Tecnológico de Monterrey

Evidencia 1

El Aprendizaje Supervisado como Herramienta de Predicción de Enfermedades en el Área de la Salud

Proyecto de Aprendizaje Supervisado

Modelación del Aprendizaje con Inteligencia Artificial

TC2034.101

Gpo. 101

Raúl Correa Ocañas	A01722401
Sebastián Miramontes Soto	A01285296
Ericka Sofia Rodriguez Sanchez	A01571463

Docente:

Dr. Santiago Enrique Conant Pablos

Monterrey, Nuevo León, 10 de marzo de 2024

1. Introducción

La humanidad siempre se encuentra en constante evolución, y con los avances tecnológicos y científicos se busca alcanzar una vida más cómoda, segura y extensa para el ser humano. La medicina es una rama que ha logrado avances significativos con el paso del tiempo, encontrando cada vez mejores maneras de diagnosticar y curar enfermedades. Por otro lado, la tecnología y los conocimientos han crecido de manera que se están combinando ramas como la estadística y la informática, para analizar datos de forma que las computadoras pueden aprender de ellos y realizar modelos efectivos capaces de clasificar y predecir etiquetas. Es entonces cuando aparece este reto de combinar ambas disciplinas para lograr en conjunto un mayor avance.

En el siguiente proyecto se llevará a cabo una investigación, mediante la cuál a través de una base de datos médica, se buscará encontrar los mejores modelos para clasificar enfermedades de acuerdo a síntomas presentados. Todo esto utilizando técnicas de aprendizaje supervisado, para entrenar modelos, analizar su desempeño, y en base a ello tomar el más adecuado para el diagnóstico en la medicina.

2. Marco Teórico

2.1 Aprendizaje Supervisado en Salud

El aprendizaje supervisado en salud ha sido funcional para muchas situaciones cotidianas, pero en cierto modo está muy relacionado con el sector salud. Principalmente se ha utilizado para la clasificación, diagnóstico, predicción o monitoreo de enfermedades o posibles enfermedades, esto mediante imágenes o conjuntos de datos específicos. Entre algunos de los Modelos de Aprendizaje Supervisado que pueden tener un uso médico están el Bosque Aleatorio, las Máquinas de Vector de Soporte (SVM), las redes neuronales, el Deep Learning, la Máquina de Aumento de Gradiente (GBM), entre otros (Aracena, 2022). Uno de

los usos más peculiares que se le da a este sistema en la Salud es el Diagnóstico de Enfermedades, algunas de estas son el Alzheimer, Cáncer, aunque también llegó a ser de utilidad durante cierto tiempo en la pandemia del COVID 19 en donde se tomaban en cuenta síntomas y características fisiológicas de los pacientes a tratar. Obviamente todavía es un riesgo confiar completamente solo en una predicción basada en Inteligencia Artificial, y tiene ciertos riesgos, pero es innegable que han existido avances gigantescos, que lo ponen como una gran posibilidad para el futuro (Hernandez, 2022).

2.2 Descripción del Dataset

Previo a comenzar con el proceso de presentar una solución de aprendizaje automatizado supervisado en un contexto médico, es crucial plantear que clase de predictores son relevantes para el diagnóstico de enfermedades. Bien que saber únicamente los síntomas de un paciente no siempre es suficiente para realizar un diagnóstico confiable y correcto, proponer un modelo que no requiera considerar todos los síntomas y que logre clasificar adecuadamente una enfermedad puede reducir el número de preguntas hechas para realizar un diagnóstico. La base de datos *Disease Prediction Data,* tomada de Kaggle, nos muestra un mapeo de cómo se distribuyen los síntomas en 42 enfermedades, con lo que se pueden aplicar conocimientos previamente adquiridos en el campo de la salud y ciencias.

Los conjuntos de datos fueron seleccionados con la intención de identificar y diagnosticar enfermedades a través de 132 síntomas en donde se incluían algunos muy generales como dolor de estómago, articulaciones o músculos, vómito, tos y fatiga, y algunas más específicas. Los datos se inicializan cargando los datasets en forma de 2 diferentes conjuntos diferentes, una de *training* y otra de *testing*, para realizar el análisis de manera más fluida. La mayor parte del código fue implementado sobre el dataset de *training* para verificar

su funcionamiento, ya que era el que tenía menos variables, y después con esto se implementa hacia el conjunto de *testing*.

