

UNIVERSIDAD DEL BÍO BÍO  
FACULTAD DE CIENCIAS EMPRESARIALES  
DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN



UNIVERSIDAD DEL BÍO-BÍO

**“Reconocimiento de patrones de movimiento/actividad  
de trabajadores en obra basado en datos recolectados  
mediante smartphone”**

Proyecto de Título para optar al Título de Ingeniero Civil Informático

Haroldo Vivallo Ocampo

Profesor Guía: Pedro Campos Soto

Septiembre de 2019

Concepción – Chile

## Resumen

Este trabajo se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil Informática, en este proyecto titulado “Reconocimiento de patrones de movimiento/actividad de trabajadores en obra basados en datos recolectados mediante smartphone” se discuten y aplican diversas técnicas del Machine Learning (Aprendizaje de Máquinas), con las cuales se detectan patrones que permiten establecer los movimientos que realizan los trabajadores en una obra, con el fin de entregar lineamientos de cómo proceder a la hora de aplicar la tecnología a procesos críticos que suceden en una empresa, donde resulta relevante conocer las actividades que realizan sus trabajadores para tomar medidas que permitan generar espacios seguros y con un mayor desempeño.

Primero se introduce la problemática y las oportunidades que existen en la actualidad respecto del Machine Learning en todos los ámbitos en donde puede ser aplicado, particularmente en el área de la construcción (para los fines de este trabajo), se introducen las diferentes técnicas de Machine Learning (algoritmos) encontradas en la literatura aplicadas al reconocimiento de actividades. Se realizan una serie de experimentos en terreno, en los cuales se simulan las actividades que se desean estudiar, para esto se utiliza un smartphone el cual captura los datos de aceleración asociados a estas actividades. Luego, con este set de datos, se procede a realizar el entrenamiento de los diferentes algoritmos (KNN, DT, ANN, SVM), entrenamiento el cual permitirá generar modelos que pueden aplicarse a múltiples sets de datos, para determinar el movimiento.

Como resultado se obtiene 4 diferentes modelos los cuales entregan una precisión relativamente alta de todos los movimientos/actividades evaluadas, teniendo en cuenta que estos modelos seguirán siendo entrenados con set de datos más grandes para que los resultados mejoren sus predicciones.

Finalmente se presenta la aplicación web desarrollada (prototipo), que funciona como una prueba de concepto, la cual permite a un usuario utilizar datos de aceleración propios para determinar distintos movimientos mediante la aplicación de los distintos modelos que fueron generados en este trabajo.

## Tabla de contenido

Resumen .....	1
Tabla de contenido .....	2
Índice Figuras .....	4
Índice Tablas .....	5
1. Introducción .....	6
1.1. Motivación .....	6
1.2. Objetivos.....	7
Objetivo General .....	7
Objetivos Específicos.....	7
1.3. Estructura del Informe .....	7
2. Marco Teórico .....	8
2.1. Definición terminología básica.....	8
2.1.1. Acelerómetro .....	8
2.1.2. Machine Learning (ML) .....	9
2.1.3. Técnicas de Machine Learning .....	10
2.1.4. Entrenamiento .....	15
2.2. Estado del arte .....	16
2.2.1. Deslizamiento – tropiezo .....	16
2.2.2. Reconocimiento basado en smartphone .....	17
2.2.3. Sensores y clasificadores para reconocer actividad de maquinaria .....	18
3. Captura de datos .....	20
3.1. Hardware .....	20
3.2. Software.....	20
3.3. Actividades y movimientos del trabajador .....	23
3.3.1. Quieto (Reposo).....	23
3.3.2. Caminar .....	23
3.3.3. Subir/bajar escaleras .....	23
3.3.4. Tropezar .....	23
3.3.5. Resbalar .....	23
3.4. Simulación .....	24

3.4.1.	Escenario.....	24
3.4.2.	Simulación de accidente.....	25
4.	Análisis y procesamiento de datos.....	25
4.1.	VARIABLES SELECCIONADAS.....	25
4.1.1.	Root Mean Square.....	26
4.1.2.	Mean.....	26
4.1.3.	Interquartile Range.....	26
4.1.4.	Standard Deviation.....	27
4.2.	Procesamiento de datos.....	28
4.2.1.	Dataset.....	28
4.2.2.	Algoritmos de aprendizaje utilizados.....	30
5.	Reconocimiento de la actividad.....	31
5.1	Matriz de confusión.....	31
5.1.1	Árbol de decisión.....	32
5.1.2	K-Nearest Neighbor.....	32
5.1.3	Support Vector Machine.....	33
5.1.4	Artificial Neural Network.....	33
5.2	Resultados.....	34
6.	Prototipo de software.....	35
6.1	Objetivo del software.....	35
6.2	Descripción del Software.....	35
6.2.1	Requerimientos.....	35
6.2.2	Entorno de desarrollo.....	36
6.2.3	Caso de uso.....	36
6.2.4	Modelo arquitectónico.....	37
6.2.5	Funcionamiento.....	38
7.	Conclusiones.....	40
8.	Bibliografía.....	42

## Índice Figuras

Figura 1 - Datos de un acelerómetro de 3 ejes [5] .....	8
Figura 2 - Esquema del Aprendizaje Automático (fuente: Towards Data Science) .....	9
Figura 3 - Ejemplo de un Árbol de Decisión (fuente: datacamp.com) .....	11
Figura 4 - Ejemplo de KNN (fuente: datacamp.com) .....	12
Figura 5 - Ejemplo SVM con 2-dimanasiones (fuente: towardsdatascience.com) .....	13
Figura 6 - Ejemplo de la Estructura de una Red Neuronal .....	14
Figura 7 - Como trabaja una ANN (fuente: blog.goodaudience.com) .....	14
Figura 8 - Proceso Machine Learning.....	15
Figura 9 – Esquema de Red Neuronal usada para la clasificación (extraída de [5]) .....	16
Figura 10 - Ejemplo nivel de detalle .....	19
Figura 11 - Información del Acelerómetro (desde Accelerometer Meter) .....	20
Figura 12 - Captura de datos (Accelerometer Meter).....	21
Figura 13 - Archivo con datos del acelerómetro.....	22
Figura 14 - Dataset de Ejemplo (features.py) .....	28
Figura 15 - Indicaciones de uso del algoritmo features.py.....	29
Figura 16 - Jupyter Notebook.....	30
Figura 17 - Matriz de Confusión .....	31
Figura 18 - Arquitectura de la aplicación .....	37
Figura 19 - Aplicación Web - Principal .....	38
Figura 20 - Aplicación Web ( <i>dropdown</i> para seleccionar modelo) .....	38
Figura 21 - Aplicación Web - Resultados .....	39
Figura 22 - Aplicación Web – Administración .....	39

## Índice Tablas

Tabla 1 - Matriz de confusión para Caminar, Resbalar y Tropezar .....	17
Tabla 2 - Resultados de la predicción de actividades .....	18
Tabla 3 - Precisión de los clasificadores para cada nivel de detalle .....	19
Tabla 4 - Movimientos y actividades .....	24
Tabla 5 - Número de muestras del dataset .....	29
Tabla 6 - <i>Precision</i> - Árbol de Decisión.....	32
Tabla 7 - <i>Precision</i> KNN.....	32
Tabla 8 - <i>Precision</i> SVM.....	33
Tabla 9 - <i>Precision</i> ANN.....	33
Tabla 10 - Resultados .....	34

# 1. Introducción

## 1.1. Motivación

Hoy en día muchas de las cosas que nos rodean son “inteligentes”, el caso más emblemático es nuestro celular o “*smartphone*” y así existen un sinnúmero de dispositivos, que nos facilitan muchos aspectos de nuestra vida mediante el uso adecuado de la tecnología. Sin embargo, aún quedan muchas aplicaciones por explorar, por ejemplo, en la construcción, que es una industria sumamente importante para nuestro país, en el programa de productividad y construcción sustentable del gobierno, “Construye 2025” se afirma que: “la construcción es un sector relevante de la economía global. En Chile representa el 7,8% del PIB; el 8,4% del empleo; y cuenta con 30.000 empresas (CChC, 2014), que en su mayoría son PYMEs. Asimismo, aporta el 55% de la inversión total del país, de este porcentaje un tercio corresponde a edificaciones de carácter comercial, público y residencial, lo que equivale a una inversión cercana a los US\$ 10.000 millones al año.” (Extracto de informe Construye 2025, 2016, según lo descrito en [1]).

Actualmente, en esta área se presentan 3 ejes que resultan de una gran importancia: prevención de accidentes, enfermedades profesionales y productividad. Temas que resulta interesante abordar mediante el uso de la tecnología, automatizando procesos para generar información que sea de utilidad para empresas, particularmente del área de Construcción, que permitan tomar mejores decisiones. Si bien los 3 temas son importantes para el efecto de este proyecto se ha optado por centrarse en la prevención de accidentes, siendo uno de los factores clave para abordar esta problemática el conocer el comportamiento de los trabajadores en obra, para esto se ha decidido como primera instancia determinar los movimientos que se realizan y así determinar patrones de movimiento/actividad, que puedan servir de insumo para mejorar la predictibilidad de accidentes. La información se entregará mediante un sistema que reciba los datos y permita detectar los movimientos que efectúan los trabajadores, de acuerdo a diferentes niveles establecidos.

Para determinar estos patrones, primero se captura datos con un acelerómetro [2] y luego, se utilizan algoritmos de Machine Learning, los cuales son algoritmos derivados de la estadística y matemática que permiten analizar conjuntos de datos y reconocer patrones en ellos. Por medio de la utilización de un *smartphone*, el cual posee un acelerómetro que permite la captura de datos relacionado al movimiento, se entrenan modelos de clasificación del movimiento para posteriormente utilizar estos modelos para predecir resultados en casos nuevos. Mediante esta metodología se puede determinar desde movimientos como caminar, subir escaleras, etc [3] hasta incluso automatizar el análisis de la actividad de los trabajadores de acuerdo a su postura [4].

El interés del proyecto se enfoca en estudiar soluciones que puedan servir tanto como para productividad y accidentabilidad laboral en obras. Como etapa necesaria para

alcanzar las metas antes planteadas, en este Proyecto de Título se aborda el problema de detectar y analizar los movimientos que realizan los trabajadores en una obra, lo que podrá servir posteriormente para abordar temas de productividad y accidentabilidad laboral.

## **1.2. Objetivos**

Con el fin de satisfacer estas necesidades se definen los siguientes objetivos:

### **Objetivo General**

Desarrollar e implementar algoritmos de detección de movimiento de trabajadores en obra basado en los datos de aceleración entregados por el acelerómetro de un smartphone, que permitan obtener modelos de clasificación de estos movimientos.

### **Objetivos Específicos**

Aplicar diferentes técnicas de aprendizaje automático utilizando los datos obtenidos por el acelerómetro del smartphone.

Determinar distintos niveles de actividades asociadas al área de la construcción y prevención de accidentes laborales.

Captura de datos en terreno, asociados a los movimientos que se desea estudiar y predecir.

Implementar un prototipo de aplicación web que permite visualizar resultados (clasificación del movimiento) basados en algoritmos implementados y modelos obtenidos.

## **1.3. Estructura del Informe**

En el capítulo 2 se presenta el Marco Teórico, en donde se definen los conceptos básicos con los que se trabaja a lo largo del proyecto, ¿Qué es un acelerómetro?, Machine Learning y sus técnicas, y un breve análisis del estado del arte; luego, en el capítulo 3 se explica la metodología utilizada para la captura de los datos; en el capítulo 4, se detallan las actividades que se pretenden reconocer y el experimento de simulación llevado a cabo para la captura de los datos; en el capítulo 5, se explica el análisis y procesamiento de los datos que fueron capturados, y como se genera el set de datos; luego, en el capítulo 6, se indican los resultados obtenidos para cada algoritmo y una breve explicación de su obtención a modo de conclusión; posteriormente, en el capítulo 7 se explica en que consiste el prototipo realizado y se incluyen imágenes del producto final; finalmente en el capítulo 8 se presentan las conclusiones generales del proyecto desde el punto de vista académico y personal.



## 2. Marco Teórico

En este capítulo se detallan, de forma breve y concisa, aquellos conceptos claves que son la base con la cual se desarrolló el proyecto, y se exploran además algunos artículos que ofrecen propuestas interesantes para abordar esta problemática.

