

Trabajo Práctico Inicial

Plataforma de Gestión de Recursos Humanos con ERP y CRM con IA

Cuarta Entrega



Universidad
Nacional de
General
Sarmiento



Integrantes del Equipo 6

Gaspar Abel Aquino

Gmail: abelaquino02@gmail.com

Cristian Jurajuria

Gmail: cljurajuria23@gmail.com

Lautaro Emanuel Moreno

Gmail: lemoreno2002@gmail.com

Rodrigo Montoro

Gmail: rodrigo.montoro@hotmail.com

Pablo Samuel Da Silva

Gmail: polilladasilva0@gmail.com

Link del repositorio

<https://github.com/LautaroMoreno2002/TPI-G6-LCS>

1. 🧠 Descripción del modelo

Para la resolución de este Trabajo Práctico Inicial, implementamos un **modelo de clasificación** que se encarga de predecir el canal de notificación más adecuado, donde se reciben avisos de eventos como cumpleaños o aniversarios laborales, para cada empleado mediante la información de la persona y su contexto laboral (Slack, Correo electrónico o Push).

El modelo procesa las siguientes tres variables:

- **Edad**: Puede correlacionar con la adopción tecnológica o hábitos de comunicación generacionales.
- **Antigüedad**: La experiencia en la empresa podría influir en la familiaridad o preferencia por ciertos canales establecidos.
- **Departamento**: El rol o área funcional puede dictar las herramientas de comunicación primarias utilizadas en el día a día.

Con esos datos ingresados en el programa, el sistema aprende a reconocer patrones entre los empleados registrados y sus preferencias de canales de comunicación.

El modelo de clasificación utilizado para la predicción de los canales de comunicación fue **Árboles de Decisión**, ya que permite generar reglas claras que toma el modelo para generar reglas claras del tipo:

"Si el empleado tiene menos de 30 años y trabaja en el departamento de TI, probablemente prefiera recibir notificaciones Slack".

Elegimos este modelo porque:

- Es fácil de entender, interpretar y explicar.
- Funciona bien con datos pequeños y estructurados. Los datos pueden ser numéricos como categóricos.
- Permite visualizar cómo se toman las decisiones y cómo se llegan a los resultados de cada predicción.

2. 🛠️ Instrucciones para ejecutar el sistema

Para poner en funcionamiento el sistema, se deben seguir estos pasos:

Instalar el entorno de trabajo:

Es necesario tener instalado Python y algunas bibliotecas que usamos, esto se puede hacer fácilmente con el comando:

```
pip install -r requirements.txt
```

El archivo requirements.txt contiene las siguientes librerías:

- pandas
- numpy
- scikit-learn
- matplotlib
- joblib

- django
- openpyxl

1. Ejecutar la aplicación:

Una vez instaladas las dependencias y teniendo el entorno python creado, abriremos la terminal en la carpeta donde se aloja el archivo “`manage.py`” y colocaremos el siguiente comando:

```
python manage.py runserver
```

Como estamos trabajando con Django, tenemos todo nuestro sistema web escrito en Python pero para verlo funcionar en el navegador, necesitamos levantar el servidor que lo ejecute.

Luego dejamos el servidor corriendo y hacemos click en la URL de nuestro servidor local que por defecto va a ser `http://127.0.0.1:8000/`

2. Cargar datos para predicción:

Prediccion individual:

El usuario debe ingresar los datos básicos de la persona (edad, antigüedad y departamento), y luego darle al botón de “`Predecir preferencia de empleado`”.

Dicho botón lo redirigirá a otra pantalla la cual mostrará primeramente la información del empleado del cual solicitó la predicción, las características y ventajas del medio de comunicación predicho (Slack, Notificación Push o Correo).

Predicción múltiple:

El usuario en este apartado le da al botón de “`Seleccionar archivo`”

Debe subir los datos de sus empleados en un archivo Excel con terminación .xlsx. La tabla debe estar dividida por columnas donde se contengan los datos: Nombre - Apellido - Edad - Antigüedad - Departamento.

Finalmente le da al botón de “`Predecir preferencia de empleados`” y en la parte inferior se podrá visualizar la información en una tabla, con los datos que el usuario subió al excel y una columna más indicando el canal de comunicación de preferencia de dicho empleado (Slack, Notificación Push o Correo).

Todo está pensado para que cualquier persona pueda utilizarlo sin conocimientos técnicos. La interfaz es amigable con solapas claves y botones reconocibles que van a la acción en concreto sin muchas vueltas, solo hay que cargar los datos solicitados, la información de un empleado en concreto o el archivo excel con la información de varios empleados.

3. Cómo entrenar el modelo

El sistema está basado en un modelo de aprendizaje automático que **necesita ejemplos para aprender**. Considerando el objetivo de predecir una categoría (el canal preferido) a partir de características del empleado, se determinó que un enfoque de clasificación supervisada dentro del Machine Learning era el más adecuado. Tras una exploración inicial de posibles algoritmos, se optó por implementar un modelo basado en Árboles de Decisión.

¿Cómo aprende el modelo?

- Se alimenta con datos estructurados:
 - Cada empleado en el dataset sintético tiene:
 - Características (X): Edad, Antigüedad, Departamento.
 - Etiqueta (y): Preferencia (ej: "Correo", "Slack", "Push").
- Identifica patrones ocultos:
 - El modelo analiza cómo se relacionan las características con las preferencias.
 - Ejemplo de regla aprendida:
 - SI Edad < 30 Y Departamento = "IT" → Preferencia = "Slack" (confianza: 92%).
- Se separan los datos en prueba y entrenamiento:
 - Entrenamiento: 70%
 - Prueba: 30%
- Optimiza sus decisiones:
 - Usa un algoritmo de Árbol de Decisión para dividir los datos en reglas jerárquicas (como un "flujo de preguntas y respuestas").

Proceso de entrenamiento:

- Creamos un conjunto de datos sintéticos con 50 empleados ficticios.
- Cada fila tiene: edad, antigüedad, departamento y canal preferido.
- Con esa información, entrenamos al modelo para que aprenda las relaciones.

El entrenamiento se hace una única vez, y genera un archivo con el modelo ya listo para usar. Si se agregan nuevos datos o se quiere mejorar el sistema, se puede repetir