2.3 Preprocesamiento

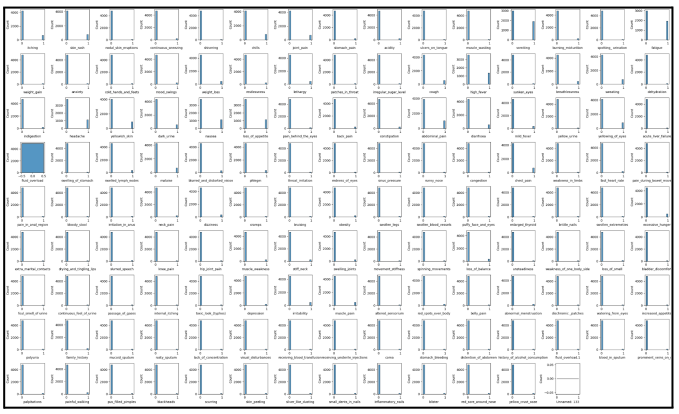
Para el preprocesamiento de los datos se inició con la eliminación de los datos que no aportan tanta información a la toma de decisiones en los modelos. Una variable identificada y descartada inmediatamente fue “Unnamed: 133”, debido a que la totalidad de esta columna resultó estar completa de NaNs. Al tener alrededor de 130 variables restantes, es evidente la posibilidad de que no todas aportan el mismo grado de información, por lo cual pueden ser redundantes. Se propuso como acción relevante el implementar un algoritmo de selección de variables, disminuyendo el número de columnas necesarias y a la vez buscar mantener métricas razonables para la resolución de la problemática. Sin tener más conocimiento en cuanto a las relaciones entre variables del set de datos, el realizar un análisis exploratorio de datos puede proporcionar un mejor panorama de las variables.

2.4 Análisis Exploratorio con Gráficos

Para el análisis exploratorio de los datos, se creó una tabla mostrando métricas estadísticas de la base de datos. Al notar que todas las variables predictoras tienen un rango de [0,1] y hacen referencia a algún síntoma, se infiere que su comportamiento es de estilo booleano. Esto indica que la base de datos ha sido codificada y transformada al estilo *One-Hot Encoding*.

Adicionalmente, se realizaron distintas gráficas para comprender mejor la naturaleza de los datos. Entre las generadas, se decidió mantener las más relevantes. La distribución de respuesta para cada síntoma es crucial para afirmar que la naturaleza de las variables predictoras efectivamente corresponde a lo observado en las métricas estadísticas. Con esto,

se pudo identificar fácilmente cuáles variables tienen valores significativamente desbalanceados. También es posible identificar cuáles fueron las que tenían más respuestas afirmativas, ya que por cuestión de lógica se sabía que en todos los síntomas siempre habría



más casos de ceros que de unos.

Imagen 1. Distribución de Respuestas para cada Síntoma.

El segundo gráfico que se analizó fue un mapa de calor en cuanto a la correlación entre síntomas. De forma visual, se pudo observar qué tan fuerte era la relación entre las variables. Las zonas rojas indican una relación directa fuerte, las blancas eran neutrales y las azules indican una relación inversamente fuerte. Se observó que en la zona del centro se encontraban cuadros rojos más grandes, mientras que otros se podían observar en zonas aleatorias. Además, se pudo notar que la mayoría del mapa tendía a ser neutral, descartando la diagonal que comparaba una variable consigo misma.

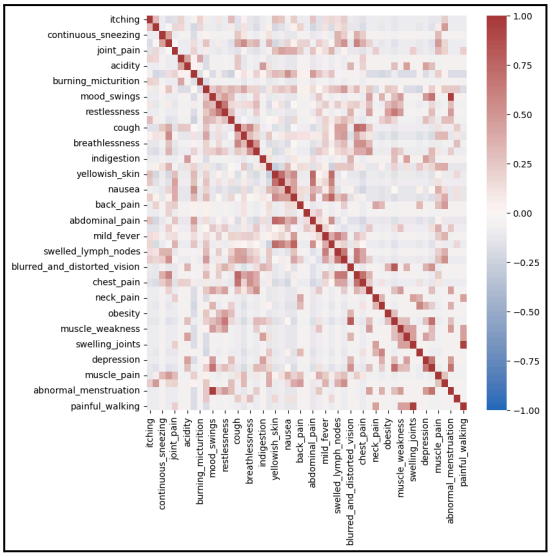


Imagen 2. Mapa de Calor de Correlación entre Síntomas

El tercer gráfico que se visualizó fue la distribución de enfermedades. En este gráfico se observó que su distribución es constante, por lo que no hay más datos de una enfermedad sobre otra. Esto es crucial para el entendimiento de los datos, debido a que no se tuviera una distribución uniforme, se tendría que aplicar un algoritmo de remuestreo de los registros. Esto se puede lograr ya sea con un método de sobremuestreo o submuestreo.

