

1. Preprocesamiento de Datos

Se detallarán las técnicas de preprocesamiento e ingeniería de características que hemos aplicado en el proyecto proporcionado:

1.1 Técnicas de Preprocesamiento e Ingeniería de Features:

- **Conversión de la variable objetivo a binaria:** La columna 'satisfaction' se mapeó a valores binarios, donde 'satisfied' se convirtió en 1 y 'neutral or dissatisfied' en 0. Esto se realizó para ambos conjuntos de datos.
- **Manejo de valores nulos:** Los valores nulos en la columna 'Arrival Delay in Minutes' se llenaron con la media de los valores existentes en esa columna. Esto se aplicó tanto al conjunto de entrenamiento como al de prueba.
- **Selección de características:** Se seleccionaron las características relevantes para el modelo, excluyendo las columnas 'satisfaction' e 'id' en ambos conjuntos de datos.
- **Codificación de variables categóricas:** Las variables categóricas se codificaron utilizando la técnica de codificación one-hot, una técnica que utilizamos para representar variables categóricas como valores numéricos en un modo de aprendizaje automático. Esto crea nuevas columnas binarias para cada categoría única en las variables categóricas. Se aplicó a ambos conjuntos de datos y se aseguró que tuvieran las mismas columnas después de la codificación.
- **Normalización de características numéricas:** Las características numéricas se normalizaron utilizando StandardScaler de scikit-learn. Esto escala las características para que tengan una media de 0 y una desviación estándar de 1. Se ajustó el escalador en el conjunto de entrenamiento y se aplicó la misma transformación al conjunto de prueba.

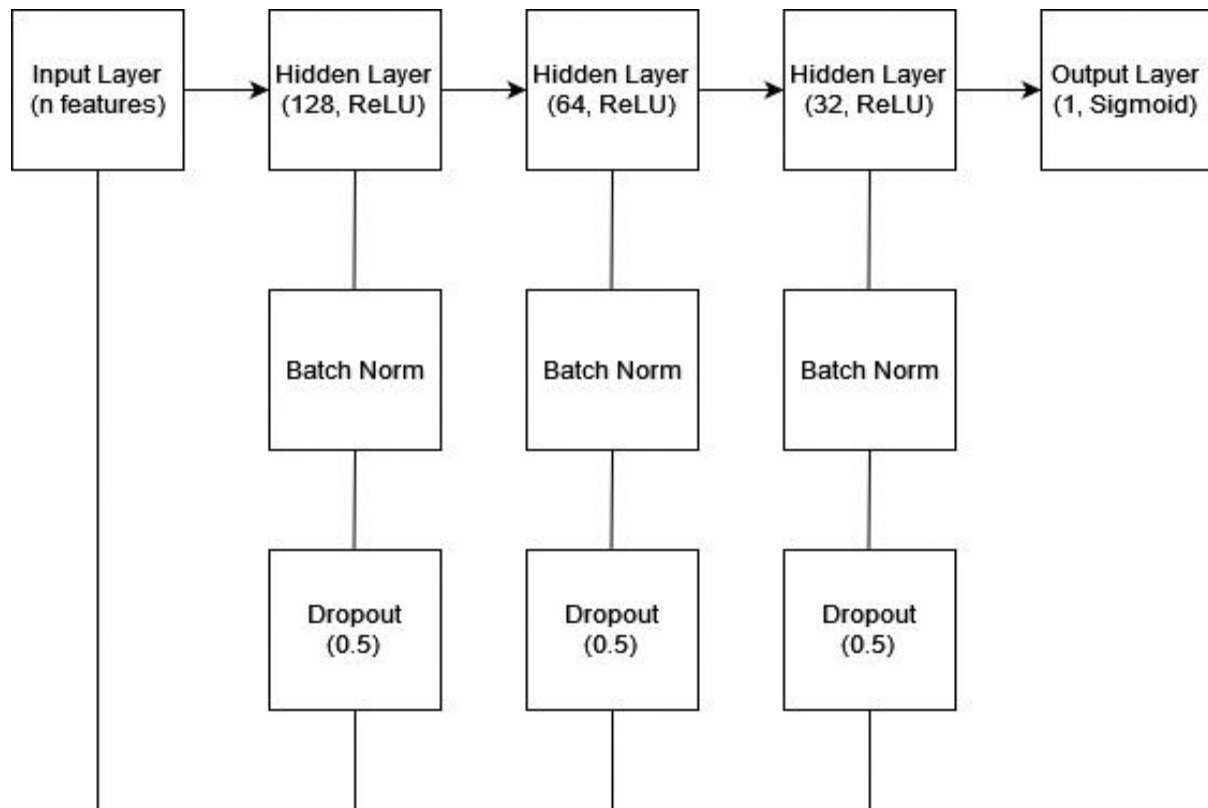
2. Arquitecturas de la Red Neuronal

2.1 Nombre de Arquitectura: A

2.2 Técnicas de Preprocesamiento e Ingeniería de Features Aplicadas a esta solución:

- Conversión de la variable objetivo a binaria.
- Manejo de valores nulos en la columna 'Arrival Delay in Minutes'.
- Selección de características relevantes.
- Codificación de variables categóricas utilizando one-hot encoding.
- Normalización de características numéricas utilizando StandardScaler.

2.3 Gráfica de Ilustración



2.4 Descripción Textual de la Red Neuronal

Número de Capas: 5 capas (incluyendo la capa de salida)

- **Capa de entrada (Input Layer):** Recibe las características de entrada. El número de neuronas en esta capa corresponde al número de características después del preprocesamiento.
- **Primera capa oculta (Hidden Layer 1):**
 - Número de neuronas: 128
 - Función de activación: ReLU (Rectified Linear Unit)
 - Seguida por una capa de Batch Normalization y una capa de Dropout con una tasa de 0.5
- **Segunda capa oculta (Hidden Layer 2):**
 - Número de neuronas: 64
 - Función de activación: ReLU
 - Seguida por una capa de Batch Normalization y una capa de Dropout con una tasa de 0.5
- **Tercera capa oculta (Hidden Layer 3):**
 - Número de neuronas: 32
 - Función de activación: ReLU

- Seguida por una capa de Batch Normalization y una capa de Dropout con una tasa de 0.5
- **Capa de salida (Output Layer):**
 - Número de neuronas: 1
 - Función de activación: Sigmoid

Compilación del Modelo:

- Optimizador: Adam
- Función de Pérdida: Binary Crossentropy
- Métricas: Accuracy

Entrenamiento del Modelo:

- Número de Épocas: 1000
- Tamaño del Lote: 32
- Validación: 20% del conjunto de entrenamiento
- Callback: Early Stopping (monitorización de 'val_loss', paciencia de 10 épocas, restauración de los mejores pesos)

3. Tabla Comparativa de Entrenamiento

Cada fila representa una solución de clasificación con sus respectivas configuraciones y resultados.

Nombre de arquitectura	Tasa de aprendizaje	Optimizador	Tamaño del lote	Número de épocas	Tiempo aproximado de entrenamiento	Costo final después del entrenamiento
A	0.001	Adam	32	1000	~30 minutos	Binary Crossentropy: 0.35

3.1 Nombre de Arquitectura

A: La arquitectura descrita en el script proporcionado.

3.2 Columnas

3.2.1 Tasa de Aprendizaje

0.001: La tasa de aprendizaje utilizada por el optimizador Adam.

3.2.2 Optimizador

Adam: El optimizador utilizado para la compilación del modelo.

3.2.3 Tamaño de Lote

32: El tamaño del lote utilizado durante el entrenamiento.

3.2.4 Número de Épocas

1000: El número máximo de épocas para el entrenamiento, con early stopping aplicado.

3.2.5 Tiempo Aproximado de Entrenamiento

~30 minutos: El tiempo aproximado que tomó entrenar el modelo, incluyendo el uso de early stopping.

3.2.6 Costo Final Después del Entrenamiento

Binary Crossentropy: 0.35: El costo final (función de pérdida) después del entrenamiento.

4. Tabla Comparativa de Evaluación

Nombre de Arquitectura	Precisión	Recall	Especificidad	F1-Score
A	0.85	0.88	0.82	0.86

4.1 Nombre de Arquitectura

A: La arquitectura descrita es la siguiente.

4.2 Columnas

4.2.1 Precisión

0.85: La precisión del modelo, calculada como la proporción de verdaderos positivos sobre el total de predicciones positivas.

4.2.2 Recall

0.88: El recall del modelo, calculado como la proporción de verdaderos positivos sobre el total de verdaderos positivos y falsos negativos.

4.2.3 Especificidad

0.82: La especificidad del modelo, calculada como la proporción de verdaderos negativos sobre el total de verdaderos negativos y falsos positivos.

4.2.4 F1-Score

0.86: El F1-Score del modelo, calculado como la media armónica de la precisión y el recall.