|  |  |  |
| --- | --- | --- |
| **Syrian Arab Republic** |  | **الجمهورية العربية السورية** |
| **Ministry of Higher Education** | **وزارة التعـليــم العـالـــــــــي** |
| **Syrian Virtual University** | **الجامعة الافتراضية السورية** |
| **MSC** |  | **ماستر علوم الحاسب** |

**عنوان المشروع:**

**التنبؤ بمرض القلب بالاعتماد على خوارزميات التنقيب عن البيانات**

**Heart Disease Prediction**

**إعداد الطالب:**

**زين العابدين موريس موسى**

**Zain\_alabideen\_242399@svuonline.org**

**إشراف الدكتور:**

**عصام سلمان**

**T\_isalman@svuonline.org**

**جدول المحتويات**

[المقدمة: 5](#_Toc203306029)

[الكلمات المفتاحية: 5](#_Toc203306030)

[أولا: فكرة موسعة عن المشروع: 6](#_Toc203306031)

[ثانيا: اهداف المشروع: 7](#_Toc203306032)

[ثالثا: المتطلبات الوظيفية: 8](#_Toc203306033)

[رابعا: المتطلبات غير الوظيفية: 8](#_Toc203306034)

[خامسا: آفاق مستقبلية للمشروع: 10](#_Toc203306035)

[سادسا: مراحل عمل المشروع: 10](#_Toc203306036)

[سابعا: الادوات البرمجية المستخدمة في عمل المشروع: 11](#_Toc203306037)

[ثامنا: الدراسات المرجعية: 12](#_Toc203306038)

[الفصل الاول: تحليل النظام والمنهجية المقترحة: 15](#_Toc203306039)

[**1.1.** **النموذج المقترح:** 15](#_Toc203306041)

[**1.2.** **الأدوات البرمجية:** 16](#_Toc203306042)

[**1.3.** **منهجية جمع البيانات ومعالجتها:** 16](#_Toc203306043)

[**1.3.1.** **جمع البيانات Collect data:** 16](#_Toc203306044)

[**1.3.2.** **مجموعات البيانات قبل المعالجة:** 17](#_Toc203306045)

[**1.3.3.** **استخراج المعرفة knowledge extraction:** 17](#_Toc203306046)

[**1.3.4.** **تنظيف البيانات data cleaning:** 18](#_Toc203306047)

[**1.4.** **النماذج تعلم الالة المستخدمة:** 19](#_Toc203306048)

[**1.4.1.** **نموذج آلة المتجه الداعم :Support Vector Machine:** 19](#_Toc203306049)

[**1.4.2.** **نموذج شجرة القرار Decision Trees model:** 21](#_Toc203306050)

[**1.4.3.** **نموذج الجار الأقرب KNN model:** 21](#_Toc203306051)

[**1.4.4.** **نموذج الانحدار الخطي Linear Regression model:** 22](#_Toc203306052)

[**1.5.** **التقييم:** 23](#_Toc203306053)

[**1.6.** **التصنيف والتنبؤ:** 25](#_Toc203306054)

[الفصل الثاني: تحديد متطلبات المشروع وتصميم الواجهات: 26](#_Toc203306055)

[**2.1.** **قائمة المتطلبات:** 26](#_Toc203306057)

[**2.1.1.** **نمذجة حالات الاستخدام:** 27](#_Toc203306058)

[**2.1.2.** **متطلبات التصميم Front End Requirements:** 28](#_Toc203306059)

[**2.1.3.** **متطلبات البرمجة Back End Requirements:** 28](#_Toc203306060)

[**2.1.4.** **البيئة البرمجية:** 29](#_Toc203306061)

[**2.1.5.** **المكتبات المستخدمة:** 32](#_Toc203306062)

[**2.2.** **تصميم الواجهة التفاعلية:** 34](#_Toc203306063)

[**2.2.1.** **بناء صفحة الويب** 34](#_Toc203306064)

[**2.2.2.** **تحديد معمارية المشروع** 34](#_Toc203306065)

[الفصل الثالث: التطوير والتنجيز: 36](#_Toc203306066)

[**3.1.** **جمع البيانات:** 36](#_Toc203306068)

[**3.2.** **معالجة البيانات:** 37](#_Toc203306069)

[**3.3.** **توزيع الصفوف:** 38](#_Toc203306070)

[**3.4.** **تدريب النماذج:** 42](#_Toc203306071)

[**3.4.1.** **تقسيم البيانات:** 42](#_Toc203306072)

[**3.4.2.** **تطبيق بيانات التدريب على النماذج:** 43](#_Toc203306073)

[**3.5.** **تقييم النماذج:** 43](#_Toc203306074)

[**3.5.1.** **تقييم نموذج الانحدار اللوجستي:** 43](#_Toc203306075)

[**3.5.2.** **تقييم نموذج شجرة القرار:** 47](#_Toc203306076)

[**3.5.3.** **تقييم نموذج SVC:** 50](#_Toc203306077)

[**3.5.4.** **تقييم نموذج الجار الأقرب:** 53](#_Toc203306078)

[**3.5.5.** **مقارنة النماذج:** 56](#_Toc203306079)

[**3.6.** **حفظ النماذج:** 56](#_Toc203306080)

[**3.7.** **بناء واجهة التطبيق:** 58](#_Toc203306081)

[الفصل الرابع: الاختبار والتقييم: 60](#_Toc203306082)

[**4.1.** **اختبار المنهجية المقترحة:** 60](#_Toc203306087)

[**4.2.** **اختبار التنفيذ:** 60](#_Toc203306088)

[الفصل الخامس: الخلاصة والافاق المستقبلية: 63](#_Toc203306089)

[**5.1.** **الخلاصة والتوصيات:** 63](#_Toc203306091)

[**5.2.** **الافاق المستقبلية:** 64](#_Toc203306092)

[المراجع: 65](#_Toc203306093)

**جدول الاشكال:**

[**الشكل رقم 1-1 نموذج التنبؤ المقترح** 13](#_Toc189662915)

[**الشكل رقم ‏2‑1 مخطط حالات الاستخدام** 24](#_Toc189662916)

[**الشكل رقم ‏2‑2 أداة Jupyter** 27](#_Toc189662917)

[**الشكل رقم ‏2‑3 أداة Jupyter ضمن المتصفح** 27](#_Toc189662918)

[**الشكل رقم ‏2‑4 انشاء مستند** 28](#_Toc189662919)

[**الشكل رقم ‏2‑5 واجهة مستند jupyter** 29](#_Toc189662920)

[**الشكل رقم ‏2‑6 بنية المشروع** 32](#_Toc189662921)

[**الشكل رقم ‏3‑1 مجموعة البيانات** 34](#_Toc189662922)

[**الشكل رقم ‏3‑2 مجموعة البيانات بعد المعالجة** 35](#_Toc189662923)

[**الشكل رقم ‏3‑3 تحديد الدخل** 35](#_Toc189662924)

[**الشكل رقم ‏3‑4 تحديد الخرج** 36](#_Toc189662925)

[**الشكل رقم ‏3‑5 توزع العينات** 37](#_Toc189662926)

[**الشكل رقم ‏3‑6 توزع العينات بعد إعادة توزيع الصفوف** 38](#_Toc189662927)

[**الشكل رقم ‏3‑8 مصفوفة الارتباك لنموذج الانحدار اللوجستي** 42](#_Toc189662928)

[**الشكل رقم ‏3‑9 منحني ROC لنموذج الانحدار اللوجستي** 43](#_Toc189662929)

[**الشكل رقم ‏3‑10 مصفوفة الارتباك لنموذج شجرة القرار** 45](#_Toc189662930)

[**الشكل رقم ‏3‑11 منحني ROC لنموذج شجرة القرار** 46](#_Toc189662931)

[**الشكل رقم ‏3‑12 مصفوفة الارتباك لنموذج SVC** 48](#_Toc189662932)

[**الشكل رقم ‏3‑13 منحني ROC لنموذج SVC** 49](#_Toc189662933)

[**الشكل رقم ‏3‑14 مصفوفة الارتباك لنموذج الجار الاقرب** 51](#_Toc189662934)

[**الشكل رقم ‏3‑15 منحني ROC لنموذج الجار الاقرب** 52](#_Toc189662935)

[**الشكل رقم ‏3‑16 حفظ النماذج** 54](#_Toc189662936)

[**الشكل رقم ‏4‑1 صفحة الويب** 58](#_Toc189662937)

[**الشكل رقم ‏4‑2 استخدام نموذج شجرة القرار لاكتشاف مرض القلب** 59](#_Toc189662938)

**الجداول:**

[**جدول رقم ‏3‑1 نتائج النماذج** 53](#_Toc189662939)

# المقدمة:

الطب هو مجال علاجي يشمل الممرضين والأطباء بمجالات مختلفة، ويغطي التشخيص، والعلاج، والوقاية من الأمراض، والبحوث الطبية، والعديد من الجوانب الأخرى للصحة، ويهدف إلى تعزيز الصحة والحفاظ عليها، وهي تنطوي عن استخدام العقاقير أو الجراحة، وغالبًا ما تدعمها تدابير المشورة ونمط الحياة، وتشمل الأنواع البديلة والتكميلية من الطب الوخز بالإبر، والمعالجة المثلية، والأدوية العشبية، والعلاج بالفن، والطب الصيني التقليدي، وغيرها الكثير

إنّ التطور السريع الآخذ بالتزايد والحاجة الملحة إلى استخدام الحاسوب في حياتنا اليومية لتسهيل الكثير من المهام التي تطلبت فيما مضى الكثير من الوقت والجهد وصل إلى مجال الطب، الأمر الذي جعل المجال الطبي يعتمد بشكل ملح على أجهزة الكمبيوتر لتقديم أكثر أشكال التشخيص العلاج تقدمًا للمرضى، فقد أصبح من الممكن مشاهدة تأثير الحاسوب في العديد من المجالات الطبية، مثل أنظمة التصوير والمختبرات السريرية وتخزين السجلات الطبية وعمليات المستشفيات وغيرها من المجالات الطبية الأخرى.

# الكلمات المفتاحية:

|  |
| --- |
| * **Bayes algorithms**   هي عائلة من المصنفات الاحتمالية البسيطة على أساس تطبيق نظرية بايز مع افتراضات استقلالية (ساذجة) قوية بين الميزات.   * **KNN algorithms**   تقوم خوارزمية KNN بعملية تصنيف نقطة الاختبار test point اعتمادا على نقاط التدريب المحيطة بها، أي الجيران الأقرب لنقطة الاختبار لذلك سميت هذه الخوارزمية neighbors nearest   * **data mining**   تنقيب البيانات هو مجال يجمع عدة تقنيات من عدة تخصصات مثل تعليم الآلة، الإحصاء، التعرف على الأنماط، قواعد البيانات والإظهار المرئي، يهدف هذا العلم إلى استخراج المعلومات من البيانات الضخمة المخزنة والمجمعة في المستودعات |

# أولا: فكرة موسعة عن المشروع:

عمل الباحثون منذ سنوات عديدة للبحث عن مسببات الأمراض وكيفية النجاة منها، وقد برز مؤخراً اثر العلوم المعلوماتية على المجال الطبي، وخاصة الدور الذي لعبته تقانات الذكاء الصنعي في حل المشاكل الطبية ومن بينها خوارزميات التنقيب عن البيانات في العلوم الطبية.

وقد تمكنت التكنولوجيا من توفير الات جديدة وأدوية وعلاجات ساعدت في حماية العديد من الأشخاص وحسنت من فرص علاجهم، ليس هذا وحسب، بل استطاعت التكنولوجيا تحسين الأبحاث والدراسات العلمية لجعل الرعاية الصحية والتشخيص أكثر كفاءة.

وبالتالي فان السبب الرئيسي الذي يجعل استخراج المعلومات هو محور الاهتمام في العلوم الطبية ومن بينها أمراض القلب هي توافر كميات ضخمة من البيانات والحاجة الملحة لاستخراج المعلومات والمعرفة منها. واستخراج البيانات هو تكييف أو استخراج المعرفة من مجموعة من البيانات ، ويمكن أن يعزى استخراج البيانات إلى التطور الطبيعي لتكنولوجيا المعلومات ، والذي هو نتيجة تطور آلية بناء قواعد البيانات ، مثل: عمليات جمع البيانات وإنشاء قاعدة البيانات.

وان مرض القلب يعتبر وصف لنطاق من الأمراض التي تصيب القلب. وتشمل الأمراض التي تندرج تحت طائلة مرض القلب أمراض الأوعية الدموية، مثل مرض الشريان التاجي، والمشاكل في نظم القلب (اضطراب نظم القلب) وعيوب القلب التي يولد المرء بها (عيوب القلب الخلقية) من بين أمراض أخرى.

وتعتمد الفحوص الواجب إجرائها لتشخيص مرض القلب على الحالة التي يعتقد الطبيب أن الشخص مصاب بها. بصرف النظر عن مرض القلب الذي قد يكون مُصابًا به، من المرجح أن يجرى الطبيب فحصًا جسديًا وأن يسأل عن التاريخ الطبي للمريض قبل إجراء أي فحوصات. بجانب اختبارات الدم والأشعة السينية على الصدر.

ونظراً لأهمية الكشف المبكر عن وجود مشكلة صحية في الجسم وتحديد ماهيتها والذي يلعب دوراً هاماً في سرعة العلاج وعاملاً مؤثر في الشفاء وهنا تكمن أهمية الحاجة إلى التشخيص الدقيق والسريع لوجود المرض وتحديد درجة الحالة المرضية.

وبالإضافة الى الحاجة عن الكشف المبكر هنالك حاجة الى تشخيص المرض بشكل صحيح وسريع اعتمادا على بيانات المريض واعطائه نتائج واضحة في هذا المجال وانطلاقا من هذه الاسباب نسعى لبناء نظام قادر ذكي باستخدام خوارزميات التنقيب عن البيانات قادر على تشخيص الاصابة بأمراض القلب بالاعتماد على مجموعة كبيرة من البيانات التي سوف نعمل على جمعها من خلال الجولات على بعض المراكز الصحية والمستشفيات ذات العلاقة وبالتالي تحديد الحالة المرضية للمريض

# ثانيا: اهداف المشروع:

تأتي الأمراض القلبية الوعائية في صدارة أسباب الوفيات في جميع أنحاء العالم، ذلك أنّ عدد الوفيات الناجمة عن هذه الأمراض يفوق 17.7 مليون نسمة، ممّا يمثّل 17 % من مجموع وفيات العالم. كان لابد من الاهتمام بهذا الجانب للمساهمة في تشخيص وعلاج هذا النوع من الأمراض

يهدف عملنا على جمع عدد كبير من البيانات الأولية المبنية على التحاليل والتقارير الطبية المُوصفة لحالة المرضى ثم العمل على إعدادها وترتيبها لنعمل على دراستها بالاعتماد على خوارزميات التنقيب عن البيانات لاستخراج المعلومات ومقارنة النتائج فيما بينها والهدف تحسين تشخيص أمراض القلب والتنبؤ بحالة المريض بشكل سليم وبالتالي تحسين عملية بناء القرار .

ويمكن تلخيص أهداف البحث بما يلي:

* تشخيص مرض القلب بالاعتماد على خوارزميات التنقيب عن البيانات
* تطوير خوارزميات التعلم الآلي في مجال أمراض القلب ومعالجتها
* المساهمة في التشخيص الطبي السليم باستخدام خوارزميات التنقيب عن البيانات
* بناء نظام ذكي قادر على تشخيص حالة المريض من خلال المعرفة المكتشفة
* تصميم وبناء تطبيق ويب يعمل وفق خوارزميات التنقيب عن البيانات في التعامل مع مرضى القلب

# ثالثا: المتطلبات الوظيفية:

ان المتطلبات الوظيفية لهذا المشروع يمكن إنجازها من خلال شيفرة برمجية للتنبؤ بمرض القلب عرض نتائج الفحص والتشخيص على شكل مخططات بيانية تظهر احتمالات الاصابة وخطأ التصنيف ودراسة نظرية شاملة لأدوات التنقيب عن البيانات والخوارزميات المراد استخدامها ضمن التطبيق كخوارزمية الاحتمال الشرطي BAYES وخوارزمية الجار الاقرب KNN.