### 2.1. Definición terminología básica

#### 2.1.1. Acelerómetro

Los acelerómetros son dispositivos electromecánicos que captan las fuerzas de la aceleración, que se define como la variación de velocidad por unidad de tiempo de un objeto. Esta aceleración se mide generalmente en  $m/s^2$ . Los acelerómetros son útiles para medir, por ejemplo, vibraciones o la orientación. Estos dispositivos pueden medir la aceleración en uno, dos o tres ejes (ver Figura 1), y, debido al desarrollo tecnológico y al bajo costo de su producción, los acelerómetros de 3 ejes son actualmente los que más se utilizan, y están presentes en la gran mayoría de smartphones existentes en el mercado.

Debido a su bajo costo y a la facilidad de uso, en la actualidad se utilizan para un sinnúmero de cosas, desde aplicaciones para el celular, hasta para analizar el comportamiento de las personas. Cada movimiento que se haga, por pequeño que sea, genera una cierta aceleración que es captada por este sensor, y que puede ser analizada posteriormente.

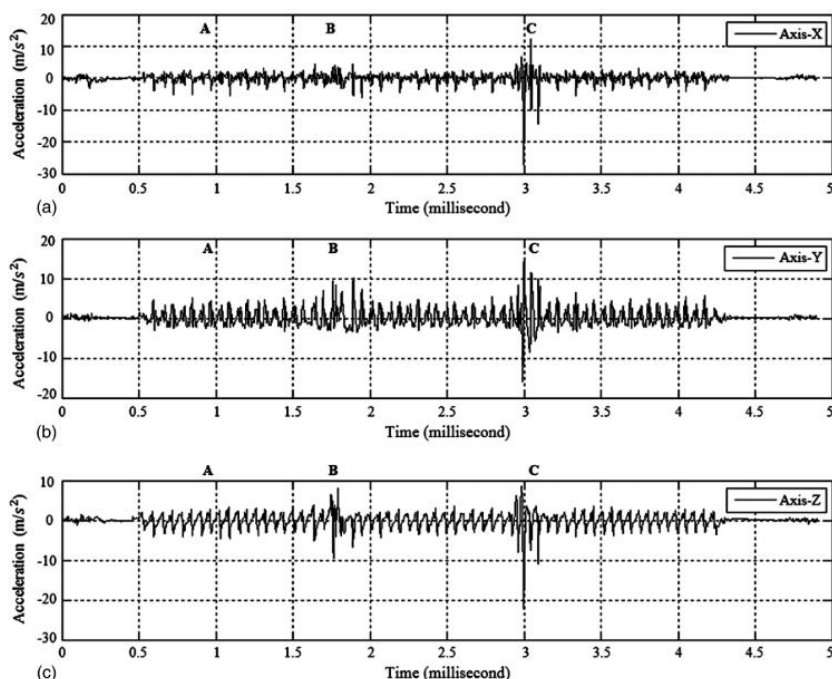


Figura 1 - Datos de un acelerómetro de 3 ejes [5]

### 2.1.2. Machine Learning (ML)

En palabras simples : “El Machine Learning es la ciencia (y arte) de programar computadoras con el objeto de que ellas puedan aprender de los datos” (traducido de [6]).

En términos más formales, corresponde al estudio científico de los algoritmos y modelos estadísticos que utilizan las computadoras para llevar a cabo una tarea específica sin la necesidad de instrucciones explícitas, sino que, más bien apoyándose en el uso de patrones e inferencia.

Cuando se habla de Machine Learning se habla generalmente de tres tipos de aprendizaje como se muestra en la Figura 2 **Error! Reference source not found.**, los cuales son:

- Aprendizaje Supervisado (Supervised Learning)

Que el aprendizaje sea supervisado se refiere a que los datos que servirán de input al algoritmo deben estar etiquetados, es decir, se le entregan **casos o ejemplos**, en este caso los atributos o características de cada caso o ejemplo, relacionadas con algo que se desea estimar o predecir, y también se le entregan los valores **de resultado para esos casos**, que corresponden a las etiquetas respectivas. De esta forma se le indica explícitamente a la máquina cual es el comportamiento esperado.

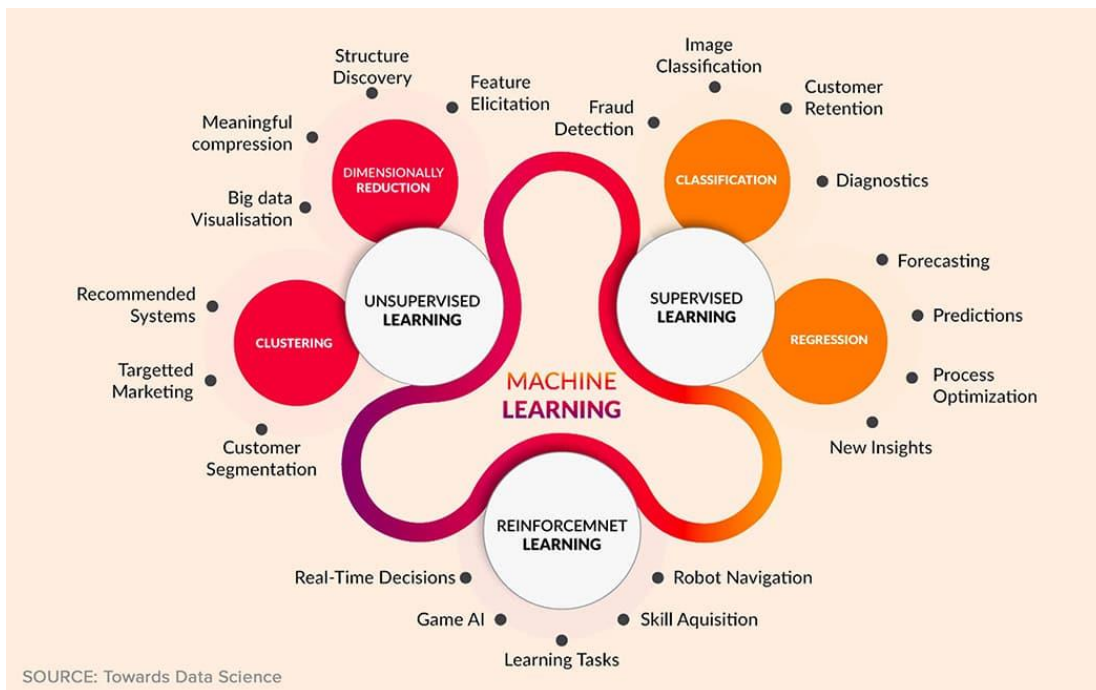


Figura 2 - Esquema del Aprendizaje Automático (fuente: Towards Data Science)

Dentro del aprendizaje supervisado encontramos dos tipos:

- Regresión

Al utilizar la regresión se espera obtener un resultado **numérico** dentro de un conjunto **infinito** de valores.

- Clasificación

Al utilizar la clasificación el resultado esperado corresponde **a una clase**, que se encuentra dentro de un número **finito** de clases.

Para este trabajo se utiliza el aprendizaje supervisado utilizando algoritmos de clasificación.

- Aprendizaje No Supervisado (Unsupervised Learning)

En el aprendizaje no supervisado, solo se definen los atributos o características de los datos, pero no se define ninguna etiqueta (valores de resultado definidos para los datos de entrada), por lo que es el algoritmo el encargado de agrupar de la manera que estime conveniente.

- Aprendizaje Reforzado (Reinforcement Learning)

En el aprendizaje reforzado, la máquina “aprende” de acuerdo con los éxitos o fracasos que tenga, es decir, evalúa las acciones llevadas a cabo y de acuerdo a si tiene éxito o fracasa vuelve a plantearse un modelo teniendo en cuenta los resultados anteriores, de esta forma, cada nuevo resultado va mejorando con cada iteración.

### 2.1.3. Técnicas de Machine Learning

Existen una gran variedad de técnicas asociadas al Machine Learning, que responden a la cantidad de problemáticas que se pueden resolver, en esta instancia, enfocados en resolver el problema del reconocimiento de actividades [3] se opta por utilizar los siguientes algoritmos:

#### Árbol de Decisión (DT)

Un árbol de decisión es un algoritmo de aprendizaje supervisado que luce como un árbol invertido, donde cada nodo representa una variable predictora (característica), las ramas representan decisiones (reglas) y cada nodo hoja representa la salida (resultado), ver Figura 3.

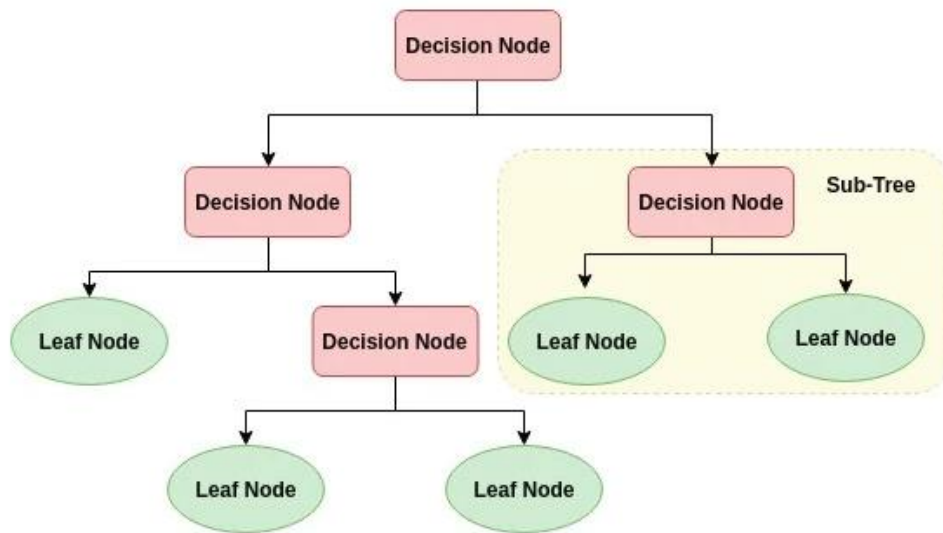


Figura 3 - Ejemplo de un Árbol de Decisión (fuente: datacamp.com)

El árbol de decisión posee la siguiente estructura:

- Un nodo raíz: El punto de partida del árbol, aquí se realiza la primera división.
- Nodos internos: Cada uno de estos nodos representa un punto de decisión que eventualmente lleva a la predicción del resultado.
- Nodos hoja o terminales: Representan la clase final (predicción) de la salida.

El algoritmo del árbol de decisión opera de la siguiente manera:

1. Selecciona la característica que mejor clasifique el dataset en la clase deseada y asigna esa característica el nodo raíz.
2. Recorre los demás nodos, mientras realiza decisiones relevantes en cada nodo de forma que cada nodo interno clasifique de la mejor manera los datos.
3. Repite el proceso.

## K-Nearest Neighbors (KNN)

El algoritmo del vecino(s) más cercano, es uno de los más simples, versátiles y conocidos en el área de Machine Learning, es un algoritmo no-paramétrico, lo que quiere decir que la estructura de su modelo está determinada por el set de datos.

En el KNN, K corresponde al número de los vecinos más cercanos, este número es el factor clave a la hora de decidir.

Los pasos que sigue este algoritmo son:

1. Calcular distancia (hacia los vecinos)
2. Encontrar aquellos puntos más cercanos (vecinos más cercanos)
3. Votar para escoger etiquetas.

