

**UNIVERSIDAD CATÓLICA BOLIVIANA “SAN PABLO”**

**UNIDAD ACADÉMICA REGIONAL COCHABAMBA**

**Departamento de Ciencias Exactas e Ingeniería de Sistemas**



UNIVERSIDAD  
**CATÓLICA**  
BOLIVIANA  
COCHABAMBA

***REPORTE TÉCNICO***

**Bustamante Quiroz Isabel**

**Lozada Peralta Jose Carlos**

**Montaño Torrico Jorge Adrian**

**Mounzon Baspineiro Adriel David**

Cochabamba-Bolivia

Mayo-2024

## 1. Preprocesamiento de datos

### 1.1. Enlistar y explicar brevemente las técnicas de preprocesamiento e ingeniería de features que se aplicaron a las diferentes soluciones.

Todas las técnicas y funciones de preprocesamiento pueden ser encontradas en el archivo “*preprocessing\_module.py*” encontrado en el repositorio de GitHub: “<https://github.com/MrJamt/proyecto-final-sis-int>”.

#### 1.1.1. Cargar datos:

**Función:** cargar\_datos(ruta\_csv)

Haciendo uso de esta funcion es que se consigue cargar los datos desde el archivo CSV enviado como parámetro, utilizando la librería pandas.

#### 1.1.2. Eliminación de Columnas:

**Función:** eliminar\_columnas(data, columnas)

Esta función fue diseñada exclusivamente para eliminar columnas innecesarias. En este caso, se elimina la columna “id”

#### 1.1.3. Manejo de Valores Faltantes:

**Función:** manejar\_valores\_faltantes(data, columna, valor)

Utilizada para llenar los valores faltantes en la columna 'Arrival Delay in Minutes' la cual es llenada con valores en 0.

#### 1.1.4. Separación de Características Numéricas y Categóricas:

**Función:** separar\_caracteristicas(data, caracteristicas\_numericas, caracteristicas\_categoricas)

Utilizado para separar las características en dos conjuntos:

- Características numéricas
- Características categóricas

### Ingeniería de Features

#### 1.1.5. Normalización de Datos Numéricos:

**Función:** normalizar\_datos(numeric\_data)

Utilizado para normalizar las características numéricas haciendo uso de la librería “StandardScaler”

#### 1.1.6. Codificación de Datos Categóricos:

**Función:** `codificar_datos(categorical_data)`

Encargado de transformar las características categóricas en valores numéricos utilizando “LabelEncoder”

#### 1.1.7. Codificación de la Variable Objetivo:

**Función:** `codificar_objetivo(data, columna_objetivo)`

Convierte la variable satisfacción, que es el objetivo, en valores binarios. Donde:

- satisfied = 1
- neutral/unsatisfied = 0.

## 2. Arquitecturas de la red neuronal, para cada solución:

### 2.1. Nombres de la arquitectura

N	Nombre	Nombre de la Arquitectura
1	A	3 capas
2	B	4 Capas y mayor complejidad de la red
3	C	5 Capas y menor complejidad en la red
4	D	Menor complejidad
5	E	Modelo con Capas más Profundas y regularización hasta la 3era capa

