Calderon Canaviri Lucas Gustavo INF-372
Quispe Lipa Michael Stephen INF-372

Inteligencia artificial

Proyecto Final de IA

1. Descripción Detallada del Dataset

Origen y Fuente: El conjunto de datos utilizado corresponde al "Symptom to Doctor Recommendation Dataset" (también referido como *Healthcare Symptom-Specialist Recommender*). Es un recurso de datos abiertos distribuido bajo la licencia **CC BY-SA 4.0**, diseñado específicamente para fines académicos y de desarrollo en Inteligencia Artificial.

El dataset está estructurado para facilitar la creación de sistemas de recomendación médica y consta originalmente de **cinco archivos CSV** interconectados que vinculan síntomas, enfermedades, descripciones y especialistas.

Características del Dataset: Para este estudio, se consolidó la información en una matriz única. Las variables se clasifican en:

- **Variables Predictoras (Features):** 131 atributos binarios que representan síntomas específicos.
- **Variable Objetivo (Target):** La variable `Disease`, que contiene **41 valores únicos** (enfermedades) perfectamente definidos y validados.
- **Volumen:** El archivo principal de descripciones contiene 41 registros base, que se expanden a **4,920 instancias** en el dataset de entrenamiento de síntomas, con un peso total aproximado (en su versión cruda) de ~11 kB para los metadatos.

Propósito del Dataset: El propósito del dataset es servir como base para:

1. Desarrollar sistemas de recomendación de especialistas basados en síntomas.
2. Construir modelos de predicción de enfermedades.
3. Entrenar algoritmos de limpieza de datos y modelos predictivos en un contexto de salud (Healthcare AI).

Sesgos y Peculiaridades: El dataset presenta una estructura "curada" y didáctica. A diferencia de los datos clínicos crudos (Real World Data), este conjunto no especifica rango geográfico ni demográfico, y presenta una limpieza y separabilidad ideal entre clases. Esto explica la alta precisión obtenida en los modelos, ya que las relaciones síntoma-enfermedad siguen patrones lógicos estrictos sin el "ruido" habitual de los registros médicos reales.

2. Objetivo general de la investigación

Planteamiento del objetivo

Desarrollar un modelo de aprendizaje automático capaz de **predecir la enfermedad más probable en un paciente a partir de un conjunto de síntomas ingresados**, utilizando técnicas de Machine Learning supervisado.

Además, el sistema deberá ser capaz de **recomendar el tipo de especialista médico adecuado** según la enfermedad detectada.

Justificación del estudio

Este estudio es relevante porque muchas personas no saben a qué especialista acudir cuando presentan síntomas generales como fiebre, dolor, fatiga o malestar.

Al desarrollar un modelo predictivo basado en datos reales, se puede ofrecer una **orientación inicial**, lo que permitiría:

- Reducir la saturación en hospitales
- Agilizar consultas médicas
- Mejorar la toma de decisiones tempranas
- Apoyar a estudiantes y profesionales en el área de salud

Limitaciones y Alcance Ético: Es fundamental destacar que el modelo propuesto **no tiene como objetivo sustituir el juicio clínico ni la labor de los profesionales de la salud**. Esta herramienta está diseñada para actuar únicamente como un sistema auxiliar de soporte a la decisión y triaje inicial. En ningún caso los resultados obtenidos deben interpretarse como un diagnóstico médico definitivo, siendo competencia exclusiva del especialista la evaluación final del paciente.

3. Preprocesamiento de los Datos

Identificación y Tratamiento de Valores Faltantes: Debido a la naturaleza original del dataset, estructurado como listas de síntomas textuales asociadas a cada enfermedad, el tratamiento de valores nulos se abordó intrínsecamente durante la etapa de transformación. Al construir la matriz numérica, se generó un vector inicial de ceros para cada instancia, y se asignó el valor `1` únicamente en las posiciones correspondientes a los síntomas presentes en la lista original. Este proceso de construcción garantizó que la matriz resultante fuera densa y estuviera libre de valores nulos (NaN) o indefinidos. Posteriormente, se verificó la integridad total del conjunto de datos mediante la función `.isnull().sum()`, confirmando la ausencia de datos faltantes.