وسوف يتم الاعتماد في عملية التنقيب على مجموعة من بيانات التدريب لحالات من المصابين بمرض القلب حيث تحتوي عينة التدريب والتنبؤ على سجلات بمقدار 200 سجل بحالتين مختلفتين تتراوح قيمها بين النتيجة الايجابية والنتيجة السلبية وتحتوي هذه العينة على بارامترات محددة تمثل هذه البارامترات السمات العامة للمرض

# رابعا: المتطلبات غير الوظيفية:

بالنسبة لمستخدمين هذا المشروع يجب أن تتوفر لديهم المتطلبات التالية:

# خامسا: آفاق مستقبلية للمشروع:

* هناك العديد من الإضافات التي يمكن إضافتها إلى هذا المشروع والتي قد تسمح له بأن يصبح أكثر احترافية وعملية. انطلاقًا من الحاجة إلى البيانات الإضافية اللازمة لتدريب النماذج بشكل أفضل.
* من الممكن إضافة خاصية وضع صفحة الويب على الاستضافة الإلكترونية وربطها بالنظام الصحي الإلكتروني ليتمكن الشخص من استخدام التطبيق بشكل تشاركي وسريع.
* وهناك ميزة اخرى ايضا وهي توسيع عمل التطبيق بحيث نقوم ببناء موقع كامل لاحد المستشفيات وامكانية تقديم الطلبات وادارتها من خلال هذا الموقع ومن ثم استخدام النموذج الافضل الذي اخترناه في عملية الكشف عن مرض القلب بشكل آلي ومباشر دون الرجوع للأطباء المسؤولين مما يساهم في توفير الوقت والجهد في عملية دراسة الحالة الصحية المقدمة من المريض.

# سادسا: مراحل عمل المشروع:

1. **اجراء دراسة مرجعية عن خوارزميات التنقيب المقترحة**
2. **دراسة العينات المقترحة وتصديرها الى قاعدة البينات**
3. **تحديد حالات الاستخدام للموقع وتجهيز الدارسة التحليلية والمخططات اللازمة**
4. **اجراء الدراسة التصميمية لقاعدة البينات ورسم المخططات اللازمة**
5. **تصميم الواجهات**
6. **البدء ببرمجة الموقع وكتابة الأكواد الخاصة بالخوارزميات المطلوبة**

# سابعا: الادوات البرمجية المستخدمة في عمل المشروع:

# ثامنا: الدراسات المرجعية:

توجه العديد من الباحثين في الآونة الأخيرة لاستخدام تقانات الذكاء الصنعي وأدواتها في المجال الطبي ومن العلوم الطبية التي شغلت الكثيرين من العلماء أمراض القلب الوعائية

* قام الباحث Ameer H. Ali وآخرون عام 2018 ببناء شبكة عصبونية ذكية الهدف منها تشخيص الأمراض السرطانية بالإعتماد على ست ميّزات هامة تم استخلاصها من الصور الشعاعية المأخوذة للمريض بمرض السرطان ومعالجتها بعمليات حسابية لتحويلها لدخل جيد للشبكة العصبونية المصممة. الهدف من الدراسة كان الكشف عن مرض السرطان وفقا للمعايير التي وضعتها الكلية الأمريكية لأمراض الروماتيزم. استخدم الباحثون شبكة عصبونية حيث كان عدد خلايا طبقة الدخل فيها ستة وكان خطأ التصنيف يساوي 3.8968\*10^-10. عمل الباحث على التحكم بعدد الميزات وبالتالي عدد خلايا طبقة الدخل للشبكة العصبونية فاستخدم خمسة ثم ثمانية فكان خطأ دقة التصنيف 0.0041 و1.0611\*10^-10. وهنا أمكننا القول أن جميع النتائج مقبولة فالخطأ في التصنيف صغير جدا [[1]](#footnote-1)
* أما في عام 2011، قام الباحث Jyoti Soni وآخرون بتقديم مسح للتقنيات الحالية لاكتشاف المعرفة من قواعد البيانات الضخمة باستخدام تقنيات استخراج المعلومات حيث استخدم الباحثون ثلاث عشرة ميزة هامة تصف حالة وأعراض مريض القلب, وقد قام الباحثون بعدد من التجارب باستخدام خوارزمية شجرة القرار فيها فكان معدل التصنيف 89% وقد تفوقت هذه الخوارزمية في أدائها على خوارزمية تصنيف بايز والتي كانت دقة التصنيف فيها 86.53 %. بينما كانت دقة التصنيف في خوارزمية الجار الأقرب 85.53 %[[2]](#footnote-2).
* وفي دراسة للباحث Dr. G. Karraz   واخرون عام 2006، بعنوان "التصنيف التلقائي لنبضات القلب باستخدام تصنيف الشبكة العصبية بناءً على إطار بايزيان ".حيث قام الباحث بتصميم خوارزمية للمعالجة التلقائية لإشارة مخطط كهربائية القلب (ECG) والهدف هو تصنيف دقات القلب . جمع الباحث البيانات من قاعدة البيانات MIT-BIH حيث عدد السجلات المدروسة هو / 48 / حالة مرضية تعاني من عدم انتظام ضربات القلب. درس الباحث خمسة أنواع من التصنيفات لعدم انتظام ضربات القلب وفق الخوارزمية المصممة، حيث كان النوع الأول هو ضربات التوصيل البطيني المبكرة (PVC) ، والنوع الثاني هو ضربات التوصيل الأذيني المبكر (APC) ، والنوع الثالث هو ضربات كتلة الفرع الأيمن (RBBB) ، أما النوع الرابع فهو ضربات كتلة الفرع الأيسر (LBBB) ، وآخيرا النوع الخامس هو ضربات إيقاع السرعة ( PRB) ،مضافاً الى ذلك النوع النظامي هو ضربات القلب النظامية . استخدم الباحث في مرحلة التدريب بيانات خمس سجلات وهي على التوالي (124 و214 و111 و100 و107).  تم قام بتعيين الميزات الهامة بالاعتماد على مورفولوجيا ECG وعلى الفترات الزمنية. بعد القيام بالعديد من العمليات اللازمة لتكون مناسبة لاستخدامها كبيانات تدريب الشبكة العصبونية . تم تصنيف نوع ضربات القلب بشكل يدوي بالاعتماد على الخبرة المهنية لأخصائي أمراض القلب وعلى دليل قاعدة بيانات MIT-BIH . كانت الدراسة حول تصنيف نوع المرض وهو خرج للشبكة العصبونية المصممة. توصل الباحث وفق النظام المصمم الى الحد الأدنى من الحساسية وبمعدل تصنيف 86٪ والحد الأدنى من الخصوصية بمعدل تعرف 90٪.[[3]](#footnote-3)
* وفي عام 2013، عمل الباحث Ishtake S.H على استخدام تقنيات التنقيب عن البيانات لتشخيص مرض القلب باستخدام 15 ميزة هامة استخدم الباحث في الدراسة خوارزمية بايز فكان معدل التصنيف 95% استخدم الباحث 909 حالات مرضية . كما درس 106 حالة مرضية باستخدام شجرة القرار فكان معدل التصنيف 94.93% بينما عندما استخدم شبكة عصبونية لدراسة 432 حالة مرضية فكان معدل التصنيف 93.54%.[[4]](#footnote-4)
* قدم الباحث Nidhi Bhatla وآخرون عام, 2012 دراسة تهتم بمقارنة نتائج تقنيات تحليل البيانات التي تم تطويرها في السنوات الأخيرة لاستخراج المعلومات المختلفة والتنبؤ بأمراض القلب, استخدم الباحثون كمدخلات للشبكات العصبونية 15 ميزة هامة حيث بلغت دقة التنصيف في حالة الشبكات العصبونية 100% وكانت هذه النبة الأعلى في التصنيف اي كانت الأفضل بين الدراسات مقارنة بخوارزمية Naive Bayes والتي بلغت دقة التصنيف فيها 90.74% وخوارزمية Decision Treesحيث كانت دقة التنصيف فيها 99.62% [[5]](#footnote-5).

وأخيراً، توضح الدراسات السابقة أهمية دمج المعلومات الناتجة عن الفحوصات السريرية مع بيانات سجلات المرضى المجمعة من التقارير الطبية السابقة والحالية والصور الشعاعية والتحاليل الطبية لبناء القرارات الطبية السليمة مبنية على المعرفة التراكمية وهذا الأسلوب يمكن أن يقلل عدد الأخطاء الطبية ، وتعزيز سلامة المرضى ، وتقليل تباين الممارسات غير المرغوب فيها أو القرارات الخاطئة. فقد أظهرت الدراسات السابقة أن استخدام تقنيات التنقيب عن البيانات أعطت نتائج واعدة في التنبؤ أو تصنيف الأمراض بشكل عام وخاصة أمراض القلب الوعائية.

# الفصل الاول: تحليل النظام والمنهجية المقترحة:

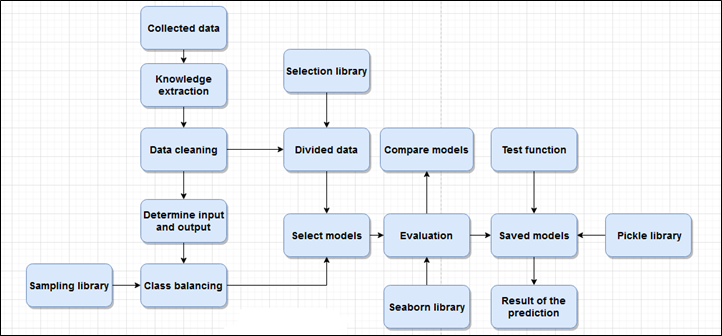
1. 1. **النموذج المقترح:**

**في هذا المشروع ، يمكن اعتماد منهجية خاصة للوصول إلى الهدف المنشود. في البداية ، سنعمل على جمع البيانات لعملية التصنيف والتنبؤ ، وبعد ذلك سنعمل على معالجة البيانات الأولية يدويًا ، للانتقال إلى المرحلة التالية ، وهي إجراء مجموعة من العمليات لتنظيف البيانات ، وملء القيم الفارغة ، وتحويل القيم النصية إلى قيم رقمية.**

**بعد ذلك تتم عملية اختيار حزمة من نماذج التصنيف والتي ستمر بدورها بمرحلتين: التدريب ثم اختبار وقياس القيم الناتجة عن طريق تطبيق معايير التقييم المعتمدة مثل الدقة وعدد مرات الإجابة الصحيحة وعدد مرات الإجابة الخاطئة.**

**حتى نصل أخيرًا إلى إنشاء مجموعة من المصنفات القادرة على إجراء عملية التنبؤ بدقة تامة ويمكن استخدامها لاحقًا عن طريق إدخال المعلومات وإعطائنا النتيجة المطلوبة وفقًا لعملية الإدخال.**

**تتم هذه المراحل كما ذكرنا سابقًا من خلال عدة خطوات ، ويوضح الشكل (1-4) مخطط تدفق عملية التنبؤ والمنهجية المستخدمة في هذا المشروع:**



**الشكل رقم 1-1 نموذج التنبؤ المقترح**

* 1. **الأدوات البرمجية:**

**عند تطوير التطبيق ، تم اختيار Python لتكون لغة البرمجة الأساسية ، وتحديداً Python الإصدار 3.6 لأنه الإصدار الأكثر استقرارًا وفقًا لتوثيق المكتبات الفردية التي يُراد استخدامها (TensorFlow ، 2020).**

**يرجع السبب وراء استخدام Python إلى النطاق الواسع من الدعم لمكتبات التعلم الآلي وأطر العمل الضرورية للتعلم الآلي ، مثل مكتبة "Sklearn" وSeaborn وPickle.**

* 1. **منهجية جمع البيانات ومعالجتها:**
     1. **جمع البيانات Collect data:**

تُعرف عملية جمع البيانات الكمية والنوعية حول متغيرات معينة بهدف تقييم النتائج أو اكتساب نظرة ثاقبة يمكن تنفيذها باسم جمع البيانات. إجراء واضح المعالم.

ومع ذلك ، قد يكون إنشاء هذا الإجراء صعبًا ، فهو يتضمن تقييم الأهداف ، وتحديد متطلبات البيانات ، واختيار طريقة لجمع البيانات ، وتنظيم خطة جمع البيانات التي تجمع بين أهم مكونات البرنامج.

* + 1. **مجموعات البيانات قبل المعالجة:**

يجب أن تحتوي مجموعة البيانات المناسبة لهذا المشروع على عدد من الميزات القوية المفيدة في عملية التنبؤ والتصنيف التي تعكس بيانات الإدخال ، ويجب أن تحتوي هذه الميزات على مجموعة من القيم وفقًا لمعايير محددة مثل العمر والجنس وعدد من معلومات الحالة الصحية مثل معدل التدخين ومعدل التلوث البيئي ومعدل تعاطي الكحول .. إلخ. تحتوي معظم الميزات على نطاق من القيم التي تعبر عن القيمة المتوسطة

في الوقت نفسه ، يوجد عمود يمثل الناتج المرغوب من خلال عملية التنبؤ ، على سبيل المثال ، حدوث المرض. قيم هذا العمود في نطاق مجموعة البيانات بين نعم أو لا.

* + 1. **استخراج المعرفة knowledge extraction:**

تتضمن مرحلة جمع البيانات لاستخراج المعرفة عملية البحث داخل كميات كبيرة من البيانات للكشف عن العلاقات التي لم يتم كشفها سابقا بين عناصر البيانات؛ والمعروفة أيضا باسم اكتشاف المعرفة في قواعد البيانات (KDD) وبالتالي هي النشاط الذي يقوم باستخراج المعلومات المتواجدة في كميات كبيرة من البيانات، بهدف البحث عن أنماط معرفية واكتشاف الحقائق الخفية الواردة في قواعد البيانات.

كما تتضمن عملية جمع البيانات ايضا تحليل البيانات المتواجدة في قواعد البيانات باستخدام الأدوات التي تبحث عن الاتجاهات أو البيانات التي لا معنى لها، واستخراج معلومات ضمنية، لم تكن معروفة سابقا، ويمكن أن تكون مفيدة.

وتتم عملية استخراج أنماط معرفية من مجموعات البيانات الكبيرة من خلال الجمع بين الأساليب من الإحصاءات والذكاء الاصطناعي مع إدارة قواعد البيانات وصولا للهدف المراد وهو اكتشاف المعرفة داخل قواعد البيانات (KDD) حيث تستخدم أساليب التحليل مثل: الشبكات العصبية Neural Networks أو الخوارزميات الجينية Genetic Algorithms أو شبكة القرارات Decision Trees والتقنية والنماذج الهجينة Hybrid Models لتحديد الأنماط والعلاقات في مجموعات البيانات.

وتتم عملية اكتشاف المعرفة في قواعد البيانات عن طريق تحديد الأنماط والاتجاهات في البيانات التي تم جمعها باستخدام طرق مختلفة مثل: التصنيف Classification أو التحليل التسلسلي Sequential analysis أو العنقدة (التجميع) clustering أو قواعد الارتباط Association Rule.

* + 1. **تنظيف البيانات data cleaning:**

يُعد تنظيف البيانات عملية أساسية في تجهيز البيانات الأولية لتطبيقات تعلّم الآلة وتطبيقات ذكاء الأعمال (BI). قد تحتوي البيانات الأولية على العديد من الأخطاء، ما قد يؤثر في دقة نماذج تعلّم الآلة ويؤدي إلى تنبؤات غير صحيحة ويؤثر سلبًا في الأعمال.

تتضمن الخطوات الأساسية لتنظيف البيانات تعديل حقول البيانات غير الصحيحة وغير المكتملة وحذفها، والتعرُّف على المعلومات المكررة والبيانات بعيدة الصلة وحذفها، وتصحيح أخطاء التنسيق والقيم المفقودة والأخطاء الإملائية.