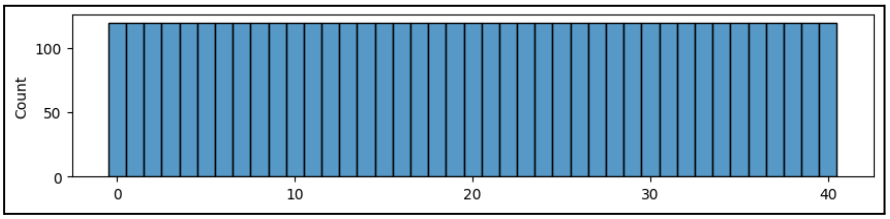


Imagen 3. Distribución de Enfermedades.

Con esta información en mente, se determina que algunas variables predictoras tienen una varianza cercana a cero. La varianza de una variable booleana se calcula con la ecuación $Var[X] = p(1 - p)$, con lo que se puede establecer un límite delimitador. Mientras menor sea la varianza de estas variables, la capacidad para utilizarse como una variable discriminatoria de clases predecidas disminuye. Por lo tanto, se implementa el algoritmo VarianceThreshold de Sci-kit Learn, en el que se remueven las variables en las cuales la proporción de la frecuencia de valor contra el total sea menor a 0.025. Esto redujo los predictores a 51 columnas.

3. Metodología

3.1 Esqueleto del Código

La primera etapa que se elaboró para manejar los datos en orden a encontrar los resultados fue la importación de librerías. En este paso, se ingresaron al código 12 librerías que se verían aplicadas en el código más adelante. Entre las más relevantes estuvo Pandas para la lectura de datos, matplotlib.pyplot para la creación de gráficos, y Sklearn que se utilizó en 6 importaciones diferentes. Sin embargo, estas últimas se vieron especialmente

necesitadas para poder implementar más adelante los modelos de Máquina de Vectores de Soporte y el Clasificador potenciado por Gradiente.

Posteriormente se realizó el preprocesamiento de los datos. En esta instancia, se ingresaron al código dos conjuntos de valores con los nombres de train y test, divididos en la cantidad de valores que tenían. Posteriormente, se utilizaron funciones como isnull para identificar las variables que contenían valores NaN. Además, se incluyeron gráficos previamente mostrados para encontrar las variables menos relevantes, las cuales fueron eliminadas de la tabla de valores mediante la función drop. Después de este proceso, se emplearon diferentes funciones para identificar las variables con menor varianza, ya que estas tampoco serían de mucha utilidad para el reto, la parte más compleja de esta sección fue el "feature selection" de las variables, lo cuál permitió reducirlas hasta llegar a un punto en el que fuera viable usar los modelos de aprendizaje supervisado.

En el análisis exploratorio de los datos a utilizar, la primera función empleada fue describir, la cual se utilizó para verificar que todas las variables restantes tuvieran la misma cantidad de datos. Además, se observó la media y la desviación estándar, lo que permitió visualizar cómo se distribuyen numéricamente los datos tanto para cada una de las variables como para el conjunto de datos en su totalidad. Posteriormente, se utilizaron los gráficos restantes para obtener una representación visual de la correlación entre los datos.

Primero se implementó el modelo de Máquina de Vectores de Soporte, a través de un código donde se escalan los números de prueba para posteriormente inicializar el modelo a utilizar, que en este caso fue el SVC. Utilizando Random Search, se encontraron los mejores hiper-parámetros para este caso. Luego, se llevó a cabo la implementación de la validación cruzada. Finalmente, se guarda la mejor configuarción para utilizarse más adelante al comparar modelos.