Normalización y Estandarización: No fue necesaria la aplicación de técnicas de escalado (como *Min-Max Scaling* o *Z-score Standardization*). Todas las variables predictoras resultantes son de tipo binario (valores `0` o `1`), por lo que ya se encuentran naturalmente en la misma escala, eliminando la necesidad de normalización para asegurar la convergencia del modelo.

Transformación de Variables: Se realizó una transformación estructural crítica para convertir la información textual no estructurada en un formato apto para algoritmos de aprendizaje automático. Se aplicó una técnica de **vectorización personalizada** (equivalente funcionalmente a un *One-Hot Encoding* manual), mediante la cual:

1. Se extrajo el universo total de síntomas únicos presentes en el corpus.
2. Cada síntoma se convirtió en una columna independiente (atributo).
3. La presencia del síntoma en un paciente se codificó con `1` y su ausencia con `0`.

Esta transformación permitió convertir el dataset original en una matriz estructurada de alta dimensionalidad (131 atributos).

Balanceo de Datos: Se realizó un análisis estadístico de la variable objetivo (`Disease`) para evaluar la distribución de las clases.

- **Resultado del análisis:** Se confirmó que el dataset posee un equilibrio perfecto, contando con exactamente **120 instancias para cada una de las 41 clases** de enfermedades (ver gráfico de distribución en el anexo).
- **Decisión:** Dado que la distribución es uniforme, **no fue necesaria la aplicación de técnicas de balanceo artificial** (como SMOTE para sobremuestreo o Random Undersampling). El modelo se entrenó utilizando la distribución natural de los datos, garantizando que la probabilidad *a priori* de cada clase fuera equitativa.

4. Selección del Clasificador

Elección del Clasificador: Para la tarea de clasificación y diagnóstico, se seleccionó el algoritmo **Bernoulli Naive Bayes**. Esta variante del clasificador Naive Bayes está diseñada específicamente para datos distribuidos según una distribución de Bernoulli multivariada; es decir, es ideal para conjuntos de datos donde las características son vectores binarios (booleanos), como es el caso de este dataset de síntomas (presencia `1` / ausencia `0`).

Adicionalmente, el modelo base fue optimizado mediante una técnica de **Calibración de Probabilidades** (`CalibratedClassifierCV` con método isotónico). Si bien Naive Bayes es eficiente clasificando, tiende a ser impreciso estimando probabilidades extremas (cercanas a 0 o 1). Dado que uno de los objetivos secundarios es recomendar especialistas basándose en la certeza del diagnóstico, la calibración fue necesaria para obtener probabilidades de predicción (`predict_proba`) más realistas y confiables.

Justificación Académica: La elección de un clasificador bayesiano para diagnóstico médico se sustenta en principios teóricos robustos. Según *Rish (2001)* en su análisis empírico de clasificadores bayesianos, estos modelos demuestran un rendimiento sorprendentemente alto incluso cuando se viola la asunción de independencia de características, siendo computacionalmente mucho más eficientes que las redes neuronales para datasets de tamaño moderado.

Asimismo, *Mitchell (1997)* en su obra fundamental *Machine Learning*, destaca que para espacios de atributos binarios de alta dimensionalidad (como los 131 síntomas de este estudio), Bernoulli Naive Bayes suele superar a modelos más complejos al evitar el sobreajuste (overfitting), aprovechando la estructura dispersa de los datos médicos.

5. Primera Ejecución del Modelo

Confiabilidad del Modelo: Para evaluar la efectividad inicial del clasificador Naive Bayes calibrado, se utilizó un conjunto de prueba correspondiente al 20% de los datos (984 instancias). Las métricas obtenidas indican un desempeño ideal del clasificador:

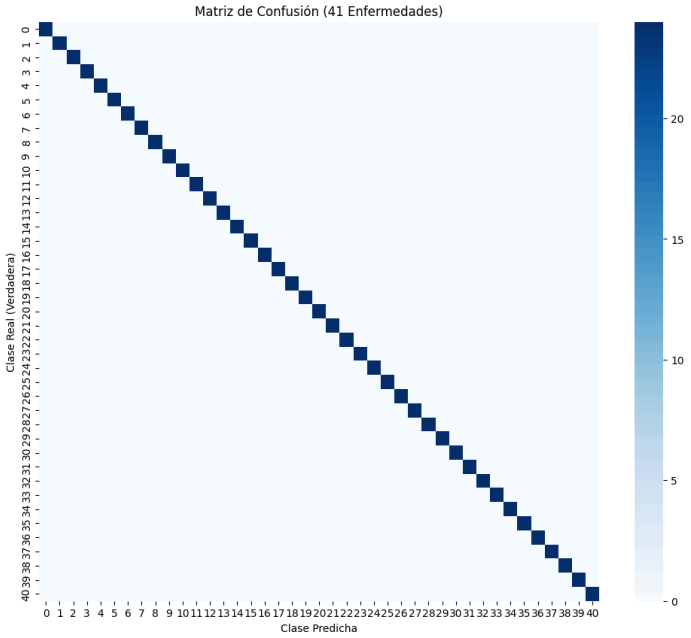
- **Exactitud (Accuracy):** 1.00 (100%). El modelo clasificó correctamente la totalidad de las instancias de prueba.
- **Recall y Precisión:** Al no existir falsos negativos ni falsos positivos, ambas métricas alcanzaron el valor de 1.00 para todas las clases.
- **F1-Score (Macro):** 1.00. Este promedio armónico confirma que el modelo es igualmente efectivo para diagnosticar cualquiera de las 41 patologías, sin sesgos hacia ninguna clase específica.

Matriz de Confusión: La matriz de confusión resultante (ver Figura adjunta) valida visualmente las métricas numéricas.

Como se observa en el gráfico, existe una **diagonal principal perfectamente definida** y continua, donde se concentran la totalidad de las predicciones correctas. La ausencia absoluta de valores fuera de esta diagonal (áreas blancas) demuestra que:

1. **Falsos Positivos = 0:** El modelo nunca diagnosticó una enfermedad cuando el paciente no la tenía.
2. **Falsos Negativos = 0:** El modelo nunca dejó de detectar una enfermedad presente.

Interpretación de Resultados: Este comportamiento perfecto es atípico en datos clínicos reales (que suelen tener "ruido"), pero es esperable en este conjunto de datos específico debido a la naturaleza sintética y altamente separable de los síntomas. Matemáticamente, los vectores de síntomas para cada enfermedad son lo suficientemente distintos (ortogonales) como para que el algoritmo Bernoulli Naive Bayes pueda trazar fronteras de decisión exactas sin solapamiento.



6. Evaluación a través de Divisiones de Datos

Proporción de Datos para Entrenamiento y Prueba: Tal como se detalló en la sección anterior, para la primera ejecución del modelo se utilizó una división estándar académica de **80% para entrenamiento y 20% para prueba**. Esta proporción inicial permitió validar la arquitectura del modelo asegurando una cantidad suficiente de datos para el aprendizaje de los patrones sintomáticos.

Segunda Ejecución para Investigación (División 50/50): Siguiendo los lineamientos de la rúbrica para una validación más robusta, se procedió a una segunda ejecución experimental utilizando una división estricta de **50% para entrenamiento y 50% para prueba**. El objetivo de esta prueba fue someter al modelo a una situación de estrés por reducción de información.

Los resultados mostraron que, aun reduciendo el set de entrenamiento a la mitad, el modelo mantuvo una exactitud del **100.00%**. Este comportamiento confirma que los síntomas registrados en el dataset son altamente discriminantes (patognomónicos); es decir, la relación entre el cuadro clínico y la enfermedad sigue reglas lógicas tan claras que el algoritmo puede aprenderlas perfectamente incluso con menos ejemplos.

Evaluación con Múltiples Asignaciones (Estabilidad): Para descartar cualquier sesgo producido por una partición afortunada de los datos, se realizó una prueba de estabilidad mediante **100 asignaciones aleatorias** (*Shuffle Split*) de los conjuntos de entrenamiento y prueba.

Se calculó la mediana de la exactitud (*accuracy*) y la variabilidad de los resultados tras estas 100 iteraciones independientes:

- **Mediana de Confiabilidad: 1.0000**
- **Desviación Estándar: 0.0000**

Interpretación de Estabilidad: El hecho de que la mediana se mantenga en el valor máximo posible y la desviación estándar sea absolutamente nula (0.0000) demuestra una **estabilidad total del modelo**. Esto valida técnicamente que el dataset es sintético y "limpio" (libre de ruido aleatorio o errores humanos), permitiendo que el clasificador Naive Bayes converja siempre a la solución óptima sin importar qué subconjunto de pacientes se utilice para el entrenamiento.