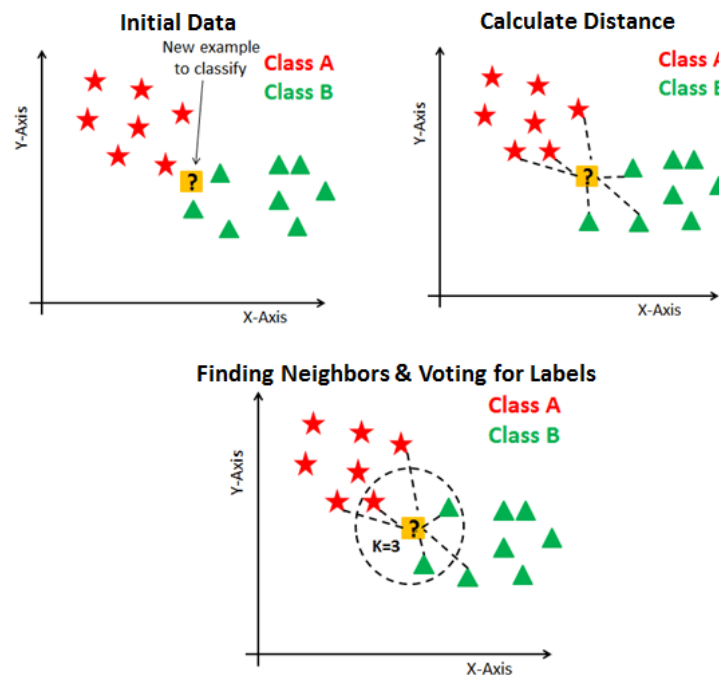


Figura 4 - Ejemplo de KNN (fuente: datacamp.com)

## Support Vector Machine (SVM)

El objetivo del algoritmo SVM es encontrar un hiperplano en un espacio n-dimensional, donde n representa el número de características, que clasifique de manera distintiva los puntos del set de datos.

Al separar los puntos del set de datos de acuerdo con sus clases, podrían existir múltiples hiperplanos entre ellos, el objetivo es encontrar un plano que maximice el margen, es decir, la distancia máxima entre los puntos de datos de las clases.

Por ejemplo, para cierto conjunto de datos y 2 clases (2 dimensiones), tenemos:

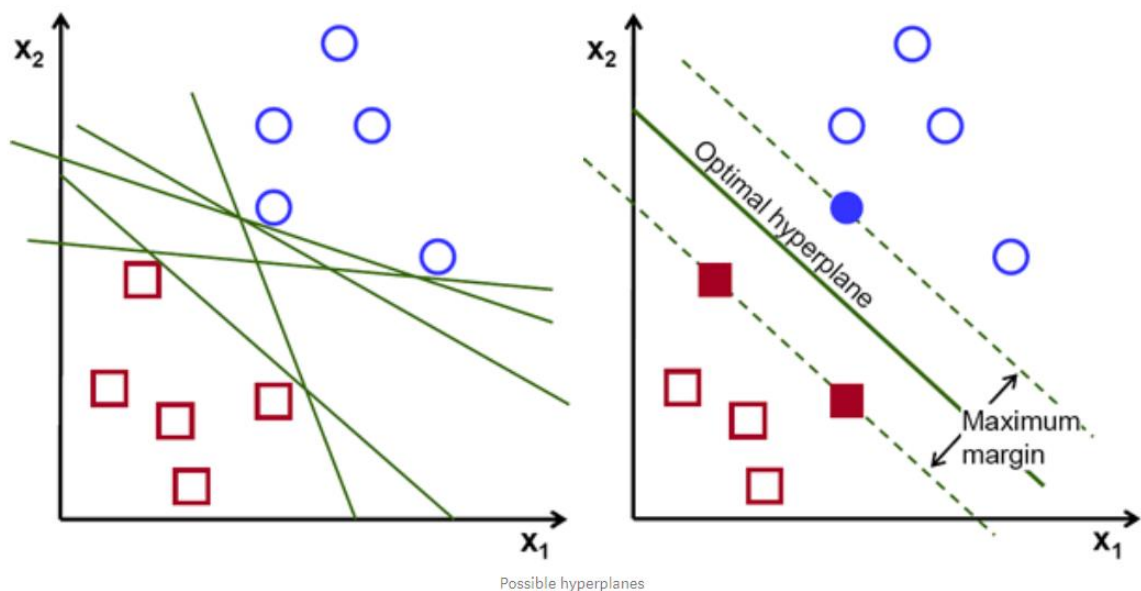


Figura 5 - Ejemplo SVM con 2-dimensiones (fuente: towardsdatascience.com)

Los hiperplanos son fronteras de decisión que ayudan a clasificar los datos, puntos que caen a uno u otro lado del hiperplano se le pueden atribuir clases distintas. Además, la dimensión del hiperplano depende de las características escogidas, si solo hay 2, el hiperplano será una recta, si existieran 3 sería un plano, para mayores dimensiones resulta más difícil imaginárselo.

Los vectores de apoyo (support vectors) son puntos que están cerca del hiperplano y que influyen la posición y orientación de este, al usar estos vectores, se puede maximizar el margen del clasificador.

## Red Neuronal Artificial (ANN)

Estos algoritmos están inspirados en las reacciones sinápticas que ocurren entre las neuronas, y que permiten a un sistema “aprender” para realizar determinadas tareas. La estructura de una Red Neuronal Artificial corresponde a grupo de “neuronas” conectadas que están organizadas en 3 capas como se observa en la Figura 6.

Estas capas son:

1. Input layer: corresponde a la capa donde se ingresan los datos, para posteriormente sean procesados por las siguientes capas de neuronas artificiales.
2. Hidden layer: La capa que se encuentra entre la capa de entrada y la capa de salida, aquí las neuronas asignan peso a los datos y producen un resultado de acuerdo con una función de activación dada, Figura 7 .
3. Output Layer: La última capa donde se produce el resultado.

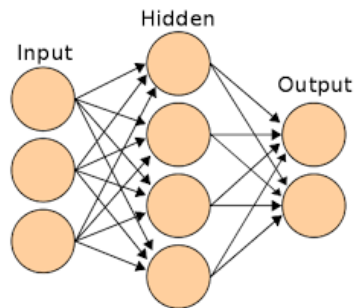


Figura 6 - Ejemplo de la Estructura de una Red Neuronal

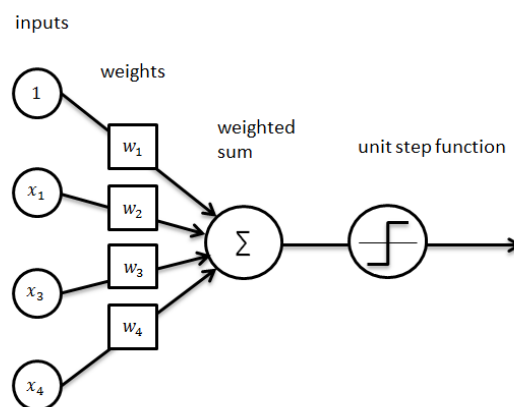


Figura 7 - Como trabaja una ANN (fuente: [blog.goodaudience.com](http://blog.goodaudience.com))

### 2.1.4. Entrenamiento

“Entrenar” un modelo de Machine Learning se refiere al proceso en el cual, se alimenta (se le proporcionan datos) a un algoritmo de Machine Learning, los datos entregados deben tener relación con el problema que se busca solucionar y deben ser una gran cantidad si se quieren obtener resultados óptimos.

El algoritmo de Machine Learning (o algoritmo de aprendizaje) se encarga de encontrar patrones en los datos de entrenamiento que están dados por los atributos de los datos que han sido ingresados (input de datos) y la respuesta que se desea predecir, y este genera un modelo que captura dichos patrones, luego este modelo se evalúa, y se puede mejorar de acuerdo a los resultados que se buscan, este es un proceso iterativo como se observa en la Figura 8.

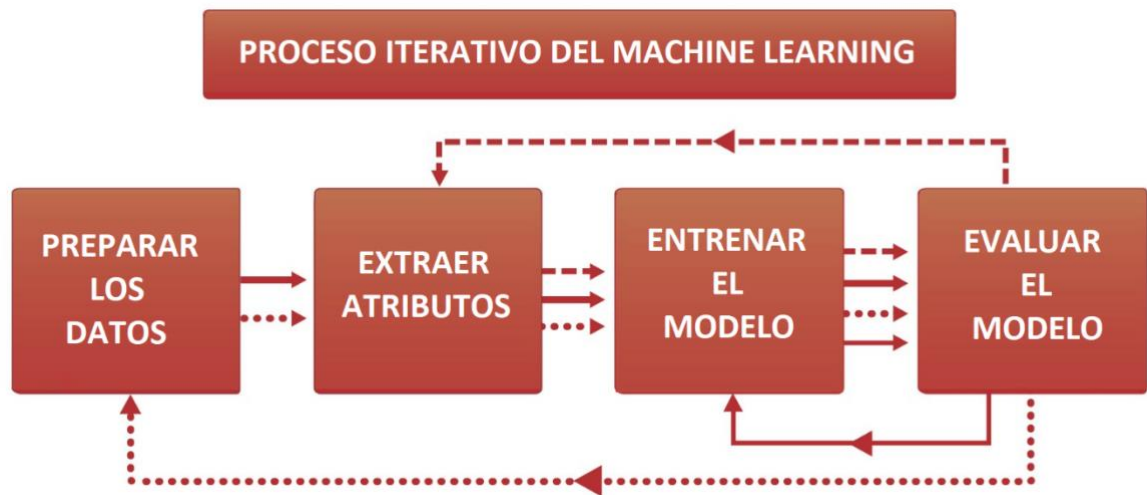


Figura 8 - Proceso Machine Learning



## 2.2. Estado del arte

Existen varios estudios en los cuales se han utilizado datos recopilados mediante *smartphones* para realizar análisis de reconocimiento de actividades en humanos [3], [7], [8], aprovechando que estos dispositivos traen incorporado un acelerómetro. Gracias a este sensor, y técnicas de ML, se ha podido detectar y reconocer la actividad en cuestión con una alta precisión, en algunos casos, sobre el 90%. A continuación, se presentan algunos de estos artículos mencionando los puntos más importantes.

### 2.2.1. Deslizamiento - tropiezo

En el artículo [5], los autores utilizan un acelerómetro en 3 ejes para determinar deslizamientos y tropiezos que sufre un trabajador.

El experimento consiste en someter a un individuo a caminar por una zona previamente establecida la cual tiene dos puntos importantes (A y B), el punto A corresponde a una zona donde el individuo debe deslizarse y el punto B donde debe tropezarse. El resto del trayecto debe hacerlo caminando, los datos son recolectados por un acelerómetro de 3 ejes, mediante un dispositivo que lleva amarrado a la cintura en la parte posterior. Con los datos obtenidos del experimento se utiliza una red neuronal artificial (Matlab), para clasificar los 3 tipos de movimiento (caminar, deslizarse y tropezarse). En la Figura 9, se observa el modelo de la red neuronal utilizada y las variables seleccionadas por los autores: media (Mean), desviación estándar (Std) y el valor más alto (Peak).

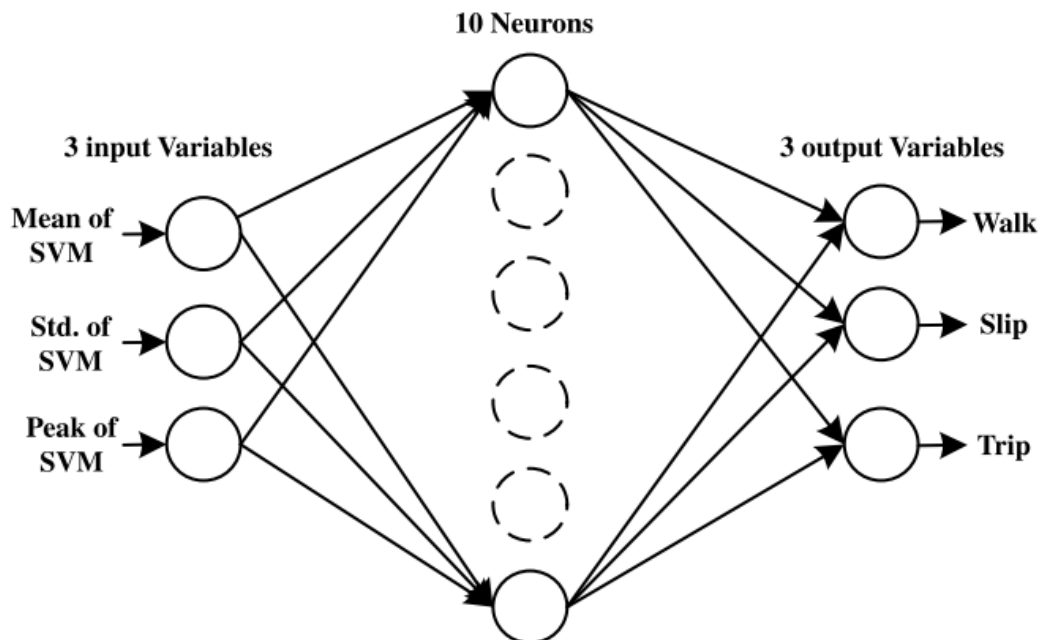


Figura 9 – Esquema de Red Neuronal usada para la clasificación (extraída de [5])

Los resultados que se obtuvieron se muestran en Tabla 1.