### 2.2. Técnicas de preprocesamiento e ingeniería de features aplicadas a esta solución.

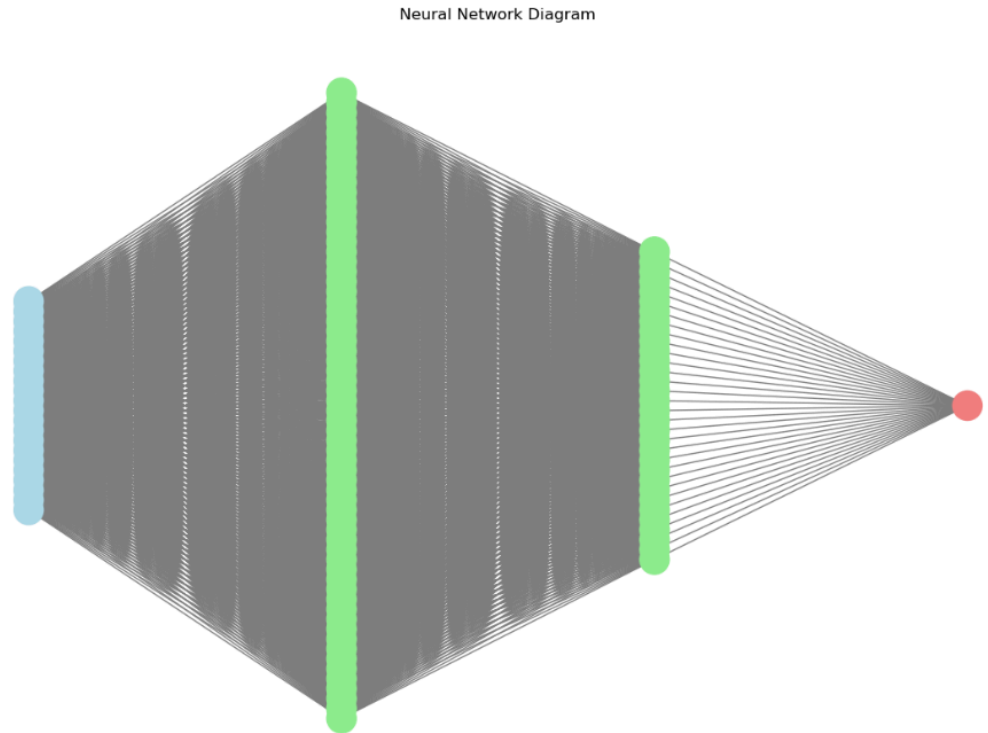
Se utilizaron las mismas técnicas de preprocesamiento e ingeniería de features redactadas en el punto 1 del presente reporte, para las cuatro arquitecturas, puesto que todas utilizan los mismos DataSet, tanto “train.csv” como “test.csv”, los cuales pueden ser encontrados en el mismo repositorio.

### 2.3. Gráfica de ilustración

Las gráficas se realizaron con la librería networkx y matplotlib.pyplot dentro de un archivo “*graphic\_module.py*” que se encuentra en el repositorio de

GitHub anteriormente mencionado:  
“<https://github.com/MrJamt/proyecto-final-sis-int>”.

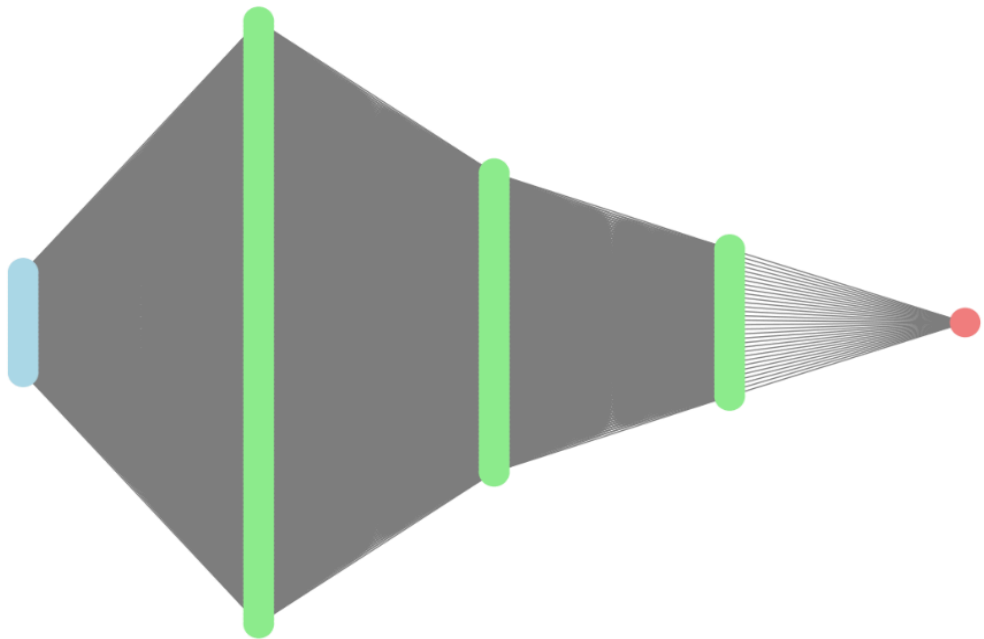
A)



**Gráfica 1:** Red neuronal A: 3 capas  
**Fuente:** Elaboración Propia

B)