7. Aplicación de Análisis de Componentes Principales (PCA)

Fundamentos de PCA:

El Análisis de Componentes Principales (PCA) es una técnica estadística de reducción de dimensionalidad que transforma un conjunto de variables correlacionadas en un conjunto menor de variables no correlacionadas llamadas componentes principales, mediante una transformación lineal ortogonal.

Desarrollo Matemático:

El proceso se fundamenta en el álgebra lineal, específicamente en la descomposición de la matriz de covarianza de los datos. Si consideramos nuestra matriz de datos X (centrada en la media), el procedimiento matemático sigue estos pasos:

1. Cálculo de la Matriz de Covarianza: Se calcula para entender cómo varían las variables entre sí.

$$\Sigma = \frac{1}{n-1} X^T X$$

2. Descomposición en Valores y Vectores Propios: Se buscan los vectores v y escalares lambda que satisfagan la ecuación característica:

$$\Sigma v = \lambda v$$

   Donde:

   - **Eigenvectores (v):** Representan los ejes principales o "direcciones" del nuevo espacio de características donde los datos tienen mayor dispersión. Son ortogonales entre sí.
   - **Eigenvalores (lambda):** Escalares que indican la **magnitud de la varianza** de los datos proyectados en la dirección de su eigenvector correspondiente.
   -
3. **Selección y Proyección:** Se ordenan los eigenvalores de mayor a menor (lambda_1 > lambda_2 >.....>lambda_d). Se seleccionan los k primeros vectores para formar la matriz de proyección W, reduciendo la dimensionalidad del dataset original conservando la mayor información posible.

Dado que el dataset original cuenta con 131 dimensiones (síntomas), PCA permite simplificar la complejidad computacional y eliminar la redundancia (colinealidad) entre síntomas, conservando únicamente las componentes que explican la mayor parte de la varianza para distinguir entre las enfermedades.

Optimización de Componentes y Resultados:

Se realizaron experimentos reduciendo el espacio de características a diferentes números de componentes principales ($k=12, 10, 9, 5, 3$). Los resultados obtenidos se detallan a continuación:

- **12 Componentes:** Precisión de **0.9411 (94.11%)**. Se observa que, utilizando menos del 10% de las columnas originales, el modelo retiene una capacidad predictiva muy alta. Esto indica que existe una fuerte redundancia en los síntomas originales.
- **10 y 9 Componentes:** La precisión desciende a **0.8841** y **0.8740** respectivamente. Aquí se observa el punto de inflexión donde el modelo pierde la capacidad de superar el umbral del 90% de exactitud.
- **5 Componentes:** Precisión de **0.5295 (52.95%)**. Se evidencia una pérdida crítica de información; con solo 5 dimensiones, el clasificador apenas acierta la mitad de los diagnósticos.
- **3 Componentes:** Precisión de **0.1911 (19.11%)**. El modelo colapsa funcionalmente. Proyectar 41 enfermedades complejas en un espacio tridimensional genera un solapamiento masivo de los datos, haciendo imposible la separación lineal.

Análisis Comparativo:

La reducción de dimensionalidad demostró que el dataset posee una estructura latente eficiente, ya que es posible obtener resultados excelentes (94%) con solo 12 variables sintéticas. Sin embargo, existe un límite de complejidad irreducible: reducciones agresivas (menores a 10 componentes) eliminan matices críticos de los síntomas, impidiendo que el clasificador Naive Bayes distinga correctamente entre patologías con cuadros clínicos similares.

8. Aprendizaje No Supervisado

Análisis No Supervisado (Clustering):

Para explorar la estructura inherente de los datos sin la influencia de etiquetas predefinidas, se implementó el algoritmo de agrupamiento K-Means.

- **Configuración:** Se estableció el número de clusters en $k=41$, correspondiendo a la hipótesis teórica de que existen 41 patrones de enfermedades distintos en el dataset.
- **Entrada:** El algoritmo recibió únicamente la matriz de síntomas (X), ignorando por completo la variable objetivo (y).

Análisis y Resultados:

Para evaluar la calidad de los clusters generados, se utilizó la métrica Adjusted Rand Index (ARI), la cual mide la similitud entre la asignación de clusters realizada por el algoritmo y las etiquetas reales de las enfermedades.