Tabla 1 - Matriz de confusión para Caminar, Resbalar y Tropezar

Clase de Salida	Clase objetivo			Precisión (%)
	Caminar	Resbalar	Tropezar	
<i>Caminar</i>	477	0	0	100
<i>Resbalar</i>	4	43	2	88
<i>Tropezar</i>	3	0	50	94

### 2.2.2. Reconocimiento basado en smartphone

En este artículo [9] se intenta reconocer la actividad de un trabajador en una obra, utilizando un smartphone, específicamente los sensores acelerómetro y giroscopio, con esto se pretende reconocer entre 8 diferentes actividades asociadas a una construcción:

1. Standing
2. Walking
3. Squatting
4. Cleaning up the template
5. Fetching and placing rebar
6. Locating the rebar
7. Binding rebar
8. Placing concrete pads

El experimento lo realizaron 9 individuos (5 hombres y 4 mujeres), los cuales tenían acoplados 2 smartphones a su cuerpo, uno en su muñeca derecha, y el otro en su pierna derecha. Estos individuos tuvieron que emular las 8 actividades mencionadas para recolectar los datos asociados a cada una de ellas.

Con esta información se utilizó un árbol de decisión (para la clasificación) debido a que este modelo siempre se ha usado en estudios de reconocimiento de actividad humana, los resultados obtenidos indican que se obtuvo una alta precisión en la clasificación de las actividades, como se observa en la Tabla 2.

Tabla 2 - Resultados de la predicción de actividades

<b>Activities</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
Standing	1.00	1.00	1.00
Walking	0.98	0.97	0.97
Squatting	0.93	0.99	0.96
Cleaning up the template	0.96	0.99	0.97
Fetching and placing rebar	0.98	1.00	0.99
Locating the rebar	0.76	0.82	0.79
Binding rebar	0.91	0.85	0.88
Placing concrete pads	0.92	0.82	0.87

### 2.2.3. Sensores y clasificadores para reconocer actividad de maquinaria

En este artículo [10] se intenta reconocer actividad, pero de maquinaria de la construcción, para lo cual se utilizan sensores, nuevamente acelerómetros, y además GPS, para conocer la localización de la máquina, ambos sensores se acoplan a las cabinas de la máquina.

La idea es medir los estados en que la maquina se encuentra:

1. Engine Off
2. Engine On
3. Idle
4. Busy
5. Moving and Scooping
6. Moving and Dumping
7. Scooping
8. Moving
9. Dumping

Con los datos obtenidos se utilizan diversos clasificadores (ANN, DT, KNN, LR SVM), los cuales entregan resultados con una alta precisión.

### Level of Detail (Nivel de detalle)

Para definir de forma correcta un movimiento o actividad, es necesario establecer su nivel de detalle, esto quiere decir, ser capaz de descomponer actividades en distintos niveles de acuerdo con su complejidad. La idea es partir de un concepto general e ir descomponiendo en actividades más particulares como se observa a continuación en la Figura 10.

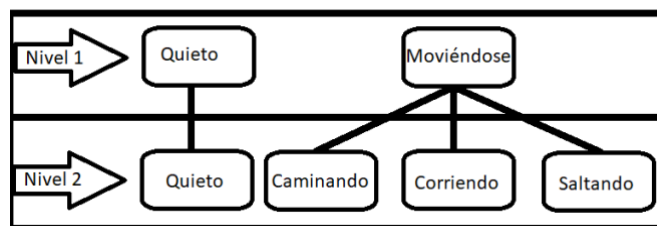


Figura 10 - Ejemplo nivel de detalle

Los resultados, al aplicar los clasificadores seleccionados, se muestran en la Tabla 3.

Tabla 3 - Precisión de los clasificadores para cada nivel de detalle

	Classifier	Accuracy (%)
Level 2	ANN	98.59
	DT	97.40
	KNN	97.65
	LR	96.93
	SVM	96.71
Level 3	ANN	81.30
	DT	81.21
	KNN	80.51
	LR	77.58
	SVM	78.03
Level 4	ANN	86.09
	DT	73.78
	KNN	84.20
	LR	84.42
	SVM	78.58

### 3. Captura de datos

En este capítulo se explica la metodología utilizada para realizar la captura de los datos, se comienza indicando el hardware y software utilizados, se mencionan las actividades y movimientos seleccionados y se explica cómo se realizó la simulación para la obtención de estos datos.

#### 3.1. Hardware

La captura de los datos se realiza con un smartphone marca *Samsung* modelo *Galaxy J5 (2016)*, el cual sólo posee acelerómetro y sensor de proximidad, a diferencia de equipos más modernos que incluyen además giroscopio. En la Figura 11 **Error! Reference source not found.** se especifica la información del acelerómetro.

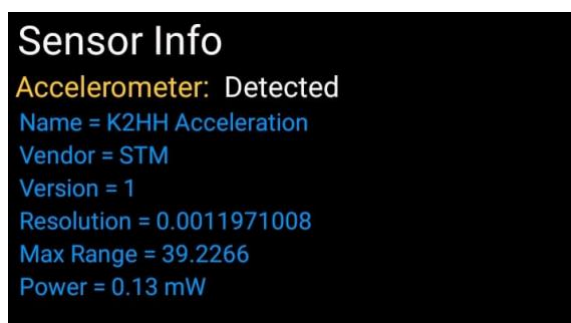


Figura 11 - Información del Acelerómetro (desde Accelerometer Meter)

#### 3.2. Software

Para el desarrollo del proyecto se opta por utilizar la aplicación Android Accelerometer Meter (aplicación de uso gratuito en Play Store), esta aplicación es capaz de capturar los datos de la aceleración a la cual está sujeta el dispositivo, en sus 3 ejes (X, Y, Z), esta información es mostrada en pantalla a través de un gráfico y además permite almacenar estos datos en la memoria interna del equipo. En la Figura 12 se puede observar el gráfico mencionado.

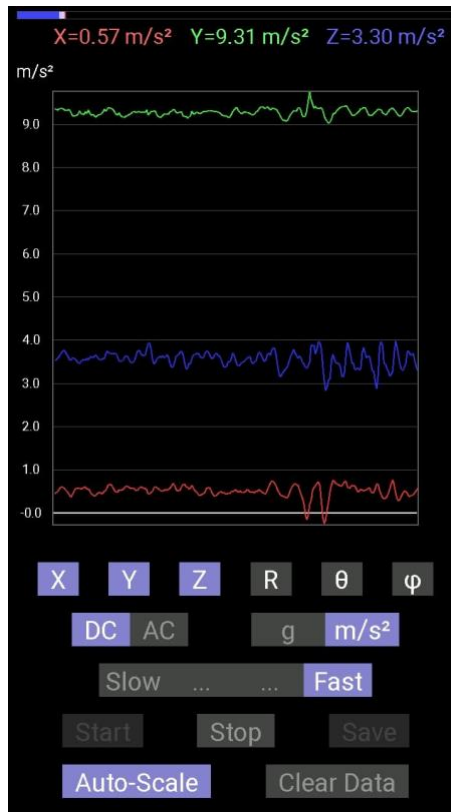


Figura 12 - Captura de datos (Accelerometer Meter)

La información que se almacena en el equipo corresponde a un archivo de texto, dentro del cual se almacena lo siguiente:

- El tiempo en segundos (tiempo en el cual se capturan los datos).
- Datos de aceleración en cada uno de los ejes (X, Y, Z; en  $m/s^2$ .)
- Los datos de aceleración de cada uno de los ejes utilizando coordenadas polares (phi, theta, R).

En la Figura 13 se muestra un archivo de texto, el cual contiene los datos mencionados anteriormente a modo de ejemplo.

```

bajar_escalera_mano_4.txt
Keuwl Accelerometer Data File
2019-04-23 13-21-05

Time (s), X (m/s2), Y (m/s2), Z (m/s2), R (m/s2), Theta (deg), Phi (deg)
0.000000,-2.541445,2.912546,8.158242,9.027668,25.352150,131.107468
0.0075070,-2.547431,2.928109,8.202535,9.074404,25.321793,131.022995
0.0167885,-2.477999,3.007117,8.294712,9.164359,25.162500,129.490036
0.0261708,-2.305616,3.124433,8.463503,9.311758,24.645538,126.424767
0.0357578,-2.149993,3.318364,8.528147,9.400175,24.874302,122.939514
0.0457536,-2.030283,3.491943,8.526949,9.435283,25.347212,120.174545
0.0558793,-1.932121,3.621230,8.647857,9.572452,25.389906,118.082314
0.0660125,-1.878251,3.746926,8.775947,9.725457,25.528830,116.623596
0.0758410,-1.960851,3.563769,8.905233,9.790229,24.549290,118.820396
0.0858524,-2.157176,3.244143,8.695741,9.528577,24.133410,123.621704
0.0956398,-2.437297,2.857480,8.166622,8.988845,24.697205,130.462631
0.1058643,-2.566584,2.770091,7.588422,8.476138,26.457016,132.816162
0.1159708,-2.458845,2.974796,7.586028,8.511353,26.965036,129.575760
0.1258418,-2.130840,3.221398,8.362947,9.211773,24.789518,123.483208
0.1359018,-1.686715,3.344700,9.526528,10.236541,21.465286,116.761574
0.1456951,-1.108515,3.390190,10.307038,10.906752,19.088516,108.106575
0.1558381,-0.967257,3.240552,10.625467,11.150665,17.655020,106.619598
0.1669212,-1.048660,3.045424,10.546458,11.027333,16.982803,109.000618
0.1758374,-1.380257,2.833538,10.569203,11.029148,16.605053,115.971382
0.1858360,-1.543063,3.365051,12.204443,12.753550,16.874115,114.634094
0.1956517,-0.855927,3.365051,13.478158,13.918222,14.446245,104.271004
0.2058436,0.175974,4.074931,12.010512,12.684181,18.757322,87.527245
0.2158744,0.920571,4.118027,10.137050,10.980227,22.600100,77.398911
0.2258129,1.074997,4.541800,8.492233,9.690284,28.792971,76.683746
0.2358344,1.074997,4.595670,7.167043,8.581510,33.366173,76.834373
0.2456442,0.994791,3.864242,7.167043,8.202955,29.106771,75.563530

```

Figura 13 - Archivo con datos del acelerómetro

Para efectos del trabajo se utilizan sólo los datos de tiempo y aceleración en los ejes X, Y, Z, se excluyen los datos en coordenadas polares.

## Frecuencia

La aplicación permite capturar datos en 4 frecuencias diferentes, las cuales están dadas al modelo de acelerómetro presente en el smartphone:

- 1-2 Hz (Slow)
- 3 Hz
- 50 Hz
- 100 Hz (Fast)

Para el fin de este proyecto la frecuencia óptima se establece a 100 Hz, es decir, 100 muestras por segundo, lo que se traduce en un mayor volumen de datos respecto del movimiento que está siendo capturado.

### **3.3. Actividades y movimientos del trabajador**

Las actividades seleccionadas para el proyecto responden a la finalidad de detectar, en una primera instancia, aquellos movimientos más sencillos y generales, para luego continuar con movimientos más específicos y enfocados en el área de prevención de accidentes del rubro de la construcción. Bajo esta misma lógica se pueden definir múltiples movimientos para ámbitos como la productividad en una obra, reconociendo movimientos más específicos aún (por ejemplo, movimientos de alguna profesión en particular, como pintores, albañiles, etc.) que también es muy interesante y responde también a necesidades reales de las empresas, sin embargo, para el fin de este proyecto se consideran sólo los movimientos mencionados a continuación:

#### **3.3.1. Quieto (Reposo)**

Esta actividad se refiere a cuando el operario está quieto, ya sea de pie o sentado, y no realiza ningún otro movimiento. También puede darse “entre” otros movimientos, por ejemplo, entre caminar y subir escaleras puede existir un tiempo de reposo.