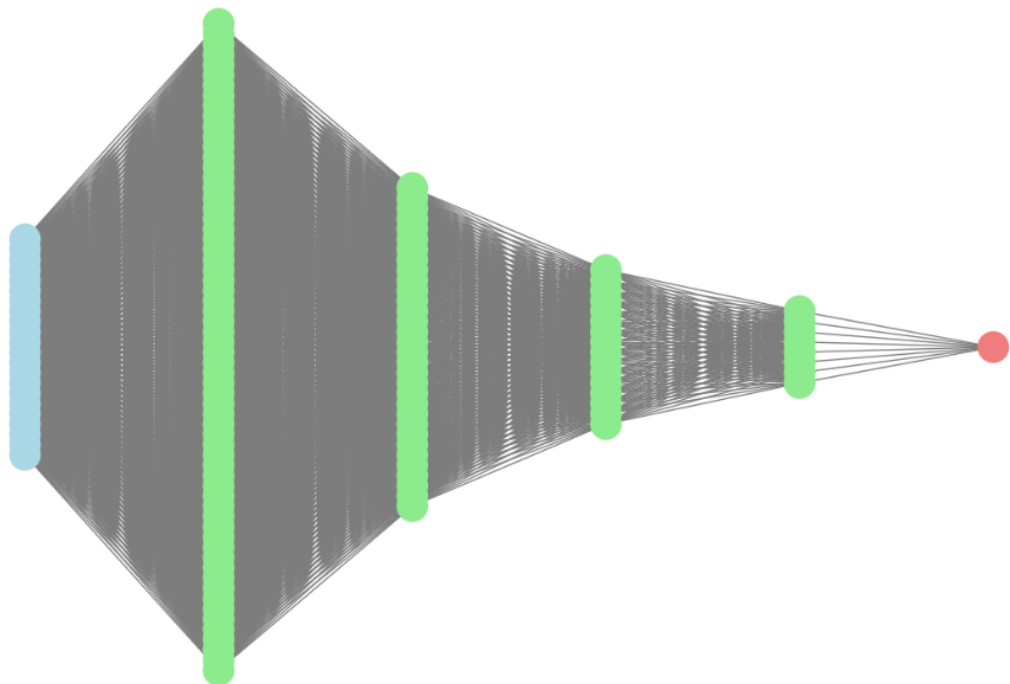
Neural Network Diagram



**Gráfica 2:** Red neuronal B: 4 Capas y mayor complejidad de la red  
**Fuente:** Elaboración Propia

C)

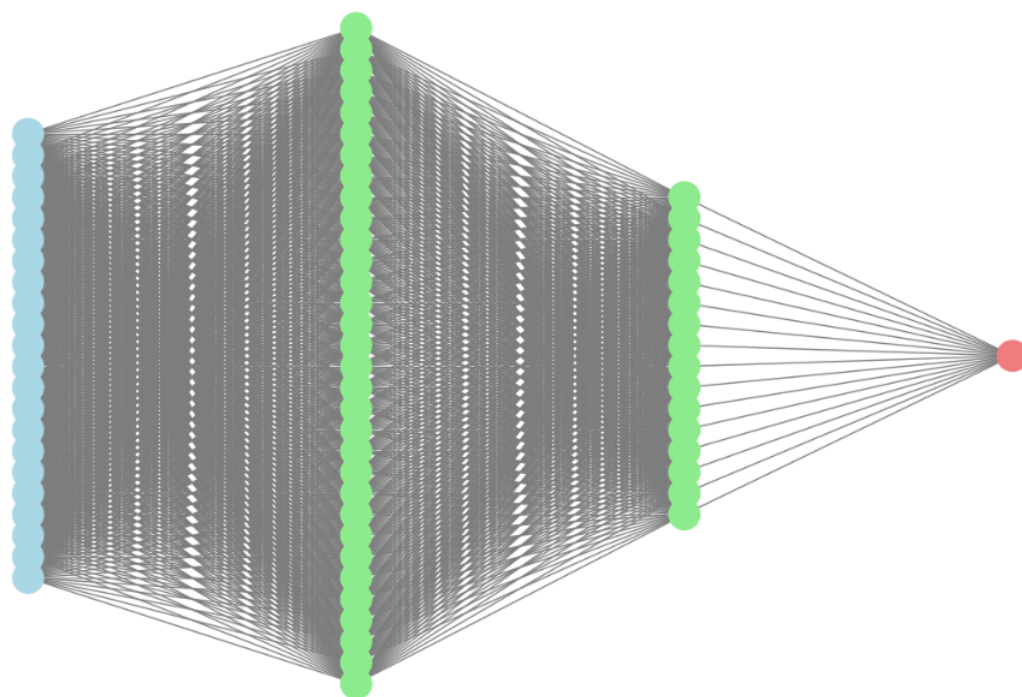
Neural Network Diagram



**Gráfica 3:** Red neuronal C: 5 Capas y menor complejidad en la red  
**Fuente:** Elaboración Propia

D)