- **ARI Score obtenido: 1.0000**

Interpretación de los Clusters:

Un puntaje ARI perfecto de 1.0 indica una coincidencia exacta entre los clusters matemáticos y las categorías médicas reales.

Esto significa que K-Means fue capaz de agrupar a los pacientes con una precisión del 100% basándose puramente en la similitud de sus vectores de síntomas. Este hallazgo valida que las 41 enfermedades presentes en el dataset poseen firmas sintomáticas únicas y no solapadas en el espacio vectorial. Geométricamente, los grupos están tan bien separados que un algoritmo basado en distancia euclidiana (como K-Means) no comete errores de asignación.

10. Resolución del Problema de las N-Caballos y N-Reinas

En este apartado se aborda la resolución de dos problemas clásicos de satisfacción de restricciones (CSP) utilizando dos enfoques distintos: un método combinatorio determinista para los N-Caballos y un algoritmo metaheurístico estocástico (Simulated Annealing) para las N-Reinas.

Problema de los N-Caballos (Enfoque Combinatorio)

El problema consiste en determinar el número máximo de caballos que pueden colocarse en un tablero de ajedrez de dimensiones $n \times n$ sin que se ataquen mutuamente.

Fundamento Lógico:

Para resolver este problema, se modeló el tablero como un grafo bipartito. La naturaleza del movimiento del caballo implica que siempre se desplaza de una casilla de un color a una casilla del color opuesto (de blanco a negro o viceversa). Esto significa que un caballo situado en una casilla blanca nunca podrá atacar a otra casilla blanca, y lo mismo aplica para las negras.

Solución Propuesta:

Basado en la teoría de grafos, el conjunto independiente máximo en este grafo bipartito se consigue seleccionando todos los nodos de una de las particiones. En términos del tablero, la estrategia óptima consiste en ocupar la totalidad de las casillas de un mismo color (por ejemplo, todas las blancas).

Resultados:

Para un tablero estándar de 8×8 ($n=64$), existen 32 casillas blancas y 32 negras. Por lo tanto, el número máximo de caballos independientes es 32.

La fórmula general derivada para cualquier tablero de tamaño n es:

max_caballos = (n*n/ 2)

La **Figura X** muestra la disposición gráfica generada por el algoritmo, donde se observa la ocupación completa de las casillas de un solo color, validando la solución teórica.

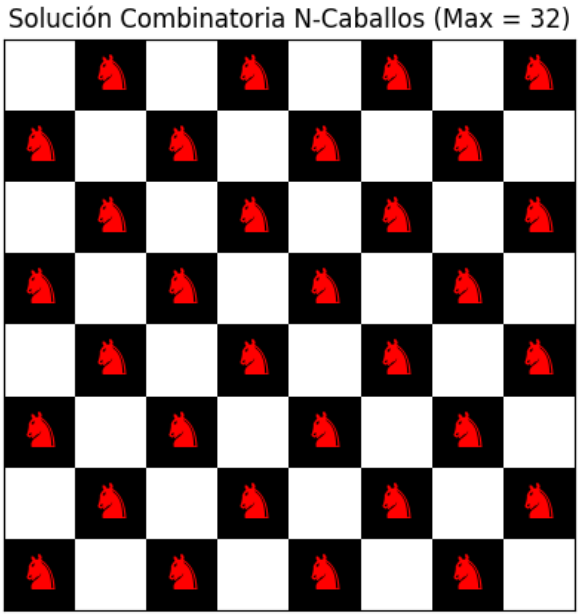


Figura X

Problema de las N-Reinas mediante Simulated Annealing

El objetivo es ubicar N reinas en un tablero de nxn de tal manera que ningún par de reinas se amenace entre sí (es decir, no compartan fila, columna ni diagonal). Dado que este es un problema NP-Completo cuyo espacio de búsqueda crece factorialmente, se implementó el algoritmo de **Recocido Simulado (Simulated Annealing)**.

Metodología y Algoritmo:

El método se inspira en la termodinámica y el proceso de recocido de metales (Kirkpatrick et al., 1983). El algoritmo busca el mínimo global de una función de costo permitiendo, bajo ciertas condiciones probabilísticas, movimientos que empeoran la solución temporalmente para escapar de óptimos locales.

