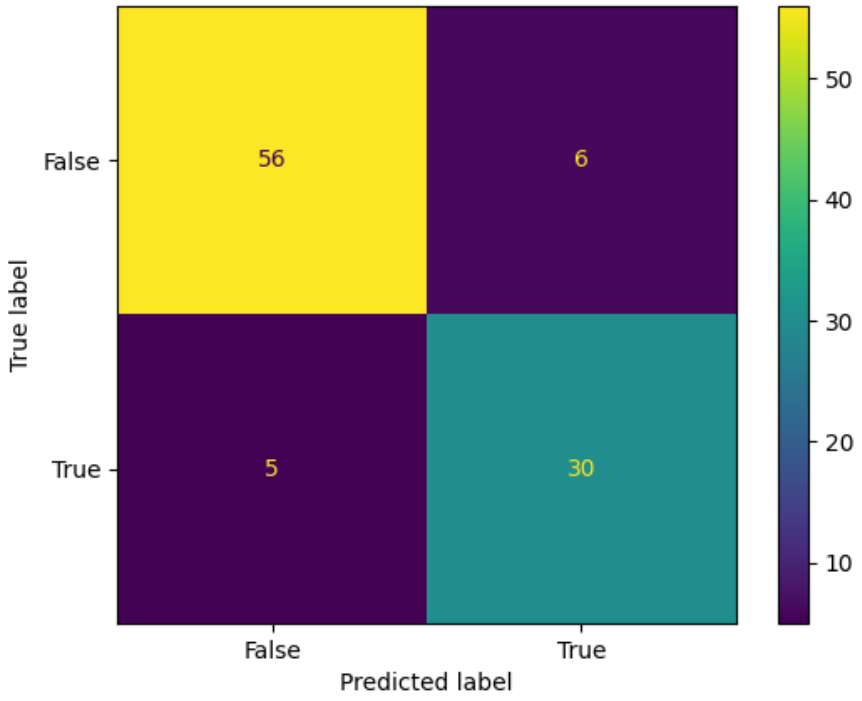
2-6-4 نتائج تصنيف البيانات لخوارزمية KNN



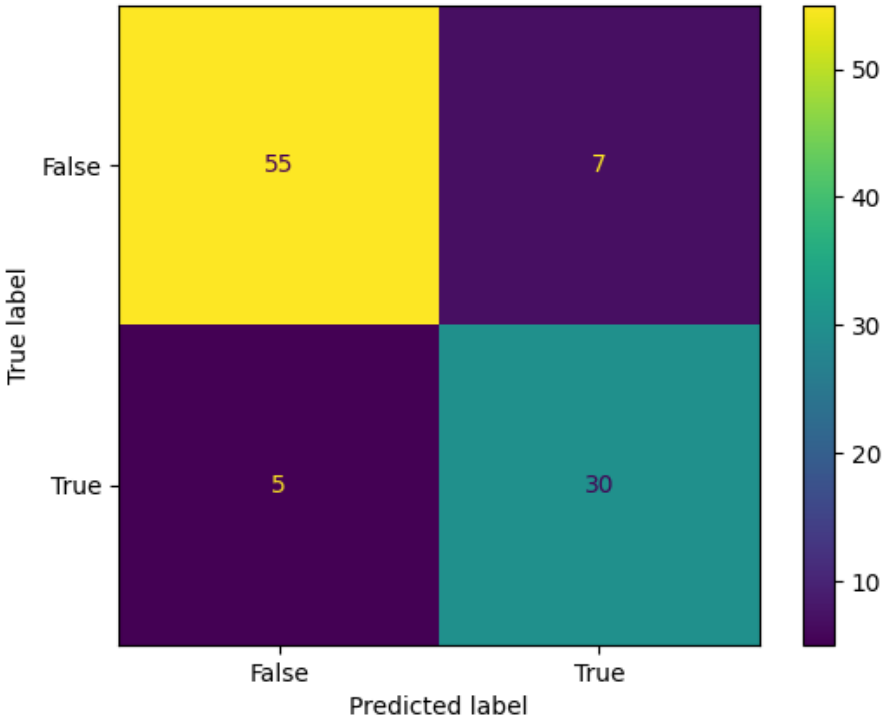
3-6-4 نتائج تصنيف البيانات لخوارزمية SVM