Neural Network Diagram

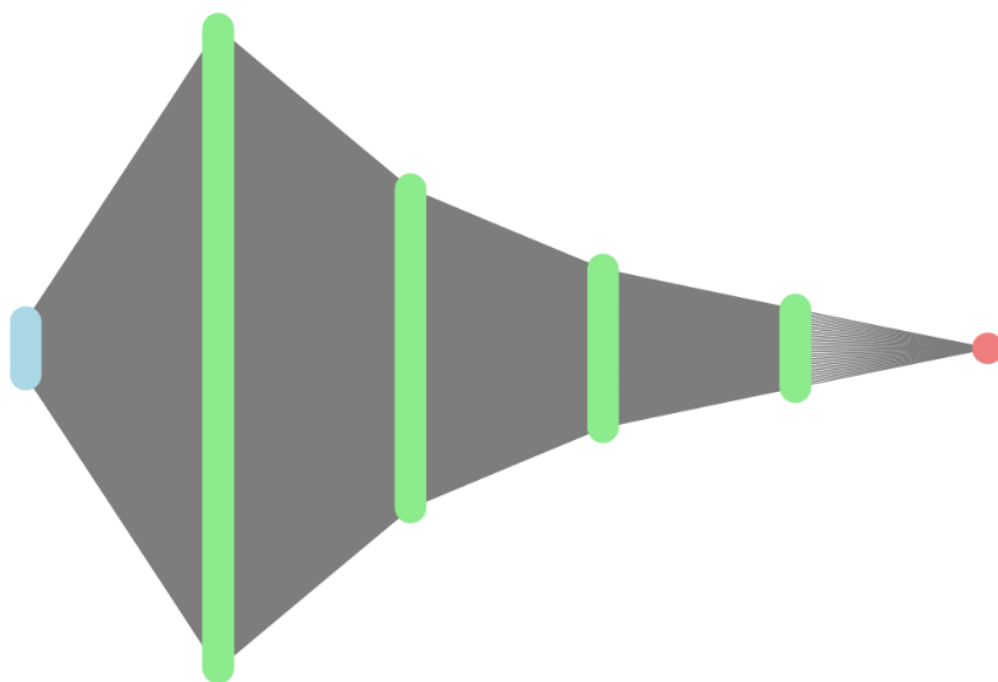


**Gráfica 4:** Red neuronal D: Menor complejidad

**Fuente:** Elaboración Propia

**E)**

Neural Network Diagram



**Gráfica 5:** Red neuronal E: Modelo con Capas más Profundas y regularización hasta la 3era capa

**Fuente:** Elaboración Propia

**2.4. Descripción textual de la red neuronal (número de capas, tipo de capas, número de neuronas, funciones de activación)**

**A)**

<b>Número Capas</b>	<b>Tipo</b>	<b>Número de Neuronas</b>	<b>Función de Activación</b>
<b>Capa 1</b>	Capa de Entrada	22	-
<b>Capa 2</b>	Capa Oculta 1	64	ReLU (Rectified Linear Unit)
<b>Capa 3</b>	Capa Oculta 1	32	ReLU (Rectified Linear Unit)
<b>Capa 4</b>	Capa de Salida	1	Sigmoid

**B)**

<b>Número Capas</b>	<b>Tipo</b>	<b>Número de Neuronas</b>	<b>Función de Activación</b>
<b>Capa 1</b>	Capa de Entrada	22	-
<b>Capa 2</b>	Capa Oculta 1	128	ReLU (Rectified Linear Unit)
<b>Capa 3</b>	Capa Oculta 2	64	ReLU (Rectified Linear Unit)
<b>Capa 4</b>	Capa Oculta 3	32	ReLU (Rectified Linear Unit)
<b>Capa 5</b>	Capa de Salida	1	Sigmoid

**C)**

<b>Número Capas</b>	<b>Tipo</b>	<b>Número de Neuronas</b>	<b>Función de Activación</b>
<b>Capa 1</b>	Capa de Entrada	22	-
<b>Capa 2</b>	Capa Oculta 1	64	ReLU (Rectified Linear Unit)