Posteriormente se realizó el Clasificador por Potenciado por Gradiente. En esta parte del código se sigue una estructura similar al paso anterior. Uno de los pocos cambios es que se define el modelo como Gradient Boosting Classifier. Luego, se ingresan diferentes diccionarios que ayudan al funcionamiento del código, así como la definición de la búsqueda aleatoria para encontrar los mejores hiper parámetros y el que contenga la mejor puntuación obtenida. Finalmente, se cierra esta parte del código ingresando los mejores hiper parámetros al modelo Gradient Boosting Classifier previamente mencionado.

Finalmente se diseñó una parte del código para facilitar la lectura de resultados y la comparación entre ambos modelos. El objetivo es determinar cuál de los dos modelos tiene una mayor precisión y, por lo tanto, cuál debe ser empleado en futuros casos similares. Según la teoría, se espera que el modelo de Gradient Boosting Classifier proporcione resultados más precisos debido a su capacidad para mejorar iterativamente la predicción mediante el ajuste del gradiente.

3.2 Modelo 1: Máquina de Vectores de Soporte

El primer modelo que fue utilizado para el proyecto fue la Máquina de Vectores de Soporte o “Support Vectors Machine”, el cuál es un algoritmo de aprendizaje automático supervisado, el cuál funciona a través de un hiperplano, lo que es una superficie de decisión lineal para a clasificación binaria de una clase y marcar la línea de decisión entre las respuestas posibles y las descartadas (Rodriguez, 2021). Una de las grandes ventajas de este Modelo es la gran efectividad que tiene en diversas aplicaciones, sobre todo en la clasificación de variables, además del uso de enfoques estadísticos para encontrar los mejores parámetros, pero una de sus más grandes ventajas es su uso con los espacios de grandes dimensiones, aparte de que son algoritmos de eficiente en memoria. Una de las desventajas que tiene el modelo es que no funciona de la mejor manera con conjuntos de datos muy

grandes ni cuando existe mucho ruido entre las variables, por lo mismo, el conjunto de datos seleccionado para la situación no tiene ninguna complejidad para trabajar con este modelo (Raj, 2022). Este sistema fue seleccionado, debido a que se considera que este modelo tiene un enfoque de clasificación prometedor para detectar personas con enfermedades comunes, en donde resaltan informes que lo relacionan con la diabetes y enfermedades cardiacas (Yu et al., 2010).

3.3 Modelo 2: Máquina de Aumento de Gradiente

El segundo modelo utilizado para este proyecto fue la Máquina de Aumento de Gradiente o "Gradient Boosting Machine" por su nombre en inglés, es un modelo que permite fortalecer los modelos de aprendizaje utilizando patrones para que el resultado final tenga un residuo cercano a 0, además de tener un funcionamiento en el que se le pueden aplicar diversas funciones de pérdida (Nelson, 2021). El funcionamiento del modelo se maneja a través de iteraciones donde se implementan fórmulas como $F_M(x_i) = F_{M-1}(x_i) + H_{M-1}(x_i)$, en donde M representa la iteración máxima alcanzada por el modelo y la $F_M(x_i)$ representa los modelos de aprendizaje débiles o "weak learners" (Zhang et al., 2019). Este método tiene distintas aplicaciones en la vida cotidiana, entre ellos los algoritmos de búsqueda o recomendación en anuncios o mercado electrónico, también tiene aplicaciones en las finanzas, pero la más relevante para este reto es su aplicación en la industria de la medicina y de la salúd, donde puede verse aplicado para recetas médicas, combinaciones de medicamentos, o el más similar al caso planteado es el diagnóstico de enfermedades, lo cuál combinado con la alta precisión, gran rendimiento y facilidad para la interpretación de los datos e implementación en diversos sistemas, fueron los principales factores para que se decidiera utilizar este modelo sobre otros como podrían ser las redes neuronales (Tuychiev, 2023).

4. Resultados

Antes de analizar los resultados de los métodos es importante conocer las métricas que se usan para evaluarlos. La exactitud se usa para obtener la proporción de clasificaciones correctas en general. La precisión sirve para calcular la proporción entre los positivos predecidos con los positivos reales. Recall o sensibilidad se usa para calcular todos los casos positivos sobre los verdaderos positivos y falsos negativos. Finalmente, el F1 combina la precisión y sensibilidad en un solo dato.

En la medicina es muy común darle más importancia o prioridad al recall, ya que con este se busca reducir el número de casos falsos negativos, ya que esto podría significar no diagnosticar una enfermedad grave de un paciente. También puede ser útil enfocarse en el F1 score, ya que mantiene un balance incluyendo precisión y sensibilidad.