#### **3.3.2. Caminar**

Esta actividad se refiere a cuando el operario se encuentra en movimiento, es decir cuando se mueve (camina) de un punto **A** hacia un punto **B** a una velocidad constante.

#### **3.3.3. Subir/bajar escaleras**

Esta actividad se refiere a cuando el operario se encuentra en proceso de subir o bajar una escalera, el entrenamiento de los modelos se realiza con escaleras de diversa índole.

#### **3.3.4. Tropezar**

Este movimiento se refiere a cuando el operario sufre un tropiezo posterior a una actividad de “caminar” o “subir escaleras”.

#### **3.3.5. Resbalar**

Este movimiento se refiere a cuando el operario se resbala, posterior a una actividad de “caminar” o “subir escaleras”.



Tabla 4 – Movimientos y actividades

<b>Actividad</b>
Quieto
Caminar
Subir Escaleras
Bajar Escaleras
Tropezar
Resbalar

Ya con estos movimientos/actividades definidas se procede a explicar los experimentos realizados para la captura de los datos necesarios para entrenar un modelo que permita predecir comportamientos en base a patrones detectados.

### **3.4. Simulación**

La simulación de los movimientos y actividades para la captura de los datos se divide en dos partes, de acuerdo con la naturaleza de la actividad que se pretende capturar.

La primera parte, corresponde al proceso de captura de datos de las actividades: quieto, caminar, subir escaleras y bajar escalera, estas actividades no requieren una preparación previa, ya que basta con realizar estas actividades en situaciones cotidianas. Sin, embargo para los eventos de tropezar y resbalar, se requiere una preparación previa ya que consiste en movimientos más complejos y que pueden presentar cierto grado de peligro, es por esto que debe prepararse un entorno seguro en el cual realizar las simulaciones para evitar cualquier posible accidente, por lo cual se dispone de un espacio adecuado y se toman las medidas pertinentes para la correcta ejecución del movimiento.

#### **3.4.1. Escenario**

Para la captura de los movimientos: tropezar y resbalar, se realizó la simulación en el exterior, en donde se definió un espacio de aproximadamente 6 x 3 metros. Se dispone una colchoneta y un pequeño obstáculo, para el caso de tropezar, y sólo la colchoneta para el caso de resbalar. Para el registro de la simulación se cuenta con 2 smartphones, uno el cual serviría para la captura de datos y otro con el cual se graba el movimiento para luego etiquetar los datos de acuerdo con el movimiento que se realizó. Lo complejo en esta instancia es simular adecuadamente ambos eventos (tropezar y resbalar) y además realizar un etiquetado correcto de estos datos, para lo cual se realiza un trabajo posterior en el cual se debe sincronizar la grabación de video con los datos obtenidos, de acuerdo al tiempo en que se desarrolló el movimiento.

### 3.4.2. Simulación de accidente

La simulación se realiza con 2 individuos encargados de realizar las acciones de tropezar y resbalar, para esto en el espacio asignado se definieron dos casos:

- La simulación del tropiezo, la cual consiste en que el individuo deba caminar hacia el obstáculo previamente colocado, y simular un tropiezo. Pudiendo eventualmente caer o no sobre la colchoneta.
- La simulación de resbalar, en la cual el individuo debe simular un deslizamiento y caer hacia atrás sobre la colchoneta.

El proceso de la simulación comienza con ejecutar la aplicación Accelerometer Meter, e iniciar la captura de datos, al mismo tiempo que se inicia la grabación en video.

El smartphone se introduce en el bolsillo derecho del individuo, este debe caminar y luego realizar la simulación (resbalar o tropezar), para luego seguir caminando, este proceso se repite 12 veces para cada una de las actividades.

Lo que da un total de 24 mediciones de “Tropezar” y 24 mediciones para “Resbalar”, que deben ser procesadas para el cálculo de las variables que se definen en la sección 4.1.

## 4. Análisis y procesamiento de datos

En este capítulo se presentan las variables escogidas para el entrenamiento, luego se explica el funcionamiento del algoritmo implementado para la generación del dataset y finalmente se explica cómo se aplican los diferentes algoritmos sobre este.

### 4.1. Variables seleccionadas

Para la elaboración del dataset de entrenamiento del modelo de reconocimiento de patrones, se utilizan las siguientes variables:

- Root Mean Square (RMS)
- Mean
- Interquartile Range (IQR)
- Standard Deviation

Las cuales se escogieron de acuerdo con la literatura revisada, siendo estas variables las que generalmente se usan para reconocimiento de actividades mediante el uso de acelerómetro[11], [12] y [13]. Estas variables se calcularon para cada eje de la aceleración obtenida mediante el acelerómetro y se detallan a continuación.

#### 4.1.1. Root Mean Square

Corresponde a la raíz de la **media** al cuadrado, como muestra la ecuación (6):

$$RMS = \sqrt{\frac{\sum_1^n x_i^2}{n}} \quad (1)$$

Donde  $n$  es el número de datos y  $x_i$  corresponde a los valores de la aceleración en sus distintos ejes  $[a_{x_i}, a_{y_i}, a_{z_i}]$ .

#### 4.1.2. Mean

La media (mean) es usada generalmente para suavizar el dataset eliminando puntos muy alejados (peaks) y ruido de los datos, la ecuación es la siguiente:

$$mean = \frac{1}{n} \sum_1^n s_i \quad (2)$$

Donde  $n$  es el número de datos y  $s_i$  corresponde a los valores de la aceleración en sus distintos ejes  $[a_{x_i}, a_{y_i}, a_{z_i}]$ .

#### 4.1.3. Interquartile Range

El rango intercuartil, que se conoce también como la distancia interior, es la diferencia entre los cuartiles superior e inferior, es decir, se dividen los datos en 2 grupos (alto y bajo) de igual tamaño. Con esto se pretende eliminar el ruido presente en los datos, ya que esta fórmula se ve afectada principalmente por aquellos valores que se encuentran en posiciones intermedias.

Primero se ordenan los datos y se calculan los cuartiles superior e inferior.

El cuartil superior se calcula:

$$Q_s = 3 \left( \frac{n}{4} \right) - 1 \quad (3)$$

El cuartil inferior se calcula:

$$Q_i = \left(\frac{n}{4}\right) \quad (4)$$

Donde  $n$  es el número de datos a evaluar.

(En este caso debido a la cantidad de datos ( $n$ ) escogida, el valor entre paréntesis se aproxima al valor entero superior)

Para calcular el IQR:

$$IQR = Q_s - Q_i \quad (5)$$

#### 4.1.4. Standard Deviation

La desviación estándar (standard deviation) se utiliza generalmente para representar el grado de dispersión del dataset completo eliminando la variabilidad individual. Su fórmula es:

$$std = \sqrt{\frac{1}{n-1} \sum_1^n (s_i - mean)^2} \quad (6)$$

Donde  $n$  es el número de datos y  $S_i$  corresponde a los valores de la aceleración en sus distintos ejes  $[a_{x_i}, a_{y_i}, a_{z_i}]$ .

El cálculo de cada una de estas fórmulas se realiza para los valores de aceleración de cada eje (X, Y, Z) obtenidos por el acelerómetro, del cual se selecciona un rango previamente definido, en este caso particular, el rango corresponde a 10 capturas de datos, es decir, por cada 10 datos capturados se realizan estos cálculos.

## 4.2. Procesamiento de datos

### 4.2.1. Dataset

El dataset utilizado corresponde a un archivo que contiene los datos de aceleración “procesados”, es decir, el cálculo de las variables seleccionadas, y para ello se implementa un script en Python denominado *features.py* el cual se encarga de recibir el archivo de texto con los datos de aceleración (señalados en sección 3.2), de los cuales se definen tramos de 10 registros cada uno y para estos se calculan las variables: RMS, Mean, IQR, Desviación Estándar definidas en la sección 4.1. En la Figura 14 se observa un extracto del dataset con sus variables y etiquetas.

```

resultados_label.txt
Time, RMS-x, RMS-y, RMS-z, Mean-x, Mean-y, Mean-z, Max-x, Max-y, Max-z, Min-x, Min-y, Min-z, IQR-x, IQR-y, IQR-z, Var-x, Var-y, Var-z, Std-x, Std-y, Std-z, Activity
0.05, 1.98, 1.99, 7.77, 1.4, 1.93, 7.74, 2.86, 2.55, 8.51, -2.21, 0.98, 6.89, 1.49, 0.78, 1.22, 2.16, 0.26, 0.42, 1.47, 0.51, 0.65, Bajar Escalera
0.15, 2.48, 3.13, 8.04, 1.32, 2.73, 8.01, 3.58, 4.95, 9.3, -3.93, 0.87, 6.91, 2.9, 3.43, 1.12, 4.91, 2.64, 0.62, 2.21, 1.62, 0.79, Bajar Escalera
0.25, 2.29, 0.59, 7.09, 2.28, 0.33, 6.86, 2.55, 1.32, 9.23, 2.13, -0.62, 4.18, 0.27, 0.63, 3.43, 0.82, 0.26, 3.5, 0.15, 0.51, 1.9, Bajar Escalera
0.35, 2.11, 2.52, 6.91, 2.09, -2.35, 6.64, 2.42, -0.62, 9.25, 1.49, -3.55, 4.62, 0.52, 1.41, 4.1, 0.1, 0.88, 4.09, 0.32, 0.94, 2.02, Bajar Escalera
0.45, 4.66, 3.5, 7.17, 3.61, -3.39, 7.1, 8.4, -2.19, 8.64, 0.12, -4.95, 5.51, 5.81, 1.44, 1.55, 9.69, 0.86, 1.11, 3.11, 0.93, 1.05, Bajar Escalera
0.55, 13.46, 0.85, 8.07, 13.37, -0.18, 8.04, 14.67, 0.62, 8.66, 9.67, -1.8, 7.04, 1.74, 1.33, 1.25, 2.81, 0.76, 0.44, 1.67, 0.87, 0.66, Bajar Escalera
0.65, 13.73, 0.8, 4.11, 13.69, -0.32, 3.83, 15.05, 0.62, 6.76, 12.29, -1.27, 2.52, 2.15, 1.65, 2.16, 1.12, 0.59, 2.43, 1.06, 0.77, 1.56, Bajar Escalera
0.75, 12.69, 1.5, 1.23, 12.63, -1.3, -0.48, 14.19, -0.37, 2.17, 10.96, -2.48, -1.63, 2.27, 1.47, 1.03, 1.48, 0.61, 1.44, 1.22, 0.78, 1.2, Bajar Escalera
0.85, 8.91, 3.72, 0.89, 8.88, -3.69, 0.51, 10.4, -2.85, 1.45, 8.0, -4.27, -0.56, 1.06, 0.82, 1.52, 0.59, 0.24, 0.58, 0.77, 0.49, 0.76, Bajar Escalera
0.95, 8.97, 3.3, 0.76, 8.97, -3.28, -0.7, 9.25, -2.85, -0.3, 8.43, -3.68, -1.16, 0.25, 0.73, 0.57, 0.07, 0.13, 0.1, 0.26, 0.36, 0.32, Bajar Escalera
1.05, 8.89, 3.17, 0.75, 8.87, -3.17, 0.58, 9.43, -2.99, 1.37, 7.99, -3.46, -0.05, 1.18, 0.22, 0.78, 0.35, 0.02, 0.25, 0.59, 0.14, 0.5, Bajar Escalera
1.15, 8.36, 3.71, 0.54, 8.36, -3.69, 0.37, 8.92, -3.24, 0.93, 7.64, -4.39, -0.24, 0.5, 0.61, 0.54, 0.14, 0.17, 0.17, 0.37, 0.41, 0.41, Bajar Escalera
1.25, 8.33, 5.32, 1.58, 8.28, -5.27, 1.18, 9.7, -4.41, 2.7, 7.41, -6.18, -0.15, 1.77, 1.33, 2.01, 0.86, 0.52, 1.22, 0.93, 0.72, 1.11, Bajar Escalera
1.35, 9.52, 6.18, 1.23, 9.5, -6.15, 0.78, 10.51, -5.57, 2.25, 8.67, -7.05, -0.62, 1.13, 1.22, 1.69, 0.41, 0.35, 1.0, 0.64, 0.59, 1.0, Bajar Escalera
1.45, 8.15, 6.73, 2.07, 8.1, -6.71, 1.87, 9.24, -6.01, 3.46, 6.67, -7.67, 0.89, 1.81, 0.87, 1.48, 0.99, 0.3, 0.86, 0.99, 0.55, 0.93, Bajar Escalera
1.55, 7.38, 6.19, 1.49, 7.36, -6.13, 1.23, 8.14, -5.3, 2.22, 6.24, -7.75, -0.32, 1.16, 1.54, 1.57, 0.39, 0.9, 0.79, 0.63, 0.95, 0.89, Bajar Escalera
1.65, 5.59, 6.24, 1.67, 5.58, -6.24, 1.62, 5.84, -5.76, 2.18, 5.23, -6.6, 0.85, 0.31, 0.16, 0.32, 0.04, 0.05, 0.19, 0.2, 0.23, 0.43, Bajar Escalera
1.75, 3.92, 6.36, 1.59, 3.75, -6.36, 1.55, 4.82, -6.17, 1.98, 1.47, -6.54, 0.81, 1.66, 0.16, 0.48, 1.39, 0.01, 0.14, 1.18, 0.11, 0.37, Bajar Escalera
1.85, 1.94, 17.97, 1.27, 1.82, -15.04, -0.41, 3.04, -4.07, 1.22, 1.02, -30.5, -1.91, 1.1, 19.28, 2.24, 0.49, 107.72, 1.61, 0.7, 10.38, 1.27, Bajar Escalera
1.95, 4.81, 9.18, 4.9, 1.56, -8.99, 3.04, 6.02, -6.75, 8.02, -6.23, -12.31, -1.99, 8.57, 3.33, 8.32, 23.06, 3.81, 16.34, 4.8, 1.95, 4.04, Bajar Escalera
2.05, 4.4, 9.25, 3.34, -4.29, -9.22, 0.28, -2.64, -8.37, 6.09, -5.61, -10.31, -3.68, 1.73, 1.36, 5.95, 1.08, 0.52, 12.3, 1.04, 0.72, 3.51, Bajar Escalera
2.15, 2.73, 9.98, 3.27, -2.63, -9.98, -2.52, -1.71, -9.22, -8.25, -3.85, -10.44, -5.74, 1.47, 0.34, 3.71, 0.61, 0.14, 4.84, 0.78, 0.37, 2.2, Bajar Escalera
2.25, 4.15, 18.68, 2.09, -4.14, -10.65, -1.44, -3.89, -9.08, 1.07, -4.47, -11.49, -3.25, 0.2, 1.49, 2.99, 0.04, 0.85, 2.53, 0.19, 0.92, 1.59, Bajar Escalera
2.35, 2.6, 10.83, 1.02, -2.53, -10.82, -0.86, -1.89, -8.99, 0.1, -3.7, -10.49, -1.83, 0.88, 0.82, 0.64, 0.4, 0.28, 0.34, 0.63, 0.53, 0.58, Bajar Escalera
2.45, 3.14, 7.61, 0.56, -3.13, -7.58, -0.51, -2.81, -6.33, -0.05, -3.33, -8.65, -0.84, 0.15, 1.36, 0.28, 0.02, 0.59, 0.06, 0.16, 0.77, 0.25, Bajar Escalera
2.55, 2.55, 5.53, 1.22, -2.47, -5.51, -1.03, -1.41, -5.02, -0.13, -3.3, -6.45, -2.03, 1.29, 0.79, 1.28, 0.49, 0.23, 0.46, 0.7, 0.48, 0.67, Bajar Escalera
2.65, 2.22, 7.62, 2.88, -2.15, -7.54, -2.82, -1.4, -6.04, -2.03, -2.89, -9.06, -3.81, 1.04, 2.29, 1.04, 0.32, 1.27, 0.4, 0.56, 1.13, 0.63, Bajar Escalera
2.75, 2.02, 8.74, 2.51, -1.97, -8.65, -2.44, -1.3, -6.54, -1.63, -2.7, -10.46, -3.57, 0.95, 2.05, 0.7, 0.26, 1.72, 0.37, 0.51, 1.31, 0.61, Bajar Escalera
2.85, 3.19, 9.88, 3.38, -3.04, -9.87, -3.35, -2.21, -9.31, -2.65, -4.9, -10.59, -3.74, 1.53, 0.77, 0.68, 1.02, 0.21, 0.18, 1.01, 0.46, 0.43, Bajar Escalera
    
```