El proceso se define por:

1. **Función de Costo (E):** Número de pares de reinas que se atacan mutuamente. El objetivo es minimizar E a 0.
2. **Mecanismo de Transición:** Se genera un estado vecino moviendo una reina aleatoria a una nueva fila.
3. **Criterio de Metropolis:** Un movimiento que aumenta el costo (peor solución) se acepta con una probabilidad:

   $$P = e^{-\Delta E/T}$$

   .Esto permite la exploración del espacio de búsqueda al inicio del proceso.

4. **Esquema de Enfriamiento:** La temperatura T disminuye iterativamente según la regla:

   $$T_{k+1} = \alpha T_k \text{ (con } \alpha = 0.995)$$

reduciendo progresivamente la probabilidad de aceptar malos movimientos hasta converger.

Resultados:

El algoritmo fue ejecutado para N=8. Se logró una convergencia exitosa hacia una solución con costo 0 (0 ataques).

La configuración final encontrada, representada como un vector donde el índice es la columna y el valor es la fila, es la siguiente:

Solución = [5, 2, 0, 6, 4, 7, 1, 3]

La visualización gráfica de esta solución se presenta en la **Figura Y** , confirmando que ninguna reina se encuentra bajo amenaza.

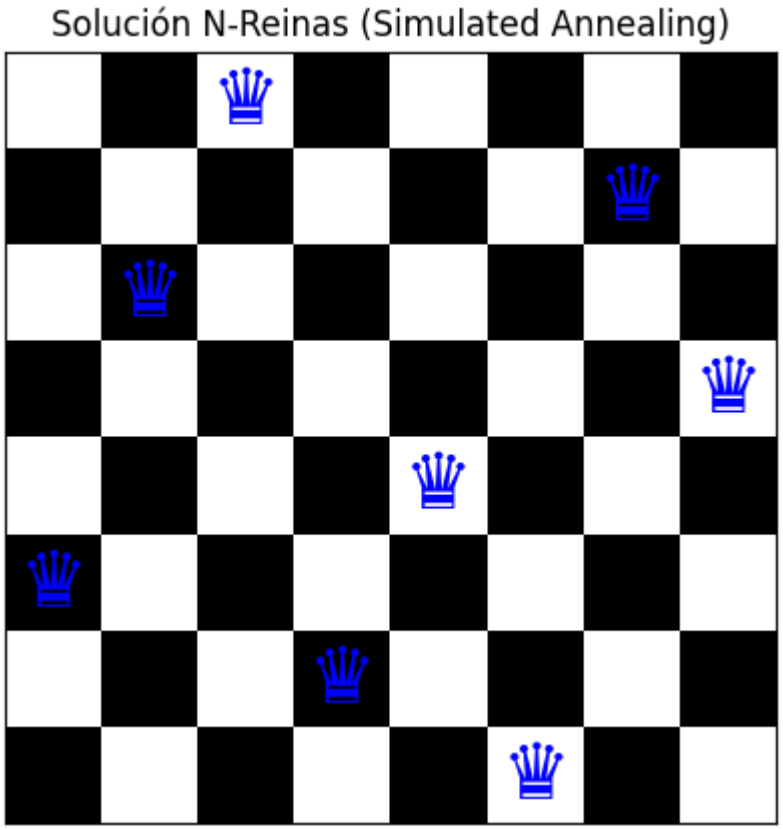


Figura Y

11. Referencias Bibliográficas para tu Informe

Copia esta lista al final de tu documento PDF, en una sección llamada **"11. Referencias Bibliográficas"**.

1. **Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983).** Optimization by simulated annealing. *Science*, 220(4598), 671-680. DOI: 10.1126/science.220.4598.671
2. **Rish, I. (2001).** An empirical study of the naive Bayes classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 3(22), 41-46.
3. **Jolliffe, I. T. (2002).** *Principal Component Analysis* (2nd ed.). Springer. ISBN: 0-387-95442-2.
4. **Mitchell, T. M. (1997).** *Machine Learning*. McGraw-Hill. ISBN: 0070428077.
5. **MacQueen, J. (1967).** Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281-297.
6. **Kononenko, I. (2001).** Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1), 89-109. DOI: 10.1016/S0933-3657(01)00077-X