4.1 Comparativa de Métodos

Modelo Implementado	Accuracy	Precision	Recall	F1
Gradient Boosting Classifier	0.976190	0.988095	0.976190	0.976190
SVC	0.928571	0.910714	0.928571	0.914286

Tabla 1. Resultados para modelos GBC y SVC

Analizando en detalle, se puede observar que el modelo de Gradient Boosting Classifier fue el mejor entre los dos utilizados. Teóricamente, se podría considerar que ambos modelos tuvieron un rendimiento notable, dado que ambos mostraron una exactitud y precisión bastante altas, ambas superando el 90%. Sin embargo, existe una diferencia significativa entre ellos, ya que la exactitud del método de gradiente es casi un 5% mayor que la de los vectores de soporte, y al mismo tiempo, la precisión es casi un 9% más alta en el método de gradiente que en el de vectores, acercándose bastante al 100% de precisión. Estos

resultados podrían haberse anticipado mediante la investigación previa, ya que la mayoría de los recursos describen al Método de Mejora de Gradiente como uno de los mejores para los desafíos de clasificación, mientras que los vectores de soporte son considerados buenos modelos, pero con ciertas restricciones o desventajas que, aunque no sean tan evidentes en este caso, podrían volverse más relevantes si se cambian ciertos aspectos.

4.2 Análisis, ventajas, desventajas, posibles mejoras

Cuando hablamos de selección de modelos es importante mencionar que esto depende del tipo de datos que se van a analizar y el objetivo de los resultados que se busca obtener. En nuestro caso, concluimos que para una elección de 50 o menos variables a analizar es mejor utilizar el Gradient Boosting Classifier, ya que como se mostró en las tablas, contó con una mejor precisión y exactitud. Sin embargo, nos percatamos de que entre mayor era el número de variables, más mejoraron los resultados de evaluación en la Máquina de Vectores de Soporte. Y es que para elegir uno u otro modelo no solamente es necesario contemplar su desempeño, sino también el costo computacional y de tiempo que requeriría su implementación. Y es que una desventaja del Gradient Boosting Classifier es que conforme más aumenta la cantidad de variables con las que se está trabajando, se vuelve más complicado para el modelo encontrar los mejores parámetros. Esto causa que utilizar el método requiere de un mayor poder computacional para poder desarrollarlo, además de que toma mucho más tiempo comparado con la Máquina de Vectores de Soporte. Aplicado dentro del ámbito médico, no sería muy útil para casos de clasificación que requieran de un resultado inmediato. Es por esto por lo que en casos con una cantidad muy elevada de variables es más óptimo optar por la máquina de vectores. Viéndolo desde un caso en el cual existe una cantidad más limitada de variables, es mejor elegir el GBC, ya que a pesar de contar con menos datos, aún es capaz de realizar modelos altamente eficientes en la

clasificación, comparándolo con la significativa disminución de desempeño que tienen las máquinas de vectores.

Si analizamos el caso desde la perspectiva de un médico, una desventaja o posible dificultad del modelo de Máquina de Vectores de Soporte podría ser la interpretación de los resultados. Al modificar los planos para hacer la clasificación de los datos, puede llegar a ser complicado para los médicos traducir estas etiquetas generadas con hiperplanos a términos médicos o clínicos. Lo mismo sucede con el Gradient Boosting Classifier, al ser armado por diferentes modelos mejorando su precisión iterativamente, puede llegar a ser complicado interpretar cuál es la contribución que cada modelo hace al resultado final.

Como posibles mejoras que pueden ser implementadas para esta investigación, sería intentar con aún menos variables predictoras y probar más iteraciones de búsqueda de hiperparametros. El impedimento a realizar esto en un inicio consiste en el coste de tiempo de ejecución para el plazo de investigación. Aún así, es posible que probando con más iteraciones y mayor variedad de parámetros por examinar en los modelos se pueda llegar a resultados similares usando menos información. Para los objetivos establecidos en este proyecto, los resultados son significativos con un coste de ejecución razonable.