Figura 14 - Dataset de Ejemplo (features.py)

Este procedimiento se realiza para cada uno de los archivos obtenidos de cada movimiento, posteriormente, se juntan todos en un solo dataset que es el que servirá para entrenar cada uno de los modelos, obtenidos de acuerdo a las técnicas mencionadas en la sección 2.1.3, que permiten determinar patrones en estas actividades. Con los modelos generados se logra reconocer estos patrones en **casos nuevos** (datos crudos de aceleración no etiquetados) y generar predicciones del movimiento que se está realizando.

```

features --bash-- 80x13
MacBook-Air-de-Haroldo:features haroldovivallo$ python features.py -h
usage: features.py [-h] -input INPUT_FILE [-output OUTPUT_FILE] [-lines LINES]
               -activity ACTIVITY

Calculate features from selected dataset

optional arguments:
  -h, --help            show this help message and exit
  -input INPUT_FILE     Dataset file
  -output OUTPUT_FILE   Features values file
  -lines LINES          Define the number of lines to calculate features
  -activity ACTIVITY    Define the activity of the dataset
MacBook-Air-de-Haroldo:features haroldovivallo$
    
```

Figura 15 - Indicaciones de uso del algoritmo features.py

Para utilizar este algoritmo se debe contar con un archivo de texto con los datos de la aceleración en los 3 ejes, en este caso obtenido por el smartphone, este archivo corresponde al parámetro “-input\_file”, luego se indica el archivo de salida, el cual contendrá el cálculo de las fórmulas indicadas en 254.1, este corresponde a “-output\_file”, luego se indica la cantidad de muestras a considerar (en este caso 10) mediante “-lines” y finalmente se indica el movimiento a que hacen referencia estos datos con “-activity”.

Finalmente, el dataset, luego de aplicar el algoritmo y juntar los resultados, queda conformado de acuerdo a lo que muestra la Tabla 5.

Tabla 5 - Número de muestras del dataset

Variables seleccionadas	Número de muestras por actividad y/o movimiento						Total
	Quieto	Caminar	Subir Escalera	Bajar Escalera	Tropezar	Resbalar	
<b>- IQR</b> <b>-RMS</b> <b>-Standard Deviation(std)</b> <b>-Mean</b>	208	211	201	206	101	98	<b>1025</b>

El dataset cuenta con un total de 1025 entradas, las cuales se utilizaron para entrenar el modelo, debido a la naturaleza de los movimientos, existe una menor cantidad obtenida en tropezar y resbalar, ya que la captura de estos requiere un esfuerzo mucho mayor que a los demás movimientos/actividades.

## 4.2.2. Algoritmos de aprendizaje utilizados

Luego de la extracción de las características y el etiquetado del set de datos, este se guarda en un archivo .csv, el cual contienen todas las actividades etiquetadas con sus respectivas características, datos que se obtienen al aplicar diferentes formulas sobre los datos originales captados por el acelerometro con le fin de que estos datos sean mas representativos.

Luego se inicia Jupyter notebook (ver Figura 16), que es un aplicación web de código abierto donde se pueden crear y compartir documentos que contiene código, ecuaciones, gráficos, texto , entre otros.

En esta aplicación se ejecuta código en lenguaje Python y se utiliza la librería SciKit Learn, ya que es ampliamente utilizada para Machine Learning [14], [15]. Esta librería incluye, entre otros, las implementaciones de los algoritmos de clasificación (DT, KNN, SVM, ANN) que se aplicaran a los set de datos previamente mencionados en la sección 4.2.1.

El proceso consiste en cargar el set de datos (log.csv) definir las columnas que contiene este dataset, luego generar set de datos de entrenamiento y de prueba, para lo cual se divide el set de datos original en una proporción de 80-20 (entrenamiento y prueba respectivamente). Luego, para cada algoritmo se crea un modelo de clasificación el cual se alimenta con los datos de entrenamiento y luego de prueba para evaluar el nivel de precisión del modelo, el resultado va a depender de que tan bien se hayan definido las características del set de datos y la cantidad y calidad de los datos de aceleración capturados.

The screenshot shows a Jupyter Notebook window titled 'Proyecto Titulo' with a last checkpoint from yesterday at 13:03. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The main area contains two code cells:

```
In [55]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#matplotlib inline
from sklearn.model_selection import train_test_split

dataset = pd.read_csv("log.csv")
dataset = dataset[['RMS-x', 'RMS-y', 'RMS-z', 'Mean-x', 'Mean-y', 'Me
dataset.shape

Out[55]: (1082, 22)
```

```
In [56]: d = dataset

X_train, X_test, y_train, y_test = train_test_split(dataset.drop('Act.
y_train.head()
```

```
Out[56]: 471      Subir
176      Caminar
920      Resbalo
213      Quieto
934      Resbalo
Name: Activity, dtype: object
```

Figura 16 - Jupyter Notebook

## 5. Reconocimiento de la actividad

Luego de realizado el procesamiento de datos y la generación de los modelos, se evalúa que tan efectiva fue la clasificación de acuerdo con la precisión obtenida a la hora de predecir los movimientos. Esta información se obtiene a través de la denominada matriz de confusión que nos indica el comportamiento de la predicción de nuestro modelo.

De acuerdo con estos resultados se determina si el modelo es válido o requiere mayor entrenamiento.

### 5.1 Matriz de confusión

La matriz de confusión es una tabla que permite observar el rendimiento de un algoritmo, en este caso particularmente de los algoritmos de clasificación del modelo entrenado con el set de datos especificado. Su nombre se refiere al hecho de que se puede observar fácilmente si el sistema confunde las clases (al etiquetarlas), ver Figura 17.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 17 - Matriz de Confusión

Con esta tabla es posible obtener la información necesaria para determinar valores como *precision* y *recall*, que definen el desempeño del modelo ver.

$$precision = \frac{VP}{VP + FP}$$

$$recall = \frac{VP}{VP + FN}$$



### 5.1.1 Árbol de decisión

De acuerdo con la matriz de confusión obtenida mediante el modelo aplicado se obtiene la siguiente información:

Tabla 6 - *Precision* - Árbol de Decisión

ACTIVIDAD	PRECISION
QUIETO	1.00
CAMINAR	0.81
SUBIR ESCALERA	0.59
BAJAR ESCALERA	0.61
TROPEZAR	0.50
RESBALAR	0.39

Como se puede observar, ‘tropezar’ y ‘resbalar’ tienen una precisión más baja que el resto incluso, en el caso de tropezar, rozando el 50%, lo que nos indica que no es capaz de determinar con claridad la clase en cuestión con el modelo entrenado, esto puede deberse al tamaño del dataset de entrenamiento y de pruebas, o de las características escogidas no son las óptimas para este modelo.

### 5.1.2 K-Nearest Neighbor

De acuerdo con la matriz de confusión obtenida mediante el modelo aplicado se obtiene la siguiente información:

Tabla 7 - *Precision* KNN

ACTIVIDAD	PRECISION
QUIETO	1.00
CAMINAR	0.72
SUBIR ESCALERA	0.73
BAJAR ESCALERA	0.66
TROPEZAR	0.78
RESBALAR	0.40

Este modelo al igual que el modelo de ANN tiene un buen desempeño a excepción de la etiqueta resbalar, la cual solo obtuvo un 40% de *precision*, lo que indica que el modelo tiene dificultades clasificando los datos para esa etiqueta.

### 5.1.3 Support Vector Machine

De acuerdo con la matriz de confusión obtenida mediante el modelo aplicado se obtiene la siguiente información:

Tabla 8 - *Precision SVM*

ACTIVIDAD	PRECISION
QUIETO	1.00
CAMINAR	0.71
SUBIR ESCALERA	0.64
BAJAR ESCALERA	0.62
TROPEZAR	0.50
RESBALAR	0.50

Este modelo genera predicciones con *precision* sobre el 50% para todas sus clases, sin embargo, las más bajas corresponden a tropezar y resbalar, al igual que en el resto de los modelos.