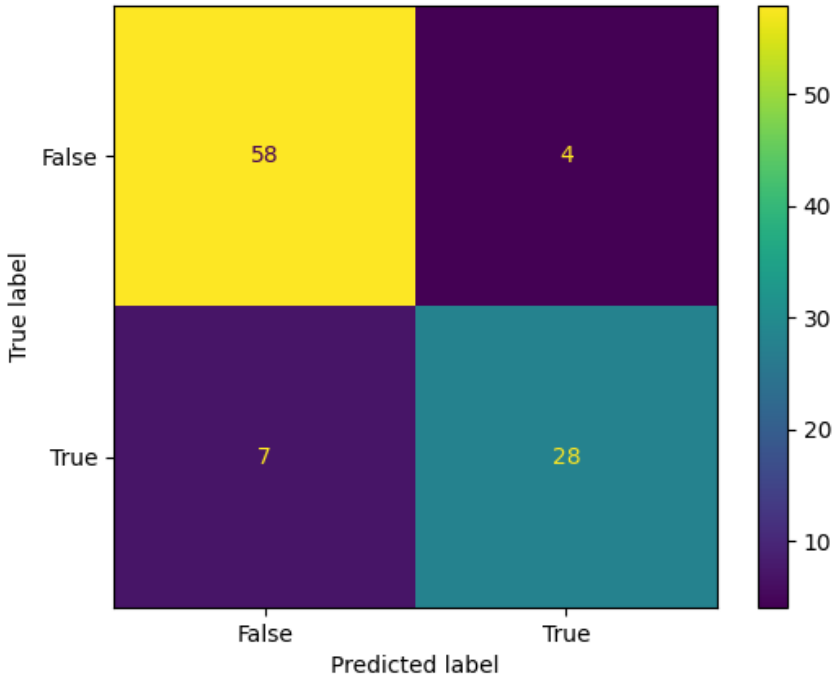
4-6-4 نتائج تصنيف البيانات لخوارزمية DT45



5-6-4 نتائج تصنيف البيانات لخوارزمية NB



6-6-4 نتائج تصنيف البيانات لخوارزمية RF



7-4 الاعمال المستقبلية :

1- استخدامه للعمل على اشعة سينية لأمراض اخرى.
2- يمكن تطويره ليكون مناسبا لاستخدامه في المستشفيات.
3- استخدام خوارزميات اخرى من اجل زيادة دقة وسرعة التشخيص.

المصادر

1- Python, W. (2021). Python. *Python Releases Wind*, *24*.

2- Vallat, R. (2018). Pingouin: statistics in Python. *J. Open Source Softw.*, *3*(31), 1026.

3- الهادي, م & ,.محمد. (2022). تعلم الآلة لتطبيقات المخاطرة العالية .مجلة الجمعية المصرية لنظم المعلومات وتكنولوجيا الحاسبات,28(28), 41-40 .

4- علي عبد الله الحسين, د. نبيل دحدوح & ,عيد العبود. (2021). تقليل عدد نطاقات الصور الطيفية باستخدام خوارزمية تحليل المكونات الرئيسية وتطبيقه في التعرف على الوجوه .*سلسلة العلوم الهندسية الميكانيكية و الكهربائية و المعلوماتية* ,43(16) .

5- مصعب عمر محمد عبد الخالق. (2020).*تشخيص مرض سرطان الثدي باستخدام خوارزمية Support Vector Machine* (جامعة إفريقيا العالمية Doctoral dissertation).

6- زيدان, محمد, عامر & ,صالح. (2022). بناء نموذج ذكي للتنبؤ باختيار الكلية الجامعية المناسبة باستخدام خوارزمية شجرة القرار (Decision Tree) .

7- Kusnia, U. (2022). Analisis sentimen review aplikasi berita online pada google play menggunakan metode Algoritma Naive Bayes Classifier dan Support Vector Machines (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).

8- اسامه إبراهيم, صلاح الدوه جي & ,سهيل الحمود. (2022). اكتشاف البيانات الشاذة في تدفقات انترنت الأشياء دراسة استقصائية وتوجهات مستقبلية .مجلة جامعة دمشق للعلوم الهندسية 38 , (4- خاص معلوماتية).

9- Zanaty, E. A. (2012). Support vector machines (SVMs) versus multilayer perception (MLP) in data classification. Egyptian Informatics Journal, 13(3), 177-183.

10- Kramer, O., & Kramer, O. (2016). Scikit-learn. Machine learning for evolution strategies, 45-53.

11- Oliphant, T. E. (2006). A guide to NumPy (Vol. 1, p. 85). USA: Trelgol Publishing.

12- Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... & Yu, T. (2014). scikit-image: image processing in Python. PeerJ, 2, e453.