<b>Capa 3</b>	Capa Oculta 2	32	ReLU (Rectified Linear Unit)
<b>Capa 4</b>	Capa Oculta 3	16	ReLU (Rectified Linear Unit)
<b>Capa 5</b>	Capa Oculta 4	8	ReLU (Rectified Linear Unit)
<b>Capa 6</b>	Capa de Salida	1	Sigmoid

**D)**

<b>Número Capas</b>	<b>Tipo</b>	<b>Número de Neuronas</b>	<b>Función de Activación</b>
<b>Capa 1</b>	Capa de Entrada	22	-
<b>Capa 2</b>	Capa Oculta 1	32	ReLU (Rectified Linear Unit)
<b>Capa 3</b>	Capa Oculta 2	16	ReLU (Rectified Linear Unit)
<b>Capa 4</b>	Capa de Salida	1	Sigmoid

**E)**

<b>Número Capas</b>	<b>Tipo</b>	<b>Número de Neuronas</b>	<b>Función de Activación</b>
<b>Capa 1</b>	Capa de Entrada	22	-
<b>Capa 2</b>	Capa Oculta 1	256	ReLU (Rectified Linear Unit)
<b>Capa 3</b>	Capa Oculta 2	128	ReLU (Rectified Linear Unit)
<b>Capa 4</b>	Capa Oculta 3	64	ReLU (Rectified Linear Unit)
<b>Capa 5</b>	Capa Oculta 4	32	ReLU (Rectified Linear Unit)
<b>Capa 6</b>	Capa de Salida	1	Sigmoid



### Aclaraciones de cada capa y los parámetros utilizados en cada modelo de Red Neuronal:

- La **capa de entrada** que recibe datos con la forma de (X\_train.shape[1]).
- El parámetro de **“Dropout”** previene el sobreajuste, de manera que el 30% de las neuronas se apagan aleatoriamente durante el entrenamiento.
- La **capa de salida** genera un valor único de probabilidad, para realizar la clasificación binaria entre “Satisfecho” o “Neutral/Insatisfecho”
- La **función de pérdida** “binary\_crossentropy” (usado en problemas de clasificación binaria).
- La **métrica** utilizada es accuracy (precisión).
- La **división de validación** es de 20%, por lo tanto el otro 80% está destinado a **entrenamiento**.
- El **verbose** es 1, es el que muestra la barra de progreso durante el entrenamiento.
- Las **predicciones** generan probabilidades en el conjunto de prueba y se convierten a etiquetas binarias utilizando un **umbral** de **0.5**.
- La **matriz de confusión** es la encargada de evaluar el rendimiento del modelo.
- La **especificidad** es la proporción de verdaderos negativos entre todos los negativos predichos.

### 3. Tabla comparativa de entrenamiento

Arquitectura	Nombre de la arquitectura	Tasa de aprendizaje	Optimizador	Tamaño de lote	Número de épocas	Tiempo aproximado de entrenamiento (seg)	Costo final después del entrenamiento
A	3 capas	0.001	Adam	32	20	72	0.105
B	4 Capas y mayor complejidad de la red	0.001	Adam	32	20	122	0.0949
C	5 Capas y menor complejidad en la red	0.001	Adam	32	20	113	0.1245
D	Menor complejidad	0.001	Adam	32	20	85	0.1301
E	Modelo con Capas más Profundas y regularización hasta la 3era capa	0.001	Adam	32	20	160	0.0886

### 4. Tabla comparativa de evaluación

Arquitectura	Nombre de la arquitectura	Clase	Precisión	Recall	Especificidad	F1-Score
A	3 capas	neutral/unsatisfied	0.95	0.98	0.98	0.97
		satisfied	0.97	0.94		0.95
B	4 Capas y mayor complejidad de la red	neutral/unsatisfied	0.95	0.99	0.99	0.97
		satisfied	0.98	0.93		0.95
C	5 Capas y menor complejidad en la red	neutral/unsatisfied	0.95	0.98	0.98	0.97
		satisfied	0.97	0.94		0.96
D	Menor complejidad	neutral/unsatisfied	0.94	0.98	0.98	0.96
		satisfied	0.97	0.92		0.95
E	Modelo con Capas más Profundas y regularización hasta la 3era capa	neutral/unsatisfied	0.96	0.98	0.98	0.97
		satisfied	0.97	0.95		0.96