### 5.1.4 Artificial Neural Network

De acuerdo con la matriz de confusión obtenida mediante el modelo aplicado se obtiene la siguiente información:

Tabla 9 - *Precision ANN*

ACTIVIDAD	PRECISION
QUIETO	0.98
CAMINAR	0.81
SUBIR ESCALERA	0.77
BAJAR ESCALERA	0.72
TROPEZAR	0.62
RESBALAR	0.47

Este modelo presenta un desempeño aceptable, salvo por la etiqueta resbalar, la cual tiene bajo el 50% de *precision*. Este modelo junto al modelo de KNN, han resultado ser los que mejor desempeño han tenido.

## 5.2 Resultados

Los resultados obtenidos por los distintos algoritmos son similares, sin embargo destacan los modelos de la red neuronal artificial y K-nearest neighbors, donde el primero entrega valores sobre 70% para: quieto, caminar, subir y bajar escaleras, sin embargo no muy buenos resultados respecto a tropezar y resbalar con un 62% y 47% respectivamente; y el segundo, también un sobre un 70% para: quieto, caminar, subir y bajar escaleras y tropezar y resbalar con un 78% y 40% respectivamente, Tabla 10.

El relativo bajo desempeño de los modelos para detectar: tropezar y resbalar, está dado principalmente por la cantidad de muestras obtenidas para estos movimientos, que fue menor que las demás, debido a que resulta más complejo capturar estos datos, y además la posibilidad de que puedan existir pequeñas discordancias a la hora de sincronizar la obtención de los datos con las grabaciones en video de la simulación.

Como desafío, o trabajo futuro, generar dataset con mayor cantidad de datos y más uniforme, incluyendo más escenarios e individuos, para que de esta manera los modelos creados puedan tener un desempeño mejor, no obstante, para las demás actividades el resultado obtenido es aceptable. Lo importante es destacar la metodología utilizada, ya que es fácilmente replicable y de gran valor a la hora de desarrollar diversas soluciones en diversas áreas, no sólo en la construcción.

Tabla 10 - Resultados

Actividad	Algoritmo de Clasificación							
	KNN		DT		SVM		ANN	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
<b>Quieto</b>	1.00	0.98	1.00	0.98	1.00	1.00	0.98	1.00
<b>Caminar</b>	0.72	0.97	0.81	0.78	0.71	0.84	0.81	0.81
<b>Subir Escaleras</b>	0.73	0.75	0.59	0.60	0.64	0.58	0.77	0.75
<b>Bajar Escaleras</b>	0.66	0.76	0.61	0.68	0.62	0.76	0.72	0.82
<b>Tropezar</b>	0.78	0.29	0.50	0.33	0.50	0.25	0.62	0.33
<b>Resbalar</b>	0.40	0.27	0.39	0.57	0.50	0.47	0.47	0.60

## 6. Prototipo de software

### 6.1 Objetivo del software

La aplicación Web tiene como objetivo presentar los modelos obtenidos en el trabajo de forma sencilla e intuitiva y que permita a los usuarios del sistema, predecir o clasificar movimientos con *dataset* propios, es decir, que se puedan reconocer patrones en datos capturados por ellos y determinar el movimiento que se está realizando. Debido a que consiste en un prototipo y a que el modelo no ha sido entrenado con la cantidad y variedad de datos necesaria para que entregue resultados con mayor precisión, los resultados no son los mejores, pero claramente sirve como una prueba de concepto de lo que se ha querido explicar a lo largo del trabajo, y por lo mismo es que se habla de prototipo de software.

### 6.2 Descripción del Software

El software, corresponde a una aplicación web, la cual está desarrollada con el *framework* Django, que utiliza el lenguaje de programación Python, en conjunto con la librería scikit-learn, utilizando como idea base lo consultado en la literatura [16].

Esta aplicación se encuentra montada en un servidor propio (Raspberry Pi 3), consiste en una página principal en la cual se puede subir un archivo de texto o archivo de valores separados por coma, el cual debe contener las mediciones de la aceleración en sus 3 ejes, y se debe seleccionar que modelo se desea utilizar, como resultado el software entregará la información detallada de los movimientos detectados por el modelo seleccionado.

#### 6.2.1 Requerimientos

Debido a que la aplicación web corresponde a un prototipo, por lo que los requerimientos son pocos y generales.

##### Requerimiento Funcionales

- El sistema permitirá a un usuario de tipo “administrador” ingresar los diferentes modelos que se utilizarán en la aplicación.
- Se podrá subir un set de datos propios (extensión csv), sin etiquetar, para ser evaluado por los modelos.
- El sistema debe mostrar una tabla con el set de datos y con su respectiva predicción de acuerdo al modelo seleccionado.
- El sistema debe permitir al usuario escoger el modelo que se desea aplicar al set de datos.

## Requerimientos No funcionales

- La aplicación debe desarrollarse utilizando el *framework* Django.
- Un usuario experimentado debe ser capaz de utilizar la aplicación luego de una capacitación de no más de 10 minutos.

### 6.2.2 Entorno de desarrollo

Para el desarrollo de la aplicación se utilizaron las siguientes herramientas.

- Equipo: MacBook Air
- Sistema operativo: MacOS Mojave
- Framework web Django, versión 2.4
- Editor de código Visual Studio Code
- Lenguaje de programación Python, versión 3.7
- Librerías de Python:
  - Scikit Learn versión 0.21.3
  - Pandas versión 0.23.4
  - Numpy versión 1.15.4
- Controlador de versiones Git - Github
- Servidor: Raspberry Pi3 - Raspian

### 6.2.3 Caso de uso

Nombre	Reconocer la actividad
Descripción	Permite a un usuario reconocer (predecir) la actividad (o actividades) asociada a un cierto set de datos.
Actores	Usuario

Escenario	Usuario	Sistema
	1. El usuario sube el archivo con su set de datos propio, y selecciona el modelo con el que evaluará sus datos.	
		2. La aplicación recibe los datos, carga el modelo (que fue seleccionado por el usuario) y lo aplica sobre el set de datos. Luego, entrega una

		tabla donde se muestra toda la información y se adjunta la predicción realizada bajo la etiqueta de 'Activity'.
	3. Si el usuario lo desea, puede repetir el proceso.	

### 6.2.4 Modelo arquitectónico

La aplicación web implementa esta modelada de acuerdo con la arquitectura que se muestra en la Figura 18.

El proceso de aprendizaje corresponde al proceso en donde se entrenan los diferentes modelos, y este proceso ocurre de manera aislada a la aplicación, es un paso previo y ocurre *offline*, este proceso alimenta la capa de modelos, la cual recibe los modelos entrenados que luego son solicitados por la aplicación de acuerdo a las solicitudes realizadas por el usuario. Los resultados generados por la aplicación se entregan al usuario de forma visual mediante el navegador web, en la capa vista.

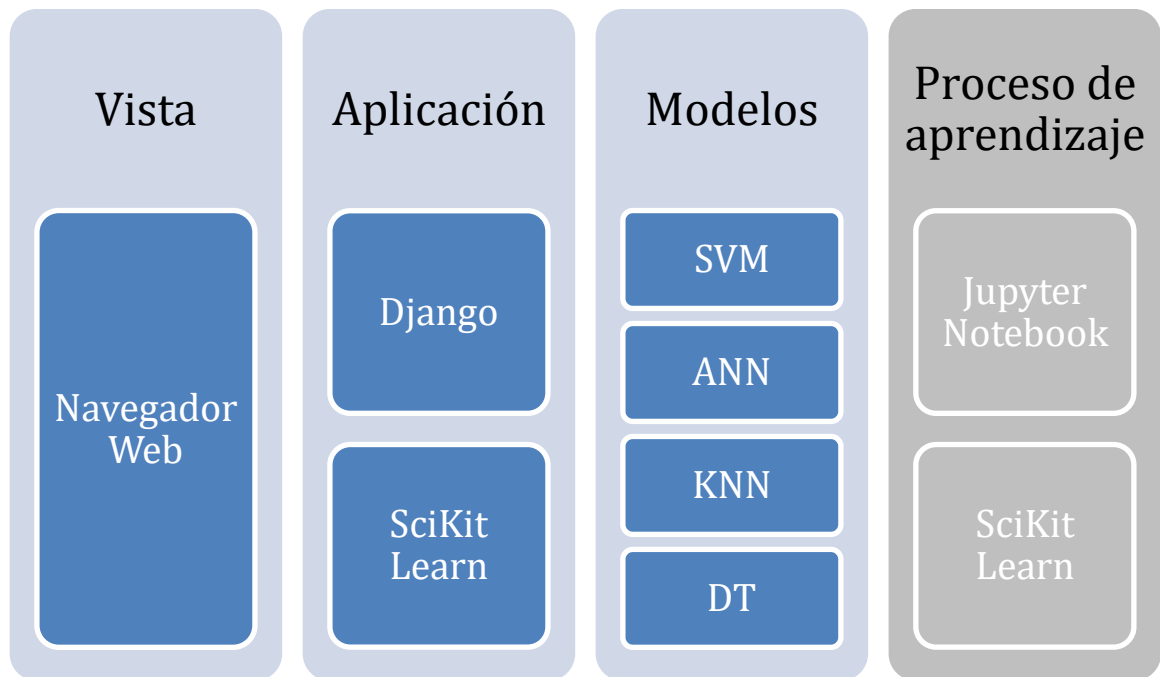


Figura 18 - Arquitectura de la aplicación

## 6.2.5 Funcionamiento

El Software se divide principalmente en 2 vistas.

La primera como se muestra en la Figura 19, corresponde a un formulario en donde se adjunta el dataset (archivo .csv) que se desea evaluar y un elemento *dropdown*, ver Figura 20, en el cual se selecciona el modelo que se desea aplicar (DT, KNN,SVM, ANN) al conjunto de datos nuevos entregados por el usuario, siendo estos modelos los obtenidos mediante el entrenamiento con los datos obtenidos de los experimentos realizados.

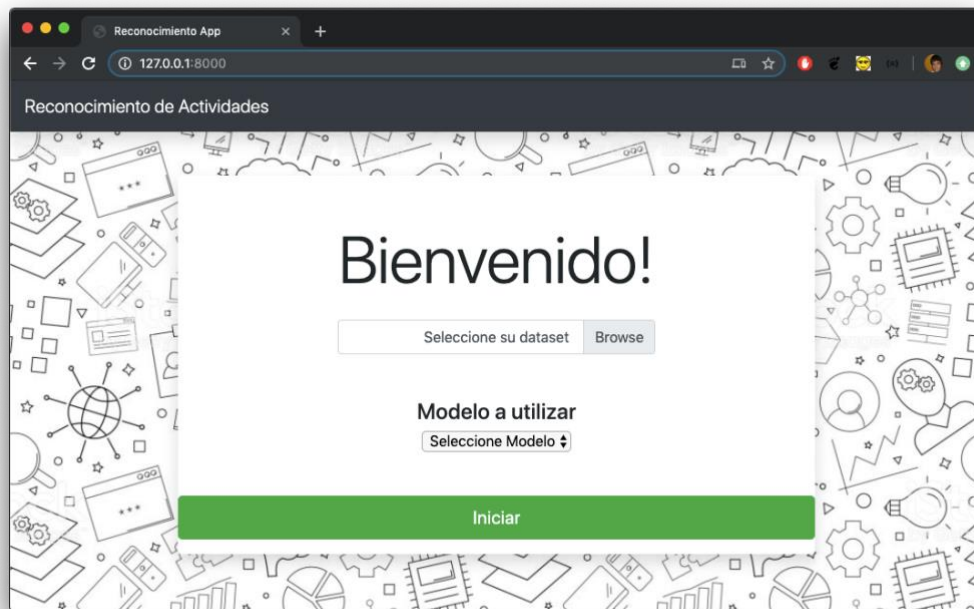


Figura 19 - Aplicación Web - Principal

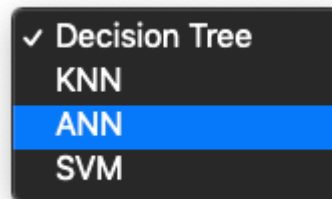


Figura 20 - Aplicación Web (*dropdown* para seleccionar modelo)

La segunda vista de la aplicación entrega todos los datos asociados al movimiento, junto con la respectiva predicción que realiza el modelo de acuerdo con los datos que recibe, y esta predicción se incorpora al dataset, ver Figura 21.