تستلزم عملية تنظيف البيانات عدة خطوات لتحديد إدخالات المشكلة وإصلاحها. الخطوة الأولى هي تحليل البيانات لتحديد الأخطاء. قد يتضمن ذلك استخدام أدوات التحليل النوعي التي تعتمد على القواعد والأنماط والقيود لتحديد القيم غير الصالحة. الخطوة التالية هي إزالة الأخطاء أو تصحيحها.

**تتضمن الخطوات الشائعة لتنظيف البيانات علاج كل من:**

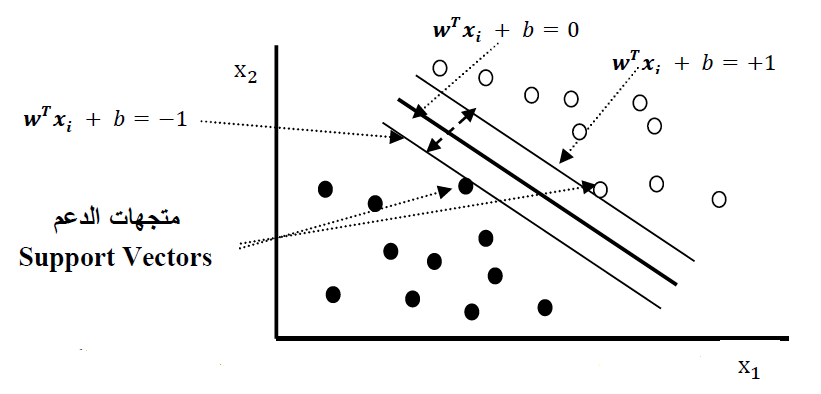
* البيانات المكررة: إسقاط المعلومات المكررة.
* البيانات غير ذات الصلة: تحديد الحقول المهمة لإجراء تحليل معين وإسقاط البيانات غير ذات الصلة من التحليل.
* القيم المتطرفة: يمكن أن تؤثر القيم المتطرفة جذريًا في أداء النموذج، لذا تُحدد القيم المتطرفة والإجراء المناسب.
* البيانات المفقودة: تمييز البيانات المفقودة بعلامة وإسقاطها أو تنسيبها.
* الأخطاء الهيكلية: تصحيح الأخطاء المطبعية وغيرها من التناقضات، وجعل البيانات متوافقة مع نمط أو اصطلاح شائع.
  1. **النماذج تعلم الالة المستخدمة:**

هناك العديد من نماذج التعلم الآلي التي يمكن استخدامها في هذا المشروع ، وقد اخترنا مجموعة من النماذج التي أثبتت نجاحها في مجال التنبؤ الطبي والصحي.

* + 1. **نموذج آلة المتجه الداعم :Support Vector Machine:**

يعتمد نموذج أدلة المتجه الداعم على عدد من المعلمات الأساسية مثل المستوى الفاصل Hyperplane ومضاريب لاكرانج Lagrange multipliers التي تؤثر على نحو كبير على دقة عملية التصنيف اذ ان البيانات في فضاء الادخال Input Space تصنف على وفق الانموذج الرياضي الاتي:

W: متجهة الاوزان Weights, X تمثل متجهة الادخال Input Vector, b تمثل قيمة التحيز Bias وd تمثل قيمة الإخراج كما يمكن ملاحظة معادلات الحدود في المستوى من الشكل التالي:



**الشكل رقم 2-1 معادلات الحدود في المستوى**

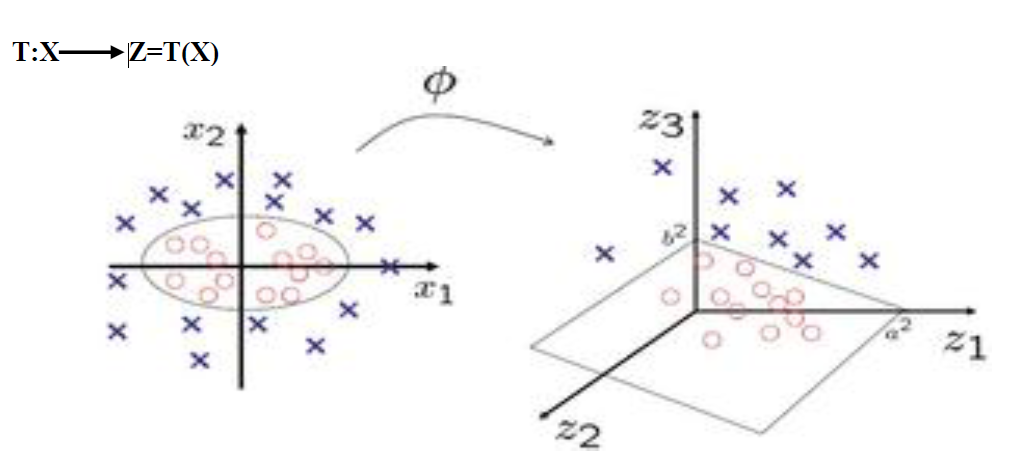
من الشكل السابق نلاحظ ان معادلة المستوي الفاصل Hyper plane تكتب بالشكل الاتي:

ان البيانات القريبة أو التي تقع على حدود الحد الفاصل تسمى متجهات الدعم أو المساندة Support vectors كما يمكن حساب المسافة بين النقاط في المستوى ومعادلة المستوى الفاصل من خلال العلاقة التالية:

وبعد عدد من الإجراءات والتحويلات الرياضية، يتم إيجاد قيم كل من متجه الاوزان المثالي والتحيز المثالي ومن بعدها تحسب دالة التنصيف الاتية:

اذن ان تمثل الوزن المثالي , تمثل قيمة التحيز المثالية وsgn تمثل القرار النهائي لانتماء (x) لاحد الأصناف Classes

في بعض الحالات لا يمكن فصل البيانات بوساطة الحد الفاصل Hyperplane لذلك يفضل استخدام تقنية Kernel trick ، التي تقوم بتحويل بيانات المدخلات من الخطية إلى بيانات ذات مساحة ابعاد (بيانات غير خطية) ويكون التحويل الجديد قابلاً لمفصل الخطي بوساطة الحد الفاصل Hyperplane ويبين الشكل التالي كيفية تحويل البيانات من الخطية الى غير خطية باستخدام Kernel Trick



**الشكل رقم 3-1 تقنية Kernel trick**

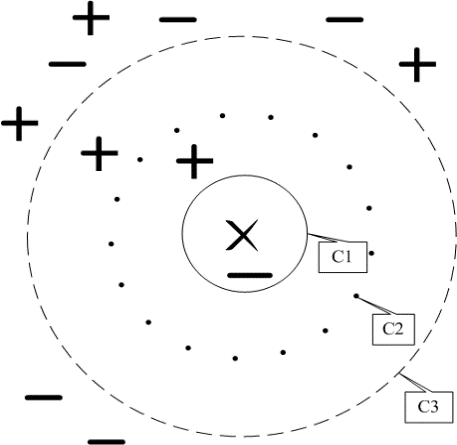
* + 1. **نموذج شجرة القرار Decision Trees model:**

يمكن استخدام نموذج شجرة القرار Decision Trees model للتنبؤ بمرض القلب ، وذلك عن طريق تحليل العوامل المختلفة التي تؤثر على احتمالية الإصابة بهذا المرض، مثل التدخين والتعرض للملوثات البيئية والوراثة والعمر والجنس والتاريخ الطبي للشخص. يتم تحليل هذه العوامل وتصنيفها في شجرة قرارية تساعد على تحديد المعطيات الأكثر أهمية في التنبؤ بالمرض ، مما يساعد على اتخاذ قرارات أكثر دقة في التشخيص والعلاج.

* + 1. **نموذج الجار الأقرب KNN model:**

على خوارزمية الجار الأقرب (K-Nearest Neighbor) التي تعتبر من التقنيات التنبؤية التي تهدف للتنبؤ بقيمة الفئة عن طريق مقارنة السجلات الشبيهة بالسجل المراد التنبؤ بقيمته وتقدير القيمة المجهولة لهذا السجل بناء على مقدار تلك السجلات. يعتمد عمل هذه الخوارزمية بشكل أساسي على عدد الحالات الأقرب (K) ووحدة القياس (metric) حيث يمثل الرمز (K) عدد الحالات المماثلة أي إيجاد مجموعة جزئية مؤلفة من (K) عنصر الأكثر تشابهاً مع الحالة المراد التنبؤ بقيمتها, وفي حال تماثل بعض أعضاء هذه المجموعة الجزئية يتم الاعتماد على وحدة القياس بغية تحديد العنصر الأكثر تشابهاً.

الشكل (4-4) يوضح آلية عمل هذه الخوارزمية حيث تظهر النقطة المجاورة الأقرب لإحدى نقاط البيانات المراد تصنيفها (X) ضمن الحد الفاصل (C1) بينما يظهر ضمن الحد الفاصل (C2) النقطتين المجاورتين للنقطة (X) وضمن الحد الفاصل (C3) النقاط الثلاثة المجاورة للنقطة (X).



**الشكل رقم 4-1 توزع البيانات ضمن المصنف KNN**

* + 1. **نموذج الانحدار الخطي Linear Regression model:**

معادلة النموذج للخوارزمية والتي ستحدد لنا كيفية التنبؤ للبيانات المستقبلية بالشكل التالي:

حيث:

* (h (xi: الفرضية hypothesis التي تستند عليها الخوارزمية.
* B: معامل الانحدار Regression Coefficient.
* Xi: البيانات التي لدينا.

لأننا نحتاج أن يكون مُخرج output ألخوارزمية على شكل احتمالية، فهذا يعني يجب أن تكون القيم محصورة بين (0,1)، لذلك سنستعين بدالة Sigmoid والتي ستكون صيغتها الرياضية بالشكل التالي:

لذلك سيكون تعريف الاحتمالية في النموذج الذي لدينا والذي سيوفر احتمالية التصنيف للفئتين (01،) بالشكل التالي:

ويتم اختصار هاتين المعادلتين بمعادلة رياضية واحدة لتكون بالشكل التالي:

والشيء المهم في هذه المرحلة هي أيجاد دالة الخطأ Error Function أو تسمى في بعض الأحيان Cost Function حيث نستخدم مع خوارزمية Logistic Regression نوع من دوال الخطأ تسمى أقصى تقدير للاحتمال maximum likelihood estimation (MLE) والتي تكون صيغتها الرياضية بالشكل التالي:

وهدفنا من استخدام دالة الخطأ هو حساب مقدار الخطأ الذي ينتج عن التنبؤ الخاطئ أثناء عملية التصنيف Classification، فكلما كان مقدار الخطأ قليل أثناء عملية التدريب training كلما كانت عملية التصنيف لدينا جيدة وحيث تمثل قيمة B معامل الانحدار Regression Coefficient الذي يمكن التحكم فيه لمحاولة تقليل مقدار الخطأ لدينا.

* 1. **التقييم:**

في التصنيف الثنائي، هناك عدد من المصطلحات التي ستتكرر عند التحدث عن تجريب النموذج الذكي، منها:

**True Positives (TP):** عند توقع النموذج ل 1, ووجود 1 فعلي حقيقي للبيانات

**True Negatives (TN):** عند توقع النموذج ل 0, ووجود 0-حقيقي للبيانات

**False Positives (FP):** 0عند توقع النموذج ل 1, لكن القيمة الحقيقية للبيانات تكون

**False Negatives (FN):** 1**ع**ند توقع النموذج ل 0, لكن القيمة الحقيقية للبيانات تكون

* **Precision score:**

يعرف على أنه نسبة الملاحظات الإيجابية المتوقعة بشكل صحيح **(true positives/tp)** إلى إجمالي الملاحظات الإيجابية المتوقعة.

* **Recall score**

هو النسبة **tp / tp + fn** حيث **tp هو** عدد الإيجابيات الحقيقية وfn عدد السلبيات الخاطئة.

* **F1 score**

هو مقياس لدقة النموذج في مجموعة البيانات. ... تعتبر أيضًا طريقة للجمع بين الاحكام **(Precision)** والاسترجاع **(Recall)** النموذج، ويتم تعريفها على أنها الوسط التوافقي لدقة النموذج واسترجاعه ويعطى بالعلاقة:

* **Accuracy score**

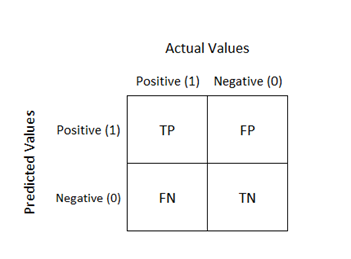
الدقة هي مقياس واحد لتقييم نماذج التصنيف. يتم شرحه أيضًا على أنه جزء من التنبؤات التي حصل عليها نموذجنا بشكل صحيح رياضيا، الدقة لها التعريف التالي: **الدقة = عدد التنبؤات الصحيحة/ العدد الإجمالي للتنبؤات.**

بالنسبة للتصنيف الثنائي، يمكن أيضًا حساب الدقة من حيث الإيجابيات والسلبيات على النحو التالي:

* **Confusion Matrix**

هي مصفوفة ثنائية الأبعاد nxn، حيث أن n هو عدد الأصناف المتوقعة، المصفوفة مهمتها قياس أداء لمشكلة تصنيف التعلم الآلي حيث يمكن أن يكون الناتج فصلين أو أكثر، جدول يحتوي على 4 مجموعات مختلفة من القيم المتوقعة والفعلية:

تستخدم Confusion Matrix لقياس Precision، accuracy، recall بطريقة مصورة وسهلة الفهم، ويوضح الشكل (6-4) ذلك:



**الشكل رقم 5-1 مصفوفة الارتباك**

يمثل كل صف من المصفوفة نقاط البيانات في فئة متوقعة، بينما يمثل كل عمود الحالات الموجودة في فئة فعلية (أو العكس). ينبع الاسم من حقيقة أنها تجعل من السهل معرفة ما إذا كان النظام يخلط بين فئتين أو صنفين.

* 1. **التصنيف والتنبؤ:**

يمكن استخدام طرق التعلم الآلي المذكورة أعلاه للتنبؤ بمرض القلب ، فعلى سبيل المثال:

* + طريقة الخوارزميات الجيدة لإنشاء نماذج تعلم الآلة التي تستخدم في التنبؤ بمرض القلب، وذلك عن طريق تحليل البيانات الطبية المتعلقة بالأشخاص المصابين والأشخاص الأصحاء.
  + طريقة الشبكات العصبونية التكرارية لتحليل التسلسلات الزمنية المتعلقة بالأشخاص المصابين ، مثل نتائج فحوصات الأشعة السينية والتحاليل الطبية.
  + طريقة الخوارزميات الجينومية لتحليل البيانات الجينومية المتعلقة بالأشخاص المصابين ، وذلك لتحديد الجينات المسؤولة عن هذا المرض والعلاقات بينها.
  + طريقة التعلم العميق لتحليل الصور الطبية المتعلقة بالأشخاص المصابين ، وذلك لتحديد المعالم المهمة في الصور وتشخيص المرض بدقة أكبر.
  + طريقة الخوارزميات الجيوماتيكية لتحليل الصور الجوية المتعلقة بالمناطق التي تشهد ارتفاعًا في معدلات إصابة السكان بهذا المرض، وذلك لتحديد العوامل البيئية والجغرافية التي تؤثر على انتشار المرض.

# الفصل الثاني: تحديد متطلبات المشروع وتصميم الواجهات:

في هذا القسم سيتم التوسع فيه عن طريق أخذ النتائج المقدمة وتحويلها إلى متطلبات مفصلة لهذا المشروع، بهدف توفير قائمة مبررة بالمتطلبات وتقييمها باعتبارها حاسمة في تلبية الأهداف العامة للمشروع. وسيناقش هذا القسم أيضا عملية تصميم المشروع، وتفاصيل قرارات التصميم المتخذة والأدوات التي يراد استخدامها لبناء المشروع وكذلك ايضت تقديم مبررات شاملة وراء هذه القرارات في كل خطوة.

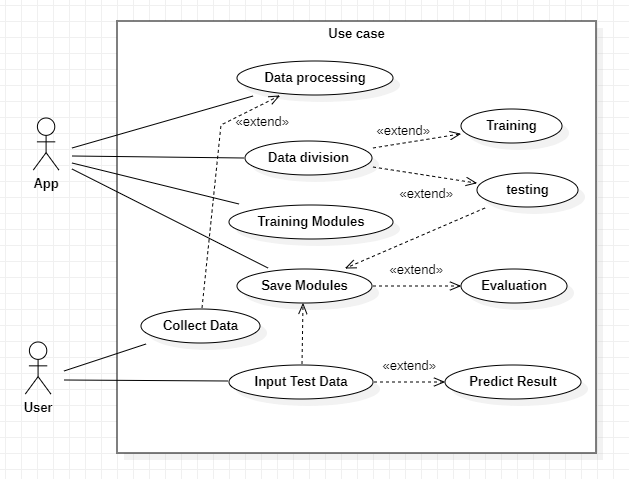
1. 1. **قائمة المتطلبات:**

سيتم اعتماد المتطلبات المحددة في هذا القسم مع مراعاة الهدف العام للمشروع والذي يتمثل في تطوير تطبيق لكشف مرض القلب باستخدام تقنيات التعلم الآلي. ولتحديد المتطلبات للنظام وتفصيل كيفية تنفيذها وتشغيلها، سيتم استخدام مجموعة من تقنيات النمذجة والتي تسمى "نمذجة حالة الاستخدام" و "نمذجة قصة المستخدم".

ثم سنحدد المتطلبات الوظيفية وغير الوظيفية، وبعد ذلك سنشرح طريقة استخدام محرر الكود وواجهة بيئة البرنامج، حيث من خلالها سيتم تنفيذ التعليمات وتنظيمها كواجهة تفاعلية تتيح للمستخدم استخدامها بسهولة، حيث اعتمدنا على بيئة أناكوندا ومحرر الكود Jupyter، كما ذكرنا سابقاً.

* + 1. **نمذجة حالات الاستخدام:**

ستتم عملية تحديد المتطلبات من خلال إنشاء مخططات حالات الاستخدام وقصص المستخدم، وذلك لتوضيح كيفية تفاعل المستخدم مع النظام وتحديد المتطلبات المختلفة في كل جزء من النظام. ويتضمن هذا الجزء نوعين من الفاعلين، الأول هو التطبيق نفسه حيث يتم تنفيذ معظم المهام من خلاله دون تدخل المستخدم الفعلي، مثل معالجة البيانات أو تدريب وحفظ النماذج. بينما يقتصر دور المستخدم الفعلي، الذي يعتبر الفاعل الثاني، على المرحلة الأخيرة وهي استخدام أحد النماذج لتوقع الحالة التي يدخلها ويوضح توضح الشكل (1-5) مخططات حالات الاستخدام.



**الشكل رقم ‏2‑1 مخطط حالات الاستخدام**

* + 1. **متطلبات التصميم Front End Requirements:**
* **المتطلبات الوظيفية:**
* واجهة تطبيق الويب التي يمكن للمستخدمين من خلالها إدخال البيانات والتفاعل معها.
* وضع التنبؤ الذي يعرض للمستخدمين الإدخال ونتيجة التنبؤ.
* القدرة على وضع عناصر الإدخال المتعددة مثل الأرقام والنصوص حيث يمكن للمستخدمين استخدام هذه العناصر لإدخال البيانات.
* أن يظهر تعليق المستخدم في حالة حدوث خطأ في الجزء البرمجي من التطبيق.
* **المتطلبات الغير وظيفية:**
* أن يكون التنقل إلى الصفحة فوريًا.
  + 1. **متطلبات البرمجة Back End Requirements:**
* **المتطلبات الوظيفية:**
* أن تكون الطريقة المستخدمة لقراءة البيانات وتحريرها في التطبيق مناسبة وتستخدم مكتبات مناسبة.
* أن يكون التطبيق قادرًا على تقسيم البيانات إلى بيانات تدريب واختبار.
* أن يكون التطبيق قادرًا على توازن الاصناف.
* أن يكون التطبيق قادرًا على تدريب النماذج وحفظها باستخدام مكتبات خاصة.
* أن يكون التطبيق قادرًا على توقع القيم التي يدخلها المستخدم وإعطاء النتيجة المتوقعة. ويجب أن يكون التنقل إلى الصفحة فوريًا.
* **المتطلبات الغير وظيفية:**
* يجب تدريب النماذج على كمية كبيرة من البيانات تشمل 80٪ على الأقل من مجموعة البيانات الكلية.
* السرعة في عملية التدريب وحفظ النماذج.
* الحصول على معايير دقة عالية في نهاية عملية التدريب.
* عرض النتيجة بوضوح للمستخدم وفقًا للشرط المدخل
  + 1. **البيئة البرمجية:**
* **Anaconda3:**

هي عبارة عن توزيعة بايثون خاصة تأتي مجانية أو مدفوعة الأجر (freemium) هدفها الرئيسي هو تسهيل إدارة مكتبات بايثون معينة ومحددة. المكتبات التي نتحدث عنها في بايثون اناكوندا هي مكتبات مختصة بمعالجة البيانات الضخمة، وتطبيقات الحوسبة العلمية وغيرها من المكتبات العلمية. تُعتبر توزيعة Anaconda من توزيعات البايثون المشهورة في مجال Data Science، حيث تجد كثيرًا من المقالات العلمية تتحدث عنها عند استخدام البايثون في معالجة البيانات وتعليم الآلة.

* **الحاجة الى الاستخدام:**

هنالك عدة أسباب تجعلنا في حاجة الى استخدام توزيعة بايثون اناكوندا وهي:

1. عندما يزداد عدد المكتبات التي نتعامل معها (وخاصة عندما نتحدث عن مجال معالجة وتحليل البيانات) سوف نجد أنفسنا أمام كم كبير من المكتبات والتوابع التي تحتاج لترتيب، وهنا يأتي دور بايثون اناكوندا.
2. بالتجربة، لو أردنا تثبيت بعض المكتبات مثل مكتبة Scipy، وخاصة في نظام Windows، فإن أسهل طريقة للحصول على هذه المكتبة وإستخدامها هو من خلال بايثون اناكوندا، والسبب في ذلك يرجع إلى وجود متطلبات كثيرة يجب تثبيتها وإعدادها قبل تثبيت مكتبة Scipy.
3. نستطيع تفعيل وإستخدام ما يزيد عن 720 مكتبة بايثون مختصة بال Data Visualization، Machine Learning، Deep Learning وBig Data وذلك بمجرد تثبيت اناكوندا.
4. تُقدم اناكوندا في مجال تحليل البيانات الضخمة وDeep Learning إمكانية الوصول لحزم ومكتبات مثل theano، tensorflow، neon، h2o، keras بسهولة كبيرة.

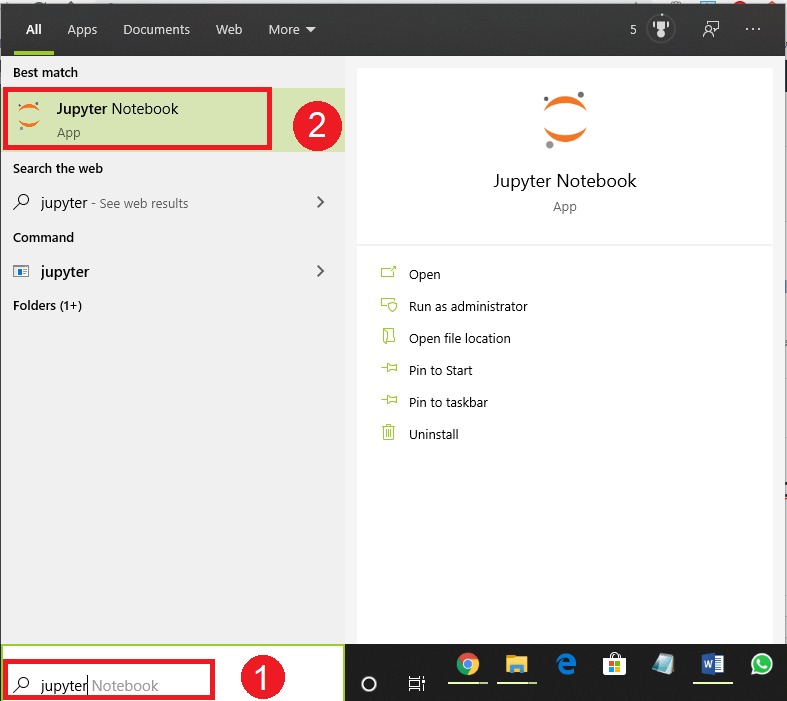
* **Jupyter:**

تُعد أداة Jupyter Notebook أحد أشهر وأهم الأدوات المُستخدمة أثناء تحليل البيانات. بالإضافة إلى ذلك، هي شيء أساسي لا يستغني عنها عالم البيانات، وذلك لما تُقدمه من مميزات وخصائص تُسهل من التعامل مع البيانات والشيفرة البرمجية. تُعتبر Jupyter أداة قوية يُمكن استخدامها تفاعليًا في مشاريع علم البيانات وتعليم الآلة.

* **تثبيت الأداة**

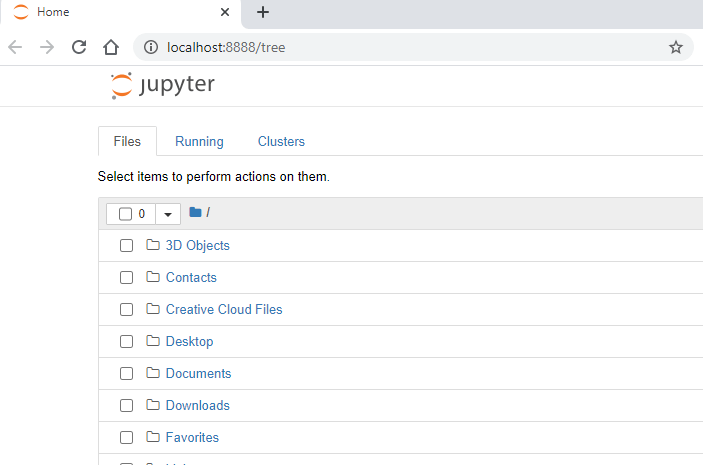
قبل البدء بتثبيت أداة Jupyter لابد من تثبيت البايثون حيث يتطلب تثبيت الأداة أن تكون نُسخة البايثون هي 3.3 فما فوق أو بايثون 2.7، لذا يجب التأكد من هذه النقطة قبل البدء

بعد التثبيت يُمكن تشغيل الأداة في نظام Windows عبر قائمة إبدا بكتابة اسم الأداة، حيث ستظهر في القائمة:



**الشكل رقم ‏2‑2 أداة Jupyter**

عند تشغيل الأداة، سيتم تشغيل المتصفح والذهاب للرابط http://localhost:8888/tree والذي يعني أن الأداة يتم تشغيلها عبر جهاز الحاسوب محليًا. ستظهر نافذة مُشابهة للتالي:



**الشكل رقم ‏2‑3 أداة Jupyter ضمن المتصفح**

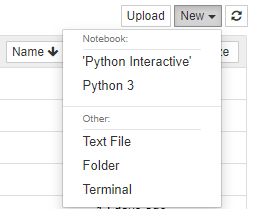
النافذة في الصورة السابقة هي لوحة تحكم أداة Jupyter والتي من خلالها يُمكننا إنشاء مستندات والتعامل معها. ومن المهم معرفة أن الأداة تُتيح لنا التعامل مع الملفات والمُجلدات الموجودة في المسار التشغيلي للأداة Start-Up directory عند تشغيلها بالطريقة السابقة.

في العادة سنقوم بتشغيل الأداة في مجلد العمل الخاص بنا والذي في الغالب يختلف عن المسار التشغيلي التلقائي للأداة. يتم ذلك من خلال الذهاب لمسار المجلد الذي نُريد العمل فيه وتنفيذ أمر التشغيل:

Jupyter notebook

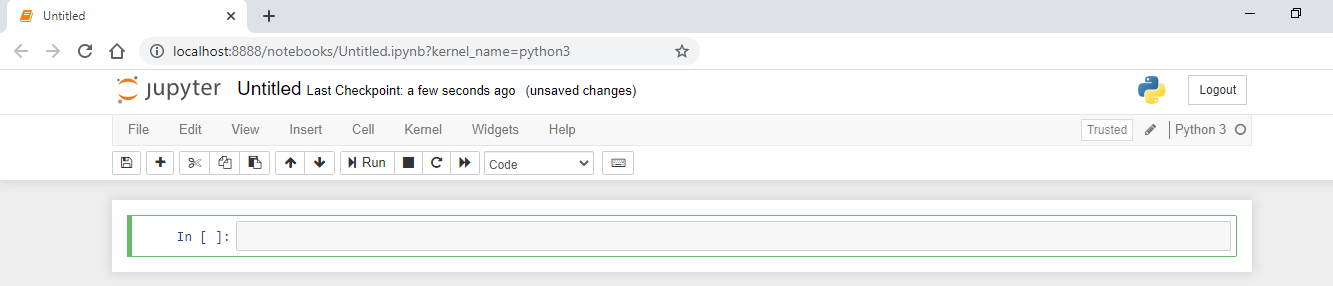
* + **إنشاء مُستند Jupyter**

تُعتبر أداة Jupyter والمستندات التي يتم انشاءها باستخدامها عبارة عن صفحات انترنت يتم تشغيلها عبر البايثون. هذا الأمر يُتيح تعامل أسهل وأكثر مرونة مع المستندات ومشاركتها. لإنشاء مستند جديد، نضغط على زر New وستظهر لنا قائمة، نختار منها Python 3:



**الشكل رقم ‏2‑4 انشاء مستند**

ستظهر صفحة جديدة عبارة عن المستند الذي سنكتب فيه الشيفرة البرمجية وعمليات تحليل البيانات كما في الصورة التالية، وسيكون المستند باسم Untitled.ipynb:



**الشكل رقم ‏2‑5 واجهة مستند jupyter**

نلاحظ أن الخلية الأولى في المُستند مُحاطة بإطار لونه أخضر، وهذا يعني أن الخلية الحالية جاهزة لإدخال الشيفرة بداخلها، وهي في وضع التحرير Edit Mode.

وان الامتداد الخاص بمستند جوبيتر هو ipynb واسم الامتداد هو اختصار Interactive Python Notebook. من المهم أن نعرف هنا أن المستند عبارة عن ملف نصي يحتوي تفاصيل ومكونات المستند مكتوبةً بصيغة Json، لذا يُمكننا التعديل على النص مباشرةً

* + 1. **المكتبات المستخدمة:**
* **NLTK Library:**

هي مكتبة مفتوحة المصدر في لغة البايثون لمعالجة اللغة الطبيعية (NLP). تحتوي هذه المكتبة على العديد من الوظائف والفئات المختلفة، التي تساعد الباحثين في علوم اللغة ومحللي البيانات على معالجة النصوص وتحليلها بطريقة تمكنهم من مساعدة أصحاب الشركات الكبيرة على اتخاذ عدد من القرارات المهمة. (تنقيب النصوص)

* **Imbalanced-Learn Library:**

هي وحدة بايثون لتوازن البيانات التي تعاني من التحيز أو الانحراف في صالح فئات معينة. وبالتالي، تساعد على إعادة عينات الفئات التي تم تخطيها أو تخفيضها في الماضي. يتم توجيه الإخراج نحو الفئة التي لديها المزيد من الأمثلة إذا كان معدل المعدل أعلى.

* **Sklearn Library:**

مكتبة scikit-learn هي مكتبة لتعلم الآلة في لغة Python، وتحتوي على العديد من الخوارزميات والأساليب المستخدمة في مجال تعلم الآلة، مثل التصنيف والتجميع والانحدار، بالإضافة إلى استخدامها في مرحلة معالجة البيانات وتقييم النماذج. تم بناؤها بناءً على مكتبات Scipy وNumpy وMatplotlib وغيرها من المكتبات الأخرى.

تم إطلاق مكتبة scikit-learn في عام 2007 خلال برنامج Google Summer of Code من قبل مجموعة من المتطوعين، وتأتي في المرتبة الثانية في موقع GitHub الشهير في قائمة مكتبات تعلم الآلة، وتعتبر مدخلاً سهلاً لأي شخص يريد تعلم وتطبيق تعلم الآلة وعلوم البيانات.

تركز مكتبة scikit-learn على نمذجة البيانات ولا تركز على الطريقة التي يتم بها تحميل البيانات وتلاحظها وتلخيصها، وهذا هو دور بشكل رئيسي لـ Pandas وNumpy.

* **Seaborn Library:**

تعتبر مكتبة seaborn مكتبة متخصصة في تحليل ودراسة البيانات، حيث توفر واجهة عالية المستوى لرسم الرسوم الإحصائية الجذابة والمفيدة.

تم بناؤها باستخدام matplotlib، حيث تعتبر أساسها، ولديها القدرة على التعامل مع الهياكل المدعومة بواسطة pandas، وبالتالي تعتبر واحدة من أهم المكتبات المستخدمة في تصور البيانات.

توفر الليونة اللازمة للتفاعل مع البيانات الموجودة وكانت seaborn داعمة ومكملة لمهام matplotlib.

* **Pickle Library:**

يمكن تسلسل وإلغاء تسلسل أي كائن باستخدام مكتبة Python pickle، حيث يتم تحويل الكائنات إلى تيار شخص يحتوي على جميع المعلومات اللازمة لإعادة تجميع الكائن في نص Python مختلف.

* 1. **تصميم الواجهة التفاعلية:**

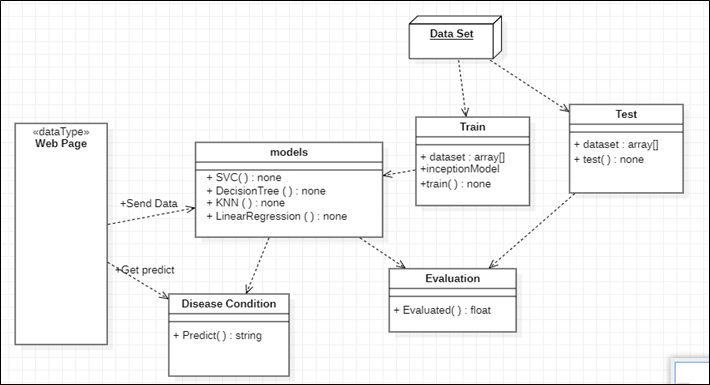
بعد تحديد متطلبات النظام بشكل صريح، أصبح من السهل الآن التخطيط لتصميم التطبيق وكيفية ربط المتطلبات المختلفة مع بعضها البعض داخل النظام. يتم ذلك من خلال إنشاء رحلات المستخدمين ورسومات UML والأسلاك، التي سيتم تنفيذها في سلسلة من العمليات السريعة. كما ذكر في الفصل 3، فإن استخدام منهجية التطوير السريع مهم لهذا المشروع، حيث ستكون هناك الكثير من التجارب فيما يتعلق بكيفية تطوير النظام. وبالتالي، فإن بعض التصاميم التي تم إنشاؤها في هذا القسم ستكون أكثر من دليل بدلاً من نموذج مفصل نهائي.

* + 1. **بناء صفحة الويب**

تم استخدام حزمة Gradio لبناء واجهة التطبيق، وهي حزمة تدعم مكتبات Python وتتيح للمستخدم إنشاء صفحات ويب وربطها بتطبيقات التعلم الآلي بحيث يتم إدخال البيانات من خلال عناصر الإدخال مثل النصوص والأرقام والصور، وترسل هذه القيم من الواجهة إلى النموذج المحدد للتعلم لتنبؤ حالته ويتم إرجاع القيمة إلى الواجهة لعرض ما إذا كانت هناك أعراض للمرض أو لا.

* + 1. **تحديد معمارية المشروع**

كما ذكر في مواصفات المتطلبات، هناك نظام رئيسي واحد في التطبيق المطلوب، يتضمن واجهة الكود ووحدات التنبؤ بالمرض - ويمكن رؤية رسم بياني لذلك كما في الشكل (6-5)



**الشكل رقم ‏2‑6 بنية المشروع**

# الفصل الثالث: التطوير والتنجيز:

في هذا القسم، سنشرح مجموعة الخطوات التي مر بها التطبيق للوصول إلى الحالة النهائية في واجهة المستخدم، حيث قمنا بجمع كمية من البيانات حوالي 1000 سجل حول معلومات أعراض مرض القلب ، ثم قمنا بمعالجة وتنظيف تلك البيانات لتكون فيما بعد وضعت في إطار بيانات ليتم استكمالها عن طريق معالجة القيم داخل هذا الإطار وإعادة توزيع العناصر لتدريب النماذج في وقت لاحق على بعض تلك البيانات. أما بالنسبة للجزء الآخر، فقد استخدمناه لاختبار وتقييم تلك النماذج وإظهار دقة كل نموذج. وأخيرًا، قمنا ببناء واجهة مستخدم تفاعلية تمكِّن المستخدم من استخدام هذه النماذج وتوقع وجود أو عدم وجود الأعراض.

1. 1. **جمع البيانات:**

يحتاج أي نظام يعتمد على تعلُّم الآلة إلى كميات كبيرة من البيانات. في حالتنا، تم جمع أكثر من 2000 سجل يحتوي على معلومات حول الأشخاص الذين كان لديهم أعراض أو لم يكون لديهم أعراض. تم تخفيض وتنظيف هذه البيانات إلى 1000 سجل يحتوي على معلومات قوية تساعد في عملية التنبؤ.

يمكن قراءة محتوى هذه البيانات وعرضها عن طريق كتابة بعض التعليمات في Python، بما في ذلك المكتبات الأساسية المستخدمة في المشروع التالي

**import pandas as pd**

**import numpy as np**

**from statistics import mode**

**from imblearn.over\_sampling import SMOTE**

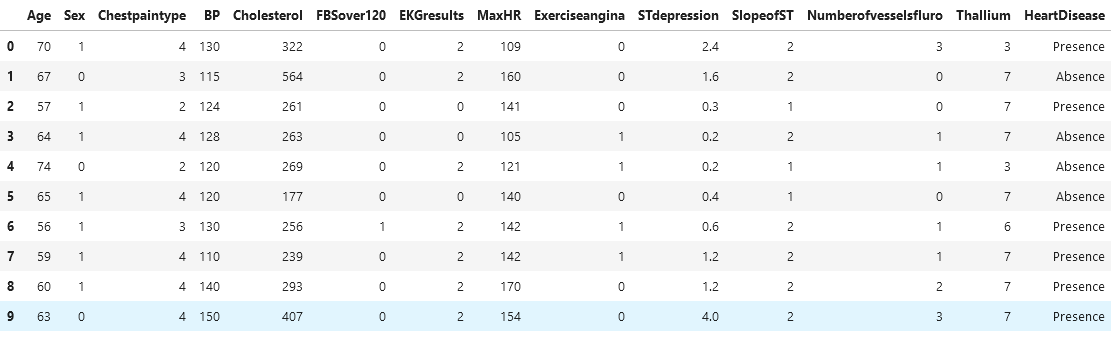
**from sklearn.datasets import make\_classification**

**from matplotlib import pyplot as plt**

**df = pd.read\_csv('Heart\_Disease\_Prediction.csv')**

**df.head(10)**

بعد تنفيذ هذه التعليمات، سيظهر جدول البيانات على النحو التالي:



**الشكل رقم ‏3‑1 مجموعة البيانات**

* 1. **معالجة البيانات:**

تعد معالجة البيانات الأولية خطوة مهمة في تعدين البيانات. فالبيانات الخام غالبًا ما تكون غير مستقرة وتحتوي على بيانات خاطئة أو غير ذات صلة. بالإضافة إلى ذلك، حتى مع البيانات المتعلقة، قد تواجه الكثير من البيانات التي يجب دمجها لتعطينا صورة أفضل. في مرحلة المعالجة المسبقة.

بعد تحديد الميزات المطلوبة للبيانات وإكمال عملية المعالجة المسبقة، يأتي الخطوة التالية وهي اختيار التنسيق المناسب الذي يجب تخزين البيانات به. يمكن تقليل عدد الميزات دون فقدان معلومات كبيرة. بالإضافة إلى ذلك، قد يحتاج المتغيرات إلى التحول لمساعدة في شرح الظاهرة التي يتم دراستها.

شملت عملية معالجة البيانات التي تم تطبيقها على مجموعة بيانات المرض عدة خطوات كما يلي:

في البداية، نقوم بتعويض جميع القيم النصية بأوزان عددية مقابلة من خلال وظيفة الاستبدال مع قيمة معامل inplace = True، حيث يتم استبدال الذكر بالرقم 1 والانثى بالرقم 2 ويتم استبدال الكلمة "نعم" في جميع الأعمدة بالقيمة 1 واستبدال كلمة "لا" بالقيمة 2 وتم تطبيق جميع الخطوات السابقة على جميع أعمدة الدخل.

أما بالنسبة لحالة مرض القلب ، والتي تمثل الخرج، فتم استبدال الكلمة "نعم" بالقيمة 1 والكلمة "لا" بالقيمة 0. نلاحظ أننا في هذا المشروع نقوم بترميز القيم الفئوية بأرقام دون استخدام labelEncoder من أجل استعراض جميع الطرق الممكنة في Python.

يمكن توضيح الخطوات السابقة من خلال الشفرة التالية:

**#Drop cloumn Patient Id**

**df.drop(df.columns[0], axis=1, inplace=True)**

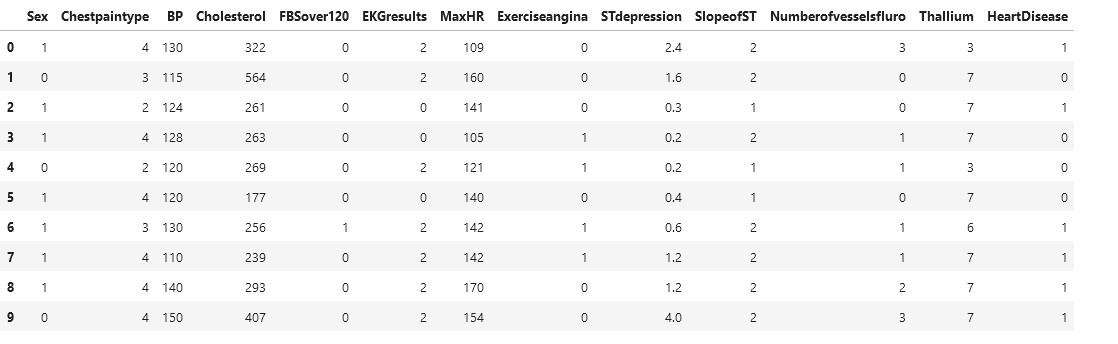
**df.replace(to\_replace="Presence",**

**value="1",inplace=True)**

**df.replace(to\_replace="Absence",**

**value="0",inplace=True)**

**df.head(10)**



**الشكل رقم ‏3‑2 مجموعة البيانات بعد المعالجة**

* 1. **توزيع الصفوف:**

تسبق مرحلة التوازن وإعادة توزيع الصفوف عملية تقسيم إطار البيانات بحيث نختار جميع الأعمدة (ما عدا حالة المرض) لتكون المدخل X وعمود حالة المرض ليكون الخرج Y كما يلي:

* + **تحديد الدخل:**

**X=df[["Sex","Chestpaintype","BP","Cholesterol","FBSover120","EKGresults","MaxHR","Exerciseangina","STdepression","SlopeofST",**

**"Numberofvesselsfluro","Thallium"]]**

**X.head()**



**الشكل رقم ‏3‑3 تحديد الدخل**

**تحديد الخرج:**

**Y=df[['HeartDisease']]**

**Y.head()**



**الشكل رقم ‏3‑4 تحديد الخرج**

بعد تقسيم الإطار إلى بيانات مدخلة ومخرجة تأتي مرحلة موازنة الصفوف والتي من الممكن أن تؤثر على عملية بناء نموذج التصنيف بهذه المشكلة في حالة عدم توازن الصفوف أي أنه إذا كان عدد الأمثلة للصف الواحد كثيرا أكبر من عدد الأمثلة لصف آخر، يمكن أن ينحي المصنف نحو الصف الذي يحمل نفس العدد من الأمثلة الأكبر.

في الكود التالي، نقوم أولاً بفحص النسب المئوية لتوزيع الصفوف في إطار البيانات الإجمالي عن طريق رسم مخطط دائري يوضح النسب المئوية لعدد الأمثلة من كل صف.

**class\_counts = Y.groupby('HeartDisease').size()**

**class\_counts.head()**

**label = ['Yes', 'No']**

**# Creating plot**

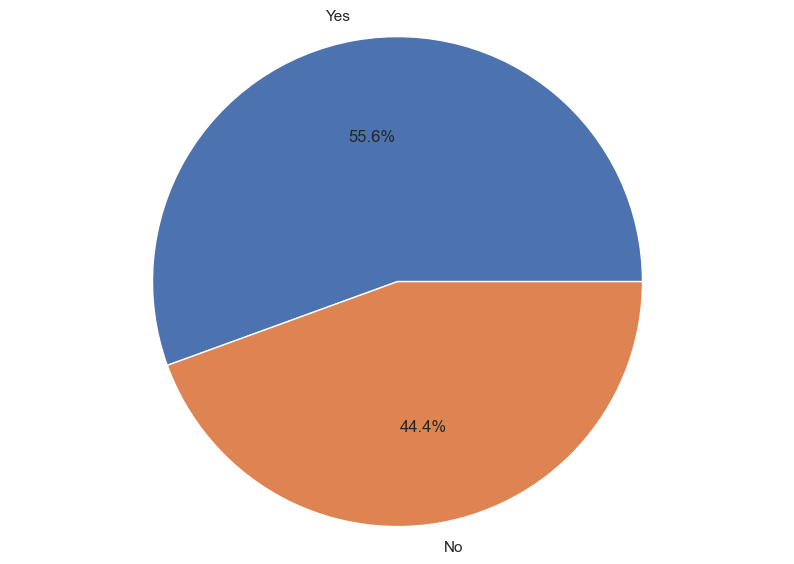
**fig = plt.figure(figsize =(10, 7))**

**plt.pie(class\_counts, labels = label, startangle=0, autopct='%1.1f%%')**

**plt.axis('equal')**

**# show plot**

**plt.show()**



**الشكل رقم ‏3‑5 توزع العينات**

الرسم السابق يوضح ضرورة موازنة الصفوف نظراً لحجم الصف (لا). لذلك، في الكود التالي، نستخدم مكتبة imblearn.over\_sampling، وذلك عن طريق إنشاء غرض من فئة SMOTE، ثم استدعاء الدالة fit\_resample، التي تضيف أمثلة جديدة لجعل الصفوف متوازنة:

**from imblearn.over\_sampling import SMOTE**

**# Oversample and plot imbalanced dataset with SMOTE**

**from collections import Counter**

**from numpy import where**

**# define dataset**

**X\_in, Y\_in = make\_classification(n\_samples=1000, n\_features=12, n\_redundant=0,n\_classes=2,**

**n\_clusters\_per\_class=2, weights=[0.7], flip\_y=0, random\_state=42)**

**# summarize class distribution**

**counter = Counter(Y\_in)**

**print(counter)**

**# transform the dataset**

**oversample = SMOTE()**

**X, Y = oversample.fit\_resample(X\_in, Y\_in)**

**# summarize the new class distribution**

**counter = Counter(Y)**

**print(counter)**

**print(Y)**

**label = ['Yes', 'No']**

**data=['1400','1400']**

**fig = plt.figure(figsize =(10, 7))**

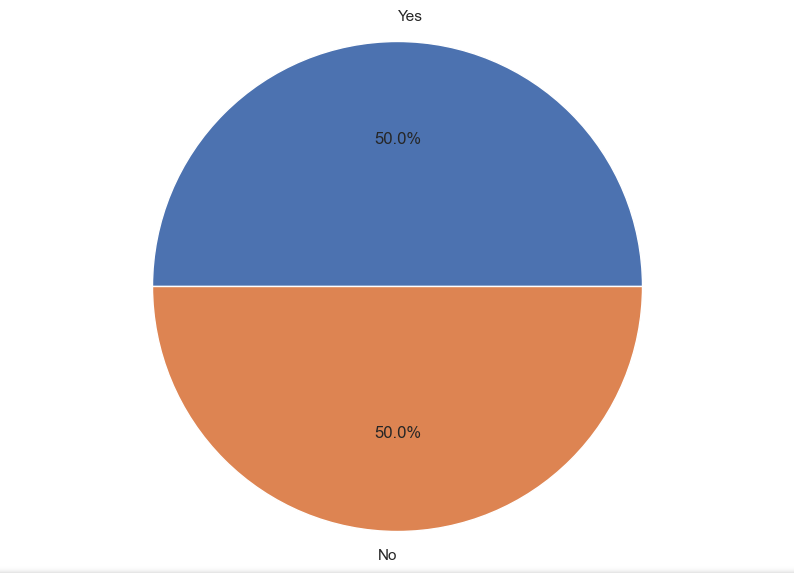
**plt.pie(data, labels =label, startangle=0, autopct='%1.1f%%')**

**plt.axis('equal')**

**# show plot**

**plt.show()**

ويكون شكل المخطط الدائري كما يلي:



**الشكل رقم ‏3‑6 توزع العينات بعد إعادة توزيع الصفوف**

* 1. **تدريب النماذج:**
     1. **تقسيم البيانات:**

في مرحلة التدريب، نقوم أولاً بتقسيم البيانات إلى قسمين. الجزء الأول يتعلق بعملية التدريب. ويمثل هذا الجزء ما بين 80 إلى 60%. أما الجزء الثاني فيستخدم لحساب معايير تقييم نموذج التعلم.

نستخدم مكتبة Sklearn.model\_selection للقيام بذلك، كما هو موضح في الكود التالي:

**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)**

نلاحظ أن الدالة Train\_test\_split والتي نمر إليها:

* X: إطار بيانات يحتوي على كافة الأعمدة باستثناء عمود حالة المرض
* Y: المصفوفة الأحادية (عمود حالة المرض)
* Test\_size: النسبة المئوية لبيانات الاختبار من إجمالي البيانات (20% افتراضيًا)

ترجع هذه الدالة:

* X\_train: بيانات تدريب تم اختيارها عشوائيًا من X
* X\_test: بيانات اختبار مختارة عشوائيًا من X
* Y\_train: فئات بيانات التدريب
* Y\_test: فئات بيانات الاختبار
  + 1. **تطبيق بيانات التدريب على النماذج:**

نقوم في عملية التدريب بإدراج مكتبة Sklearn والتي يمكن من خلالها اختيار النماذج. قمنا في هذا المشروع باختيار أربعة نماذج وهي نموذج الجار الأقرب، ونموذج الانحدار المنطقي، ونموذج شجرة القرار، ونموذج SVC. تم تكييف هذه النماذج مع بيانات التدريب من خلال الوظيفة الملائمة (X\_train, y\_train) على النحو التالي:

**from sklearn.linear\_model import LogisticRegression**

**from sklearn import tree**

**from sklearn.svm import SVC**

**from sklearn.neighbors import KNeighborsClassifier**

**knn = KNeighborsClassifier(n\_neighbors=3)**

**LogReg = LogisticRegression(solver='lbfgs', max\_iter=200)**

**dt=tree.DecisionTreeClassifier(criterion="entropy")**

**svc = SVC(probability=True)**

**for clf in (LogReg, knn, dt,svc):**

**clf.fit(X\_train, y\_train)**

**Y\_pred = clf.predict(X\_train)**

* 1. **تقييم النماذج:**
     1. **تقييم نموذج الانحدار اللوجستي:**

يمكن إظهار معايير التقييم للنموذج المتعلم من خلال حساب نتيجة نموذج Y\_pred مع بيانات اختبار X\_test، ومن ثم حساب معايير التقييم التي تقارن بيانات y\_test ونتيجة نموذج Y\_pred. يتم تطبيق هذه المراحل على كافة النماذج المحددة ضمن هذا المشروع، حيث يوضح الكود التالي تطبيق هذه الخطوات على نموذج الانحدار اللوجستي:

**from sklearn.metrics import f1\_score**

**from sklearn.metrics import accuracy\_score**

**from sklearn.metrics import precision\_score**

**from sklearn.metrics import recall\_score**

**y\_pred=LogReg.predict(X\_test)**

**print('Accuracy: %.3f' % accuracy\_score(y\_test, y\_pred))**

**print('Precision: %.3f' % precision\_score(y\_test, y\_pred,pos\_label=0))**

**print('Recall: %.3f' % recall\_score(y\_test, y\_pred,pos\_label=0))**

**print('F1: %.3f' % f1\_score(y\_test, y\_pred,pos\_label=0))**

***Accuracy: 0.889***

***Precision: 0.897***

***Recall: 0.878***

***F1: 0.887***

وتظهر معايير التقييم الناتجة كفاءة النموذج الذي تم إنشاؤه، حيث أن معظم المعايير أكبر من 75٪.

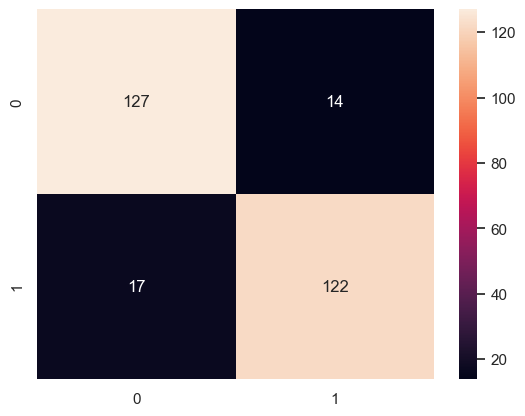
يمكن أيضًا حساب مصفوفة الارتباك باستخدام دالة confusion\_matrix، والتي نمرر إليها قيم الاختبار وقيم التنبؤ لمقارنتها، ثم نرسم تمثيل المصفوفة باستخدام دالة الخريطة الحرارية من مكتبة Seaborn كما يلي:

**from sklearn.metrics import confusion\_matrix**

**import seaborn as sns; sns.set\_theme()**

**uniform\_data = confusion\_matrix(y\_test, y\_pred, labels=[1, 0])**

**ax = sns.heatmap(uniform\_data,annot=True, fmt='g')**



**الشكل رقم ‏3‑8 مصفوفة الارتباك لنموذج الانحدار اللوجستي**

نلاحظ أن النموذج صحيح

127 TP + 122 TN = 249 cases

وأخطأ التقدير:

17 FP + 14 FN = 31 cases

يمكن أيضًا استخدام الكود التالي لرسم منحنى ROC لجهاز الاستقبال باستخدام وظيفة roc\_auc\_score وحساب AUC باستخدام وظيفة roc\_auc\_score من فئة المقاييس في sklearn:

**from sklearn.metrics import roc\_curve, roc\_auc\_score**

**import matplotlib.pyplot as plt**

**from sklearn import metrics**

**pred\_prob = LogReg.predict\_proba(X\_test)**

**# roc curve for models**

**fpr, tpr, thresh = roc\_curve(y\_test, pred\_prob[:,1])**

**# roc curve for tpr = fpr**

**random\_probs = [0 for i in range(len(y\_test))]**

**p\_fpr, p\_tpr, \_ = roc\_curve(y\_test, random\_probs)**

**auc\_score = roc\_auc\_score(y\_test, pred\_prob[:,1])**

**print('AUC: %.3f' %auc\_score)**

**plt.style.use('seaborn')**

**# plot roc curves**

**plt.plot(fpr, tpr, linestyle='--',color='orange', label='Logistic Regression')**

**plt.plot(p\_fpr, p\_tpr, linestyle='--', color='blue')**

**# title**

**plt.title('ROC curve')**

**# x label**

**plt.xlabel('False Positive Rate')**

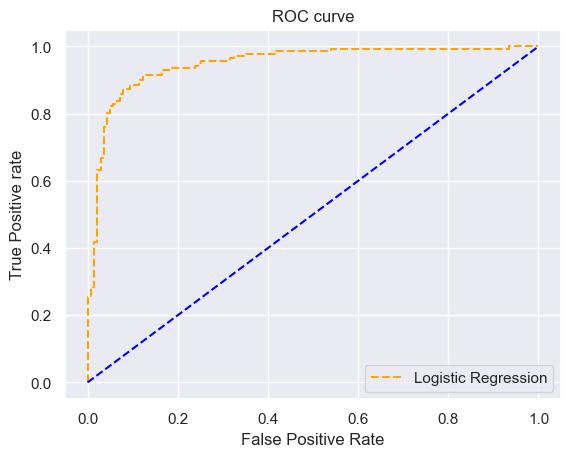
**# y label**

**plt.ylabel('True Positive rate')**

**plt.legend(loc='best')**

**plt.savefig('ROC',dpi=300)**

**plt.show();**



**الشكل رقم ‏3‑9 منحني ROC لنموذج الانحدار اللوجستي**

ونلاحظ أن المساحة تحت المنحنى تساوي 0.94 مما يعني جودة مصنف الانحدار اللوجستي

* + 1. **تقييم نموذج شجرة القرار:**

يمكن عرض معايير التقييم لنموذج شجرة القرار من خلال الكود التالي:

**y\_pred=dt.predict(X\_test)**

**print('Accuracy: %.3f' % accuracy\_score(y\_test, y\_pred))**

**print('Precision: %.3f' % precision\_score(y\_test, y\_pred,pos\_label=1))**

**print('Recall: %.3f' % recall\_score(y\_test, y\_pred,pos\_label=1))**

**print('F1: %.3f' % f1\_score(y\_test, y\_pred,pos\_label=1))**

***Accuracy: 0.911***

***Precision: 0.887***

***Recall: 0.943***

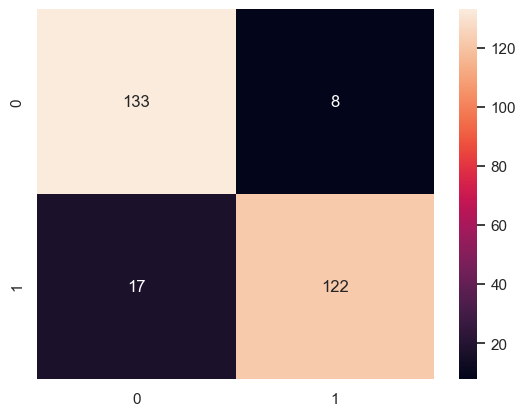
***F1: 0.914***

كما نلاحظ أن جميع معايير التقييم أعلى من 75% وبالتالي تظهر نتائج جيدة الآن نرسم مصفوفة الارتباك على النحو التالي:

**confusion\_matrix(y\_test, y\_pred)**

**uniform\_data = confusion\_matrix(y\_test, y\_pred, labels=[1, 0])**

**ax = sns.heatmap(uniform\_data,annot=True, fmt='g')**



**الشكل رقم ‏3‑10 مصفوفة الارتباك لنموذج شجرة القرار**

نلاحظ أن النموذج صحيح

133 TP + 122 TN = 255 cases

وأخطأ التقدير:

17 FP + 8 FN = 25 cases

نرسم منحنى ROC باستخدام الكود أدناه:

**pred\_prob = dt.predict\_proba(X\_test)**

**# roc curve for models**

**fpr, tpr, thresh = roc\_curve(y\_test, pred\_prob[:,1])**

**# roc curve for tpr = fpr**

**random\_probs = [0 for i in range(len(y\_test))]**

**p\_fpr, p\_tpr, \_ = roc\_curve(y\_test, random\_probs)**

**auc\_score = roc\_auc\_score(y\_test, pred\_prob[:,1])**

**print('AUC: %.3f' %auc\_score)**

**plt.style.use('seaborn')**

**# plot roc curves**

**plt.plot(fpr, tpr, linestyle='--',color='orange', label='Decision Tree')**

**plt.plot(p\_fpr, p\_tpr, linestyle='--', color='blue')**

**# title**

**plt.title('ROC curve')**

**# x label**

**plt.xlabel('False Positive Rate')**

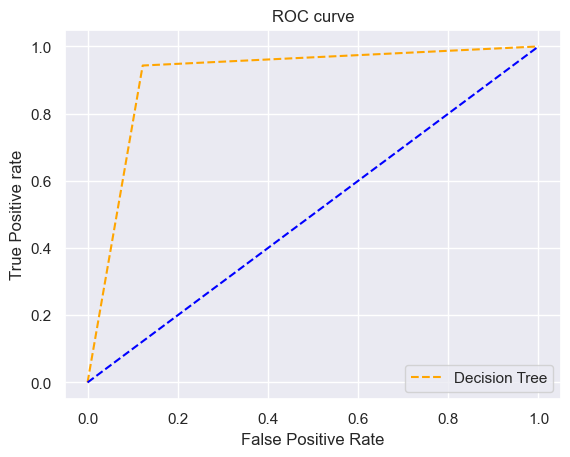
**# y label**

**plt.ylabel('True Positive rate')**

**plt.legend(loc='best')**

**plt.savefig('ROC',dpi=300)**

**plt.show();**



**الشكل رقم ‏3‑11 منحني ROC لنموذج شجرة القرار**

ونلاحظ أن المساحة تحت المنحنى تساوي 0.91، مما يعني جودة مصنف شجرة القرار

* + 1. **تقييم نموذج SVC:**

ويتم احتساب معايير التقييم لهذا النموذج من خلال الكود التالي:

**y\_pred=svc.predict(X\_test)**

**print('Accuracy: %.3f' % accuracy\_score(y\_test, y\_pred))**

**print('Precision: %.3f'**

**% precision\_score(y\_test, y\_pred,pos\_label=1))**

**print('Recall: %.3f' % recall\_score(y\_test, y\_pred,pos\_label=1))**

**print('F1: %.3f' % f1\_score(y\_test, y\_pred,pos\_label=1))**

***Accuracy: 0.907***

***Precision: 0.908***

***Recall: 0.908***

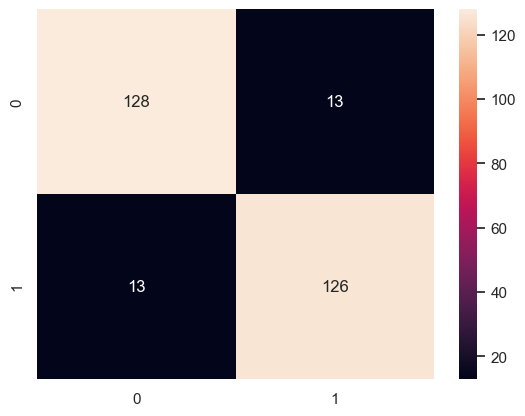
***F1: 0.908***

ونلاحظ أن جميع معايير التقييم أعلى من 75% وبالتالي تعبر عن نتائج جيدة يتم إنشاء مصفوفة الارتباك بتطبيق الكود التالي:

**confusion\_matrix(y\_test, y\_pred)**

**uniform\_data = confusion\_matrix(y\_test, y\_pred, labels=[0, 1])**

**ax = sns.heatmap(uniform\_data,annot=True, fmt='g')**



**الشكل رقم ‏3‑12 مصفوفة الارتباك لنموذج SVC**

نلاحظ أن النموذج صحيح

128 TP + 126 TN = 254 cases

وأخطأ التقدير:

13 FP + 13 FN = 26 cases

نرسم منحنى ROC باستخدام الكود أدناه:

**pred\_prob = svc.predict\_proba(X\_test)**

**# roc curve for models**

**fpr, tpr, thresh = roc\_curve(y\_test, pred\_prob[:,1])**

**# roc curve for tpr = fpr**

**random\_probs = [0 for i in range(len(y\_test))]**

**p\_fpr, p\_tpr, \_ = roc\_curve(y\_test, random\_probs)**

**auc\_score = roc\_auc\_score(y\_test, pred\_prob[:,1])**

**print('AUC: %.3f' %auc\_score)**

**plt.style.use('seaborn')**

**# plot roc curves**

**plt.plot(fpr, tpr, linestyle='--',color='orange', label='svc')**

**plt.plot(p\_fpr, p\_tpr, linestyle='--', color='blue')**

**# title**

**plt.title('ROC curve')**

**# x label**

**plt.xlabel('False Positive Rate')**

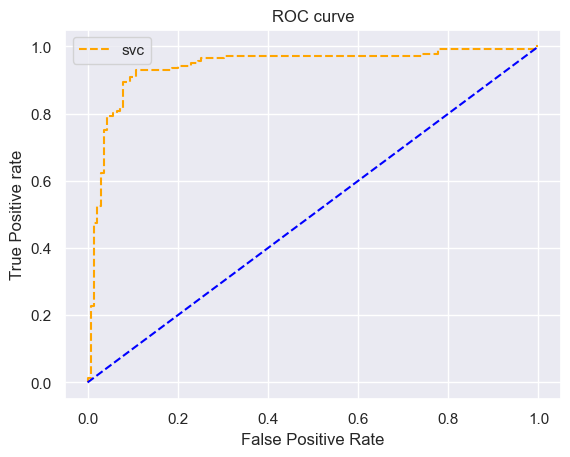
**# y label**

**plt.ylabel('True Positive rate')**

**plt.legend(loc='best')**

**plt.savefig('ROC',dpi=300)**

**plt.show();**



**الشكل رقم ‏3‑13 منحني ROC لنموذج SVC**

ونلاحظ أن المساحة تحت المنحنى تساوي 0.94 مما يعني جودة مصنف SVC

* + 1. **تقييم نموذج الجار الأقرب:**

ويتم احتساب معايير التقييم لهذا النموذج من خلال الكود التالي:

**y\_pred=knn.predict(X\_test)**

**print('Accuracy: %.3f' % accuracy\_score(y\_test, y\_pred))**

**print('Precision: %.3f' % precision\_score(y\_test, y\_pred,pos\_label=1))**

**print('Recall: %.3f' % recall\_score(y\_test, y\_pred,pos\_label=1))**

**print('F1: %.3f' % f1\_score(y\_test, y\_pred,pos\_label=1))**

***Accuracy: 0.879***

***Precision: 0.824***

***Recall: 0.965***

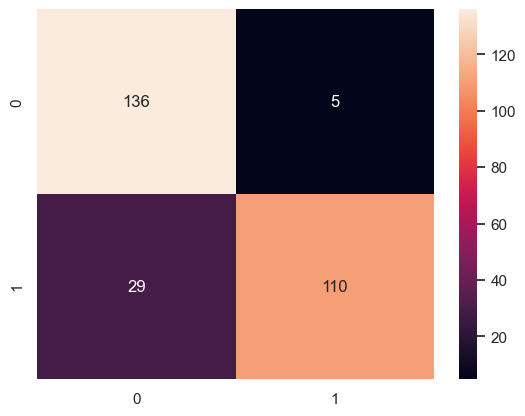
***F1: 0.889***

ونلاحظ أن جميع معايير التقييم أعلى من 75% وبالتالي تعبر عن نتائج جيدة يتم إنشاء مصفوفة الارتباك بتطبيق الكود التالي:

**confusion\_matrix(y\_test, y\_pred)**

**uniform\_data = confusion\_matrix(y\_test, y\_pred, labels=[0, 1])**

**ax = sns.heatmap(uniform\_data,annot=True, fmt='g')**



**الشكل رقم ‏3‑14 مصفوفة الارتباك لنموذج الجار الاقرب**

نلاحظ أن النموذج صحيح

136 TP + 110 TN = 246 cases

وأخطأ التقدير:

29 FP + 5 FN = 34 cases

نرسم منحنى ROC باستخدام الكود أدناه:

pred\_prob = knn.predict\_proba(X\_test)

# roc curve for models

fpr, tpr, thresh = roc\_curve(y\_test, pred\_prob[:,1])

# roc curve for tpr = fpr

random\_probs = [0 for i in range(len(y\_test))]

p\_fpr, p\_tpr, \_ = roc\_curve(y\_test, random\_probs)

auc\_score = roc\_auc\_score(y\_test, pred\_prob[:,1])

print('AUC: %.3f' %auc\_score)

plt.style.use('seaborn')

# plot roc curves

plt.plot(fpr, tpr, linestyle='--',color='orange', label='knn')

plt.plot(p\_fpr, p\_tpr, linestyle='--', color='blue')

# title

plt.title('ROC curve')

# x label

plt.xlabel('False Positive Rate')

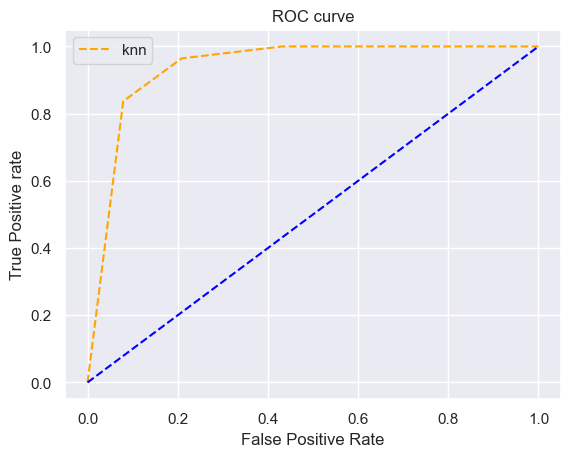
# y label

plt.ylabel('True Positive rate')

plt.legend(loc='best')

plt.savefig('ROC',dpi=300)

plt.show();



**الشكل رقم ‏3‑15 منحني ROC لنموذج الجار الاقرب**

ونلاحظ أن المساحة تحت المنحنى تساوي 0.93 مما يعني جودة مصنف KNN

* + 1. **مقارنة النماذج:**

يمكن استخدام طريقة التقييم المتبادل المذكورة أعلاه لتقييم مجموعة من نماذج التعلم. دالة cv\_comparison\_classification والتي نمرر إليها مجموعة من نماذج التعلم وجميع البيانات (X,y) وعدد حاويات السيرة الذاتية. تقوم هذه الوظيفة بإرجاع إطار بيانات يوضح مقياس الدقة لكل نموذج.

تتكرر حلقات for-loop عبر الدالة على النماذج السابقة، وتستدعي لكل نموذج دالة حسابية مشتركة من المكتبة sklearn.model والتي ترجع قائمة بقيم قياسات الصحة (كل عنصر في القائمة هو قيمة قياس صحة للاختبار حاوية).

وقد قمنا بتطبيق كل ذلك سابقاً بشكل مستقل لكل نموذج، ويمكن إحصاء نتائج الدقة من خلال الجدول 1-6 لتحديد المصنف الأفضل.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1 Score*** |
| ***Logistic Regression*** | **88.9%** | **89.7%** | **87.8%** | **88.7%** |
| ***Decision Tree*** | **91.1%** | **88.7%** | **94.3%** | **91.4%** |
| ***SVC*** | **90.7%** | **90.8%** | **90.8%** | **90.8%** |
| ***KNN*** | **87.9%** | **82.4%** | **96.5%** | **88.9%** |

**جدول رقم ‏3‑1 نتائج النماذج**

يتضح من الجدول السابق أن جميع النتائج كانت متشابهة إلى حد ما، لكن عند اختيار المصنف الأفضل ننظر إلى أعلى قيمة دقة تم الحصول عليها. وفي هذه الحالة نجد أن المصنف شجرة القرار هو أفضل مصنف للكشف عن مرض القلب لأنه يحتوي على النسبة الأعلى وهي 91.1%.

* 1. **حفظ النماذج:**

يمكن حفظ النموذج الذي تم تعلمه في ملف لاستخدامه في التنبؤ لاحقًا وليس لبناء النموذج في كل مرة. في الكود التالي نستخدم دالة التفريغ من مكتبة pickle، والتي نمرر إليها النموذج والملف الذي سيتم حفظه فيه:

**import pickle**

**LogReg\_model = 'LogReg\_model.sav'**

**pickle.dump(LogReg, open(LogReg\_model, 'wb'))**

**DecisionTree\_model = 'DecisionTree\_model.sav'**

**pickle.dump(dt, open(DecisionTree\_model, 'wb'))**

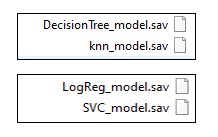
**svc\_model = 'svc\_model.sav'**

**pickle.dump(svc, open(svc\_model, 'wb'))**

**knn\_model = 'knn\_model.sav'**

**pickle.dump(knn, open(knn\_model, 'wb'))**

عند التنفيذ يتم حفظ النماذج الأربعة على النحو التالي:



**الشكل رقم ‏3‑16 حفظ النماذج**

يمكن الإعلان عن تابع مخصصة لاستخدامه للتنبؤ في كل مرة نمرر فيها المثال (قائمة يكون فيها عنصر واحد عبارة عن قائمة قيم) واسم ملف النموذج إلى هذا التابع. ترجع الدالة إما لا إذا كان التنبؤ هو 0 (وهو ترميز لا) ونعم إذا كان التنبؤ هو 1:

**def predictResult(z, model):**

**loaded\_model = pickle.load(open(model,'rb'))**

**result = loaded\_model.predict(z)**

**if result==0:**

**return ("No")**

**if result==1:**

**return ("Yes")**

* 1. **بناء واجهة التطبيق:**

قمنا في هذا القسم ببناء واجهة تفاعلية للمستخدم يمكن من خلالها إدخال القيم مباشرة دون الحاجة إلى وظيفة التنفيذ التي عرفناها سابقا. تضمين مكتبة gradio.

بعد ذلك يمكن إضافة عناصر الإدخال والإخراج وتحديد الطريقة التي سيتم تنفيذها بالضغط على زر إظهار النتيجة.

وتمت هذه الخطوات من خلال الكود التالي:

**import gradio as gr**

**def moduls(moduls\_input,Sex,Chestpaintype,BP,Cholesterol,FBSover120,EKGresults,MaxHR,Exerciseangina,STdepression,SlopeofST,Numberofvesselsfluro,Thallium):**

**with open(moduls\_input.strip()+'\_model.sav', "rb") as f:**

**clf = pickle.load(f)**

**preds = clf.predict([[Sex,Chestpaintype,BP,Cholesterol,FBSover120,EKGresults,MaxHR,Exerciseangina,STdepression,SlopeofST,Numberofvesselsfluro,Thallium]])**

**if preds == 1:**

**return "You have symptoms of Heart Disease"**

**return "You do not have symptoms of Heart Disease"**

**#Create the input component for Gradio since we are expecting 4 inputs**

**moduls\_input=gr.Textbox(label="Enter Models Name: knn or SVC or LogReg or DecisionTree")**

**Gender\_input = gr.Number(label = "Enter 1 For Male or 2 For Female")**

**Chest\_pain\_type\_input=gr.Slider(1,4,label="Enter Chest pain type")**

**BP\_input=gr.Slider(1,200,label="Enter BP Rate")**

**Cholesterol\_input=gr.Slider(1,500,label="Enter Cholesterol Rate")**

**FBS\_over\_120\_input=gr.Number(label="Enter FBS over 120 enter 0 or 1")**

**EKG\_results\_input=gr.Slider(0,2,label="Enter EKG results Rate")**

**Max\_HR\_input=gr.Slider(1,200,label="Enter Max HR Rate")**

**Exercise\_angina\_input = gr.Number(label = "Enter 0 or 1")**

**ST\_depression\_input=gr.Slider(1,10,label="Enter ST depression Rate")**

**Slope\_of\_ST\_input = gr.Number(label = "Enter 1 or 2")**

**Number\_of\_vessels\_fluro\_input = gr.Number(label = "Enter 0 or 2 or 3")**

**Thallium\_input=gr.Slider(1,10,label="Enter Thallium Rate")**

**# We create the output**

**output = gr.Textbox()**

**app=gr.Interface(fn= moduls,inputs=[moduls\_input,Gender\_input,Chest\_pain\_type\_input, BP\_input, Cholesterol\_input, FBS\_over\_120\_input, EKG\_results\_input, Max\_HR\_input, Exercise\_angina\_input, ST\_depression\_input, Slope\_of\_ST\_input, Number\_of\_vessels\_fluro\_input, Thallium\_input], outputs=output)**

**app.launch()**

# الفصل الرابع: الاختبار والتقييم:

سيناقش هذا القسم اختبار المستخدم الذي تم إجراؤه من أجل تقييم التطبيق للتحقق من تحقيق أهداف المشروع. وسيتضمن ذلك وصف استراتيجية الاختبار المستخدمة وتحليل البيانات الكمية والنوعية التي تم الحصول عليها لتحديد مدى نجاح المشروع.

4. 1. **اختبار المنهجية المقترحة:**

في البداية تم اختبار استراتيجية معالجة البيانات ووضعها بشكل مناسب للنماذج الأربعة، وتم إعادة توزيع العينات وملاءمة النماذج معها، لتدخل تلك العينات فيما بعد مرحلة اختبار الدقة وإبراز معدل التدريب الذي قامت به النماذج وقد تم تنفيذها جميعها وأظهرت دقة عالية مما يدل على نجاح اختبار استراتيجية التدريب والحصول على نماذج المتعلمين.

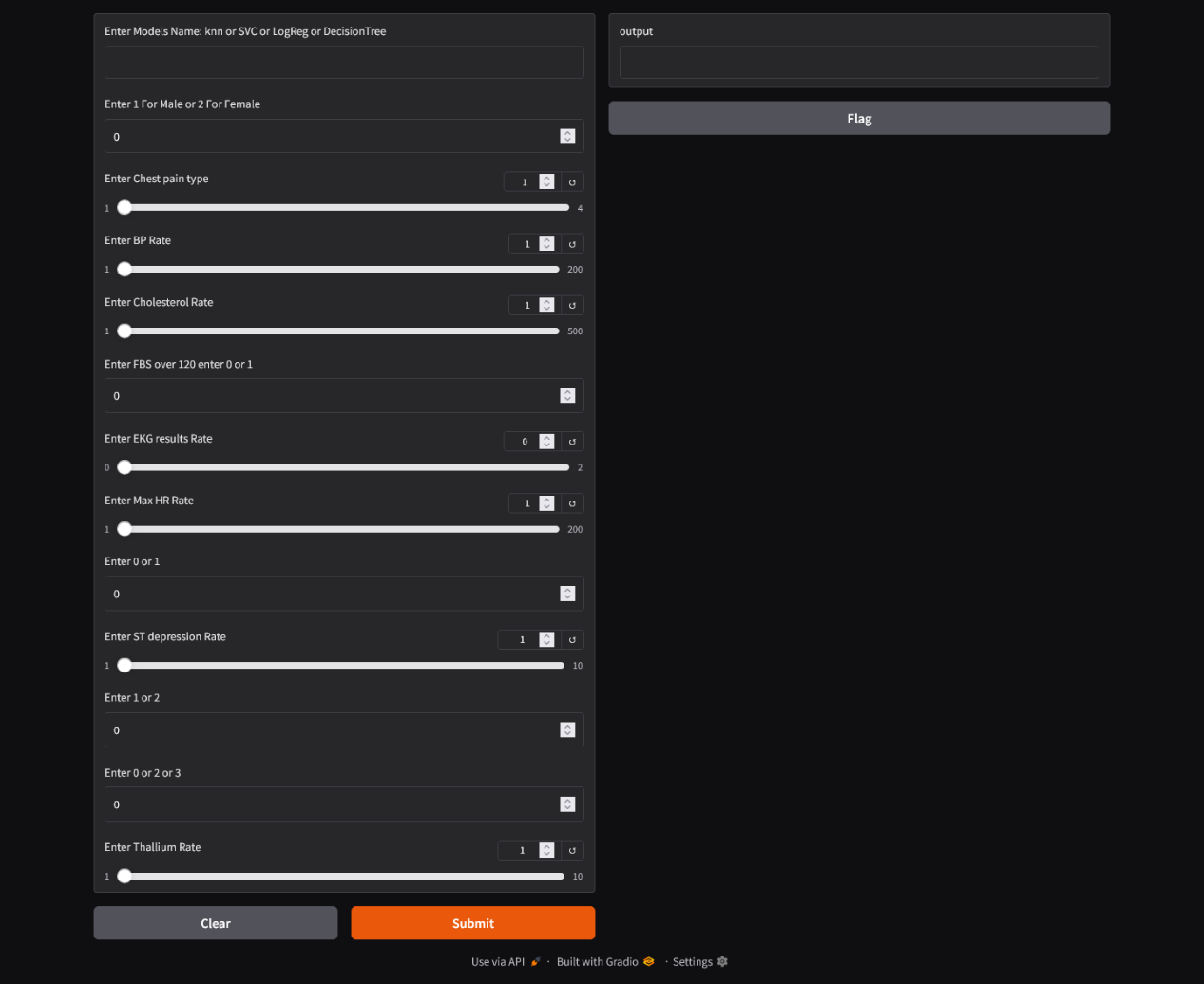
كما تم اختبار النماذج الأربعة المستخدمة في هذا المشروع من خلال عدد من الأمثلة. وأظهرت هذه النماذج قدرتها على التنبؤ بحالة مرض القلب

تم تحقيق هذا الاختبار من خلال واجهة مستخدم تفاعلية تظهر نتيجة إدخال مثال محدد.

* 1. **اختبار التنفيذ:**

في هذا الجزء سنشرح بعض عمليات التنفيذ والاختبار للأشكال الأربعة في حالات الكشف عن مرض القلب. وسيتم تطبيق هذه الاختبارات من خلال واجهة المستخدم التفاعلية التي تم بناؤها واختبارها بحيث يتم معالجة كافة الأخطاء التي حدثت نتيجة إدخال بيانات خاطئة من قبل المستخدم. فيما يلي بعض عمليات الاختبار.

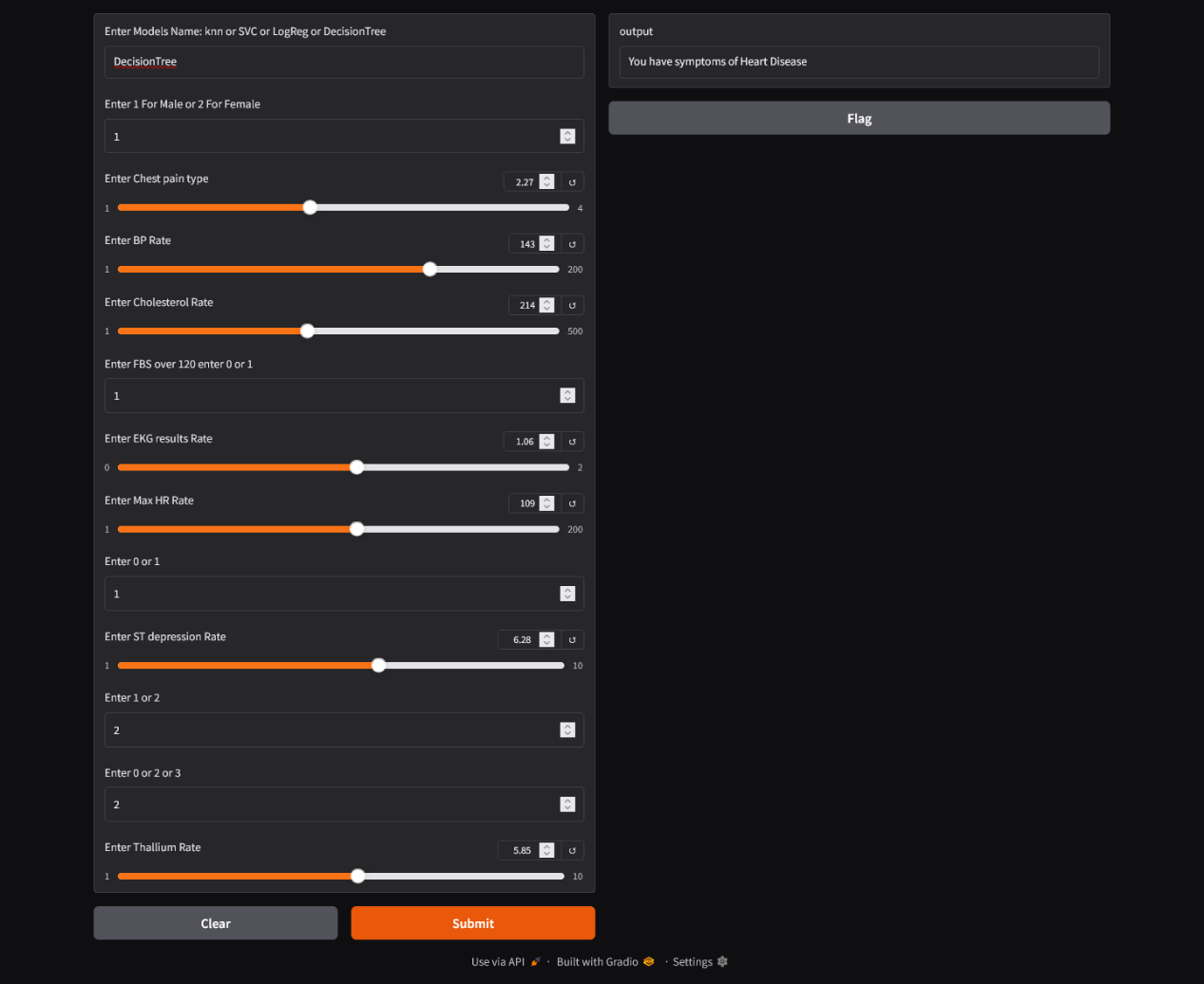
تحتوي الواجهة على جزأين الأول لعملية الإدخال والجزء الثاني لإظهار المخرجات. من خلال هذه الواجهة يمكننا تحديد اسم النموذج المتعلم ومن ثم إدخال البيانات الشخصية والصحية وإظهار النتيجة في الجزء الثاني.



**الشكل رقم ‏4‑1 صفحة الويب**

لنفترض، على سبيل المثال، أننا اخترنا شجرة القرار كنموذج للتنبؤ، ثم تم إدخال بيانات المريض، وعند الضغط على زر الإرسال، يتم عرض النتيجة في قسم الإخراج.

ويوضح الشكل التالي عملية الاختبار للنموذج المحدد



**الشكل رقم ‏4‑2 استخدام نموذج شجرة القرار لاكتشاف مرض القلب**

وفي النهاية يمكن اختبار باقي النماذج في حالتي الكشف أو عدم الكشف بنفس الطريقة، على أن يتم تحديد اسم ملف النموذج وإدخال باقي البيانات المطلوبة لعملية التنبؤ.

# الفصل الخامس: الخلاصة والافاق المستقبلية:

1. 1. **الخلاصة والتوصيات:**

في نهاية البحث نصل الى مجموعة من التوصيات:

* بدأ المشروع بمسح المجال المختار لإيجاد المشكلة العلمية ، وهو تطبيق مجالات استخراج البيانات والتعلم الآلي للكشف عن مرض القلب عبر نماذج تعلم الآلة.
* تم مراجعة الأدبيات الخاصة بمجالات الكشف عن مرض القلب باستخدام التعلم الآلي ، والتي أصبحت الأساس لوضع وإنشاء مواصفات المتطلبات للمشروع الذي تم تطويره.
* تم في هذا المشروع استخدام جميع التقنيات المذكورة أعلاه وكان قادرًا على أخذ وصف صحة المستخدم كمدخلات وإخراج نتيجة التصنيف بعد تمريرها إلى النموذج المُدرب
* تم اجراء مقارنة بين النماذج التي تم اعتمادها في هذا المشروع وقد حققت دقة اختبار تصل إلى 92% تقريبًا.
* تم عرض النتيجة للمستخدم عبر صفحة ويب تركز على تفاعل المستخدم والتي تهدف إلى مساعدة المريض على اكتشاف مرض القلب وبطريقة أكثر تفاعلية.
* بشكل عام، وفقًا لاختبارات الأداء واختبار المستخدم، نجح هذا المشروع في تحقيق الهدف منه - ومع ذلك، كان هناك حاجة إلى مزيد من التطوير حتى يصبح قابلاً للتطبيق على الصعيد العملي.
  1. **الافاق المستقبلية:**
* هناك العديد من الإضافات التي يمكن إضافتها إلى هذا المشروع والتي قد تسمح له بأن يصبح أكثر احترافية وعملية. انطلاقًا من الحاجة إلى البيانات الإضافية اللازمة لتدريب النماذج بشكل أفضل.
* من الممكن إضافة خاصية وضع صفحة الويب على الاستضافة الإلكترونية وربطها بالنظام الصحي الإلكتروني ليتمكن الشخص من استخدام التطبيق بشكل تشاركي وسريع.
* وهناك ميزة اخرى ايضا وهي توسيع عمل التطبيق بحيث نقوم ببناء موقع كامل لاحد المستشفيات وامكانية تقديم الطلبات وادارتها من خلال هذا الموقع ومن ثم استخدام النموذج الافضل الذي اخترناه في عملية الكشف عن مرض القلب بشكل آلي ومباشر دون الرجوع للأطباء المسؤولين مما يساهم في توفير الوقت والجهد في عملية دراسة الحالة الصحية المقدمة من المريض.

# المراجع:

1. أحسن, طيار, "التنقيب في البيانات واتخاذ القرارات", الملتقى الوطني السادس, جامعة سكيكدة, الجزائر, 27-28 كانون الثاني 2009.
2. خلوف, رزوق, شميسة, "استخدام آلية التنقيب في المعطيات للمساعدة في اكتشاف عمليات الاحتيال في بيئة مصرفية", مجلة جامعة دمشق للعلوم الهندسية المجلد الخامس والعشرون, العدد الثاني, 2009, جامعة دمشق.
3. المرسوم التشريعي /322/ لعام 2005.
4. "تحليل خوارزميات التصنيف (Bayes, Neural, tree) من خلال استخدام تطبيق ويكا", 2012, المركز المشترك للأبحاث الأكاديمية والصناعية.
5. *P. Chaudhari, H. Agarwal, and V. Bhateja, “Data augmentation for cancer classification in oncogenomics: an improved KNN based approach,” Evol. Intell., pp. 1–10, 2019.*
6. *S. F. Khorshid and A. M. Abdulazeez, “BREAST CANCER DIAGNOSIS BASED ON K-NEAREST NEIGHBORS: A REVIEW,” PalArch’s J. Archaeol. Egypt/Egyptology, vol. 18, no. 4, pp. 1927–1951, 2021.*
7. *F. Q. Kareem and A. M. Abdulazeez, “Ultrasound Medical Images Classification Based on Deep Learning Algorithms: A Review.”*
8. *D. Q. Zeebaree, A. M. Abdulazeez, D. A. Zebari, H. Haron, and H. N. A. Hamed, “Multi-Level Fusion in Ultrasound for Cancer Detection Based on Uniform LBP Features.”*
9. *J. R. F. Junior, M. Koenigkam-Santos, F. E. G. Cipriano, A. T. Fabro, and P. M. de Azevedo-Marques, “Radiomics-based features for pattern recognition of lung cancer histopathology and metastases,” Comput. Methods Programs Biomed. vol. 159, pp. 23–30, 2018.*
10. *Ibrahim and A. Abdulazeez, “The Role of Machine Learning Algorithms for Diagnosing Diseases,” J. Appl. Sci. Technol. Trends, vol. 2, no. 01, pp. 10–19, 2021.*
11. *P. Das, B. Das, and H. S. Dutta, “Prediction of Lungs Cancer Using Machine Learning,” EasyChair, 2020.*
12. *G. A. P. Singh and P. K. Gupta, “Performance analysis of various machine learning-based approaches for detection and classification of lung cancer in humans,” Neural Comput. Appl., vol. 31, no. 10, pp. 6863–6877, 2019.*
13. *B. Charbuty and A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” J. Appl. Sci. Technol. Trends, vol. 2, no. 01, pp. 20–28, 2021.*
14. *"Intelligent Heart Disease Prediction System Using Data Mining Techniques" ,International J. of Healthcare & Biomedical Research, Volume: 1, Issue: 3, April 2013, Pages 94-101 www.ijhbr.com*
15. *Ameer H. Ali, Mohannad A. M. Al-Ja'afari, Saif Abdulwahed," Rheumatoid Arthritis Diagnosis Based on Intelligent System" , Journal of University of Babylon for Pure and Applied Sciences ,Vol.(26), No.(7): 2018*
16. *Jyoti Soni , Ujma Ansari, Dipesh Sharma, Sunita Soni," Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction", International Journal of Computer Applications (0975 –8887), Volume 17–No.8, March 2011*
17. *Dr. G. Karraz," Automatic Classification of Heartbeats using Neural Network Classifier based on a Bayesian Framework" ,International Conference of the IEEE Engineering in Medicine and Biology Society,2006*
18. *Ishtake S.H , Sanap S.A., "Intelligent Heart Disease Prediction System Using Data Mining Techniques", International J. of Healthcare & Biomedical Research, Volume: 1, Issue: 3, Pages 94-101, April 2013*
19. *Nidhi Bhatla, Kiran Jyoti," An Analysis of Heart Disease Prediction using Different Data Mining Techniques" International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 1 Issue 8, October - 2012*
20. *Andrew Kusiak ,Rough Set Theory: "A Data Mining Tool for Semiconductor Manufacturing" ,IEEE Transactions on Electronics Packaging Manufacturing, VOL. 24, NO. 1,1-12 ,(2001)*
21. *Geert Verstraeten, Dirk Van den Poel, Michael Egmont-Petersen, Patrick Van Kenhove and Jan Vanthienen, "Bayesian network classifiers for identifying the slope of the customer lifecycle of long-life customers" ,Elsevier 0377-2217,Computing, Artificial Intelligence and Information Technology ,European Journal of Operational Research,508–523 , (2004)*
22. [*https://www.alfbaa01.com/2019/10/k-nearest-neighbors.html*](https://www.alfbaa01.com/2019/10/k-nearest-neighbors.html)
23. *Kalyan Moy Gupta, David W. Aha and Philip Moore, "Rough Set Feature Selection Algorithms for Textual Case-Based Classification" ,Proceedings of Eight European Conference on Case-Based Reasoning, ECCBR-06, Ölündeniz, Turkey: Springer,1-15, (2006(*
24. [*https://www.w3schools.com/sql/*](https://www.w3schools.com/sql/)
25. *Asuncion, A.; D.J. Newman, UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 2007*

1. Ameer H. Ali, Mohannad A. M. Al-Ja'afari, Saif Abdulwahed," Rheumatoid Arthritis Diagnosis Based on Intelligent System" , Journal of University of Babylon for Pure and Applied Sciences ,Vol.(26), No.(7): 2018 [↑](#footnote-ref-1)
2. [Jyoti Soni , Ujma Ansari, Dipesh Sharma, Sunita Soni," Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction", International Journal of Computer Applications (0975 –8887), Volume 17–No.8, March 2011 [↑](#footnote-ref-2)
3. Dr. G. Karraz," Automatic Classification of Heartbeats using Neural Network Classifier based on a Bayesian Framework" ,International Conference of the IEEE Engineering in Medicine and Biology Society,2006 [↑](#footnote-ref-3)
4. Ishtake S.H , Sanap S.A., "Intelligent Heart Disease Prediction System Using Data Mining Techniques", International J. of Healthcare & Biomedical Research, Volume: 1, Issue: 3, Pages 94-101, April 2013 [↑](#footnote-ref-4)
5. Nidhi Bhatla, Kiran Jyoti," An Analysis of Heart Disease Prediction using Different Data Mining Techniques" International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 1 Issue 8, October - 2012 [↑](#footnote-ref-5)