5. Conclusiones

5.1 Conclusión - Sebastián Miramontes Soto

Durante este semestre y este proyecto, yo pude darme cuenta de cómo se puede vincular distintos modelos relacionados a la inteligencia artificial con casos reales o inclusive de la vida cotidiana, y aunque a veces pueden existir obstáculos, como lo puede ser una mala base de datos o una mala práctica, suele ser muy útil para las predicciones o clasificaciones en diferentes ámbitos. En el caso específico de este proyecto yo pude encontrar que muchos de los métodos pueden ser utilizados en el sector salud, pero para encontrar los mejores se tiene que hacer un estudio o a veces analizarlo de manera empírica. Los resultados mostrados

en nuestro caso particular fueron muy de la mano con las investigaciones o análisis realizados antes del código, por lo que podríamos decir que todo salió como se planeaba.

5.2 Conclusión - Raúl Correa Ocañas

En esta investigación he tenido la oportunidad de realizar un trabajo exhaustivo para la comparación de modelos de aprendizaje automático. Previo a este curso desconocía métodos de validación de modelos y algoritmos de optimización de hiperparametros. En cuanto a proyectos utilizando inteligencia artificial, considero de alto valor el implementar estas técnicas para comparar la mejor versión de cada modelo ante una misma base de prueba. Así mismo, aprendí sobre el muestreo de datos y la importancia de tener variables distribuidas para formar modelos que eviten sesgos al intentar optimizar métricas como *accuracy.* En esa misma nota, el aprender sobre otras métricas de evaluación para problemas de clasificación, especialmente la métrica de F1, me abrieron una perspectiva para encontrar casos en donde los mejores resultados no siempre implican tener los mejores modelos. Definitivamente, el conocimiento obtenido en esta unidad es crucial para modelar los distintos algoritmos de aprendizaje.

5.3 Conclusión - Ericka Sofia Rodriguez Sanchez

Durante el transcurso de esta materia pude percatarme de cómo mediante la inteligencia artificial, la ciencia de datos puede combinarse con muchas otras disciplinas. La creación de modelos de clasificación utilizando diferentes métodos de aprendizaje para máquinas puede ser útil para analizar datos de muchos tipos, desde aprobaciones en el banco o segmentación de público hasta clasificación de enfermedades. Además, logré comprender que no sólo es necesario el saber cómo escoger un modelo, sino que también la limpieza de datos y selección de variables juegan un papel muy importante en el proceso.

Gracias a este proyecto me di cuenta de las aplicaciones que esto puede tener en la medicina. Además, puedo concluir que los resultados fueron satisfactorios, al llegar a obtener modelos con grados de precisión y exactitud mayores al 90%. Me sirvió para reforzar los conocimientos adquiridos en clase, poniéndolos en práctica y profundizandolos más.

6. Referencias

Aracena, C., Villena, F., Van Der Huck Arias, F., & Dunstan, J. (2022). Aplicaciones de aprendizaje automático en salud. *Revista Médica Clínica las Condes*, *33*(6), 568-575. https://doi.org/10.1016/j.rmclc.2022.10.001

Hernández, R. (2022, 10 febrero). Aprendizaje automático aplicado al diagnóstico médico. *Encora*. https://www.encora.com/es/blog/machile-learning-applied-to-medial-diagnosis

Nelson, D. (2021, 1 marzo). *¿Qué es el aumento de gradiente?* Unite.AI. https://www.unite.ai/es/%C2%BFQu%C3%A9-es-el-aumento-de-gradiente%3F/

Raj, A. (2022, 6 agosto). Everything About Support Vector Classification — Above and Beyond. *Medium*. https://towardsdatascience.com/everything-about-svm-classification-above-and-beyond-cc665bfd993e

Rodriguez, C. C. (2021, 15 diciembre). Maquina de Soporte vectorial (SVM) - César Chique Rodriguez - Medium. *Medium*. https://medium.com/@csarchiquerodriguez/maquina-de-soporte-vectorial-svm-92e9f1b1b1ac

Tuychiev, B. (2023, 27 diciembre). *A Guide to The Gradient Boosting Algorithm*. https://www.datacamp.com/tutorial/guide-to-the-gradient-boosting-algorithm

Yu, W., Liu, T., Valdéz, R., Gwinn, M., & Khoury, M. J. (2010). Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. *BMC Medical Informatics And Decision Making*, *10*(1). https://doi.org/10.1186/1472-6947-10-16

Zhang, Z., Zhao, Y., Canes, A., Steinberg, D., & Lyashevska, O. (2019). Predictive analytics with gradient boosting in clinical medicine. *Annals Of Translational Medicine*, *7*(7), 152. https://doi.org/10.21037/atm.2019.03.29