	RMS-x	RMS-y	RMS-z	Mean-x	Mean-y	Mean-z	Max-x	Max-y	Max-z	Min-x	Min-y	Min-z	IQR-x	IQR-y	IQR-z	Var-x	Var-y	Var-z	Std-x	Std-y	Std-z	Activity
0	1.79	10.39	1.92	-0.19	-10.35	0.64	1.59	-9.64	3.55	-3.22	-13.18	-1.55	3.28	0.34	3.75	3.49	1.03	3.60	1.87	1.02	1.90	Caminar
1	1.55	8.77	0.85	1.54	-8.76	-0.80	1.95	-8.29	-0.42	1.32	-9.29	-1.20	0.20	0.62	0.47	0.04	0.12	0.09	0.19	0.35	0.39	Caminar
2	2.71	9.36	2.37	2.71	-9.36	-2.30	2.88	-9.15	-1.34	2.19	-9.99	-3.09	0.13	0.21	0.95	0.05	0.08	0.38	0.22	0.28	0.31	Caminar
3	1.58	10.68	1.85	1.46	-10.68	0.24	2.53	-10.23	4.89	0.48	-11.22	-1.79	0.90	0.65	1.59	0.40	0.12	3.72	0.63	0.35	1.93	Caminar
4	1.48	13.00	5.26	1.00	-12.96	4.57	2.47	-11.88	8.79	-0.62	-14.36	0.60	2.34	1.99	4.89	1.32	1.05	7.53	1.15	1.02	2.74	Caminar
5	1.52	13.14	5.15	-0.90	-13.13	-4.94	0.91	-12.33	-1.42	-2.93	-14.05	-6.21	1.94	0.79	1.48	1.66	0.33	2.26	1.29	0.58	1.50	Subir
6	1.66	9.79	3.29	-0.41	-9.75	-3.25	1.56	-8.97	-2.52	-2.86	-11.74	-3.95	3.11	1.00	1.06	2.88	0.94	0.31	1.70	0.97	0.56	Subir
7	1.18	8.73	3.20	1.10	-8.72	-3.19	2.06	-8.10	-2.83	0.71	-9.20	-3.42	0.67	0.88	0.31	0.19	0.20	0.05	0.44	0.44	0.22	Caminar
8	4.82	7.42	2.18	4.75	-7.41	-1.74	5.37	-6.82	0.45	2.81	-8.07	-3.22	1.08	0.88	2.58	0.82	0.22	1.90	0.91	0.47	0.38	Caminar
9	3.66	10.15	4.84	2.02	-9.97	4.36	5.27	-7.63	7.52	-4.58	-13.85	0.99	4.14	2.72	3.77	10.33	4.02	4.90	3.21	2.00	0.21	Caminar
10	4.29	9.48	2.01	-0.02	-9.13	-0.39	5.43	-6.44	3.72	-5.78	-14.16	-2.33	9.17	3.95	3.19	20.41	7.19	4.31	4.52	2.68	1.08	Tropiezo
11	3.86	11.60	4.07	3.77	-10.42	2.16	5.79	-3.48	7.55	2.87	-17.68	-2.34	1.45	9.42	6.45	0.82	29.00	13.25	0.91	5.39	3.64	Caminar
12	1.57	11.27	4.83	0.95	-10.66	-4.45	2.78	-3.99	-0.80	-0.55	-15.17	-6.72	2.61	6.02	2.79	1.75	14.91	3.87	1.32	3.86	1.97	Caminar
13	0.69	10.00	1.37	0.36	-9.94	1.01	1.19	-8.38	2.39	-0.47	-11.63	-0.32	1.14	2.21	1.89	0.38	1.40	0.98	0.62	1.19	0.89	Caminar
14	2.69	8.44	2.08	2.56	-8.42	-1.87	3.63	-7.77	-0.47	1.24	-9.05	-3.27	1.37	0.76	1.68	0.72	0.22	0.96	0.85	0.47	0.99	Caminar
15	4.27	8.82	3.26	4.24	-8.79	-3.22	4.80	-8.01	-2.02	3.23	-10.30	-3.72	0.86	1.42	0.80	0.30	0.74	0.34	0.55	0.86	0.58	Caminar
16	1.35	11.00	1.10	0.99	-10.98	-1.07	2.74	-10.22	-0.65	0.06	-12.29	-1.62	1.65	1.09	0.29	0.94	0.51	0.08	0.97	0.71	0.28	Subir

Figura 21 - Aplicación Web - Resultados

La aplicación permite además mediante herramientas propias del *framework* Django, agregar y eliminar los modelos que se desea aplicar, simplemente se crean nuevos modelos en el panel de administración como se observa en la Figura 22. Al crear un modelo nuevo se debe indicar un nombre y además subir archivo del modelo, para que estos aparezcan luego en el *dropdown* de la primera vista, para ser seleccionados por el usuario.

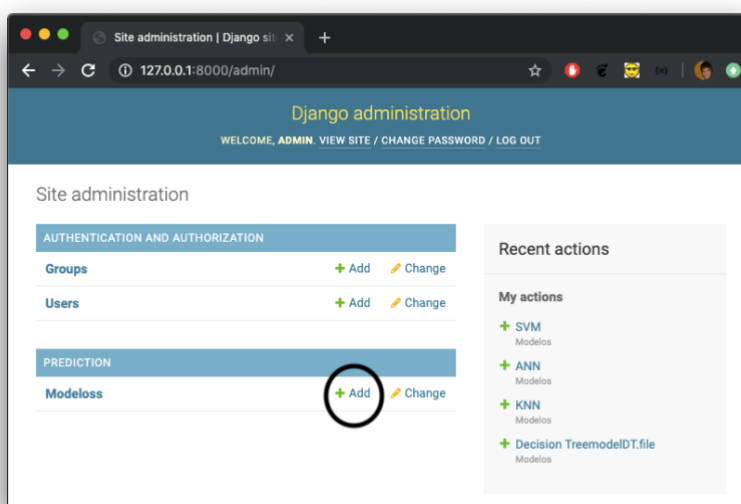


Figura 22 - Aplicación Web – Administración



## 7. Conclusiones

Las herramientas que entrega el Machine Learning permiten realizar un sinnúmero de aplicaciones que son de una gran utilidad para las personas, en todos los ámbitos que se pueda imaginar, ya que actualmente nos encontramos en una era digital y de la información, donde existen grandes volúmenes de datos al alcance de la mano, datos que se pueden analizar para entregar soluciones a problemas tanto cotidianos como complejos. Es por esto que es importante comprender las bases de lo que estas tecnologías representan (Machine Learning) y lograr aplicarlas de forma adecuada de acuerdo al problema que se enfrente, ya que existen muchas técnicas y puede resultar confuso. Sin embargo, es necesario comenzar a familiarizarse con ellas e implementarlas poco a poco en la industria. Si bien es una tecnología que lleva varios años vigentes, ha sido la última década en donde más se ha hablado de ellas, y en nuestro país el uso y aplicación de estas tecnologías no está del todo explotado, especialmente en áreas como la educación, administración, construcción, salud entre otras.

En este contexto se lleva a cabo este Proyecto de Título, en el cual se desarrolló una metodología de captura y análisis de datos, utilizando un teléfono celular, mediante un proceso en el cual se simulaban actividades cotidianas como caminar y subir escaleras y movimientos relacionados con la accidentabilidad en obra, como lo son tropezar y resbalar. Para esto se realizaron varias simulaciones las cuales permitieron capturar datos asociados a estos movimientos, con los que a su vez permitieron entrenar 4 modelos de predicción de movimientos (DT, KNN, SVM y ANN). Es importante destacar que estos modelos deben seguir siendo entrenados para entregar resultados más precisos y también que es factible aumentar las actividades a determinar, pudiendo incluso definir actividades más particulares de la construcción como: pintar, martillar, cavar, aserruchar, etc. Para esto se revisó la literatura y se seleccionaron aquellas técnicas más comúnmente utilizadas y las que mejor se comportan con el reconocimiento de actividades humanas, de la misma forma se realizó la selección de variables a utilizar.

Se implementó además un algoritmo que permite generar un set de datos con las variables que se desea evaluar, el cual recibe un archivo de texto (los datos crudos del acelerómetro) y devuelve un archivo .csv con todas las variables determinadas y la respectiva etiqueta del movimiento/actividad. Lo que permite luego tener un set de datos para entrenar los modelos.

Finalmente se desarrolló un prototipo de aplicación web la cual contiene los modelos que se entrenaron y permite a un usuario subir su propio set de datos de acelerómetro, escoger el modelo el cual desea utilizar para reconocer la actividad y la aplicación etiqueta estos datos con la actividad que este predice de acuerdo con el modelo escogido, lo que en definitiva muestra de forma práctica la finalidad de este Proyecto,

la cual corresponde a reconocer el movimiento y la actividad de diversos individuos, enfocado principalmente en el área de la construcción (accidentabilidad).

La aplicación de las nuevas tecnologías es una tarea de todos, y es necesario comenzar dando pequeños pasos, como lo es este trabajo, el cual pretende entregar una pequeña introducción al Machine Learning y además presentar una manera en la que se pueden utilizar sus técnicas para abordar distintos problemas y entregar soluciones nuevas a la industria y a las personas en general.

## 8. Bibliografía

- [1] Y. Tsutsumi, “EVALUACIÓN DE UNA NUEVA METODOLOGÍA PARA LA MEDICIÓN Y EVALUACIÓN DE LA PRODUCTIVIDAD DE LA MANO DE OBRA,” 2017.
- [2] R. W. DeVaul and S. Dunn, “Real-time motion classification for wearable computing applications,” 2001.
- [3] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” *Aaai*, vol. 5, pp. 1541–1546, 2005.
- [4] T. Cheng, J. Teizer, G. C. Migliaccio, and U. C. Gatti, “Automated task-level activity analysis through fusion of real time location sensors and worker’s thoracic posture data,” *Autom. Constr.*, vol. 29, pp. 24–39, 2013.
- [5] T. K. Lim, S. M. Park, H. C. Lee, and D. E. Lee, “Artificial neural network–based slip-trip classifier using smart sensor for construction workplace,” *J. Constr. Eng. Manag.*, vol. 142, no. 2, p. 04015065, 2015.
- [6] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. 2017.
- [7] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine,” in *International workshop on ambient assisted living*, 2012, pp. 216–223.
- [8] R. Akhavian and A. H. Behzadan, “Smartphone-based construction workers’ activity recognition and classification,” *Autom. Constr.*, vol. 71, pp. 198–209, 2016.
- [9] M. Zhang, S. Chen, X. Zhao, and Z. Yang, “Research on construction workers’ activity recognition based on smartphone,” *Sensors*, vol. 18, no. 8, p. 2667, 2018.
- [10] R. Akhavian and A. H. Behzadan, “Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers,” *Adv. Eng. Informatics*, vol. 29, no. 4, pp. 867–877, 2015.
- [11] P. Siirtola and J. Roning, “Ready-to-use activity recognition for smartphones,” *Proc. 2013 IEEE Symp. Comput. Intell. Data Mining, CIDM 2013 - 2013 IEEE Symp. Ser. Comput. Intell. SSCI 2013*, no. February 2016, pp. 59–64, 2013.

- [12] M. Ouchi, K.; Doi, “Indoor-outdoor activity recognition by a smartphone,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012.
- [13] O. D. . M. A. L. Lara, “A mobile platform for real-time human activity recognition,” in *2012 IEEE consumer communications and networking conference (CCNC)*, 2012.
- [14] I. Lisowska, A; Wheeler, G.; Ceballos, V; Poole, “An Evaluation of Supervised, Novelty-Based and Hybrid Approaches to Fall Detection Using Silmee Accelerometer Data,” in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, 2015, pp. 10–16.
- [15] A. Pai, O. Nachum, and M. Kanter, “Activity Classification using Smartphone Accelerometer Data,” pp. 1–10, 2013.
- [16] A. Isoni, *Machine Learning for the Web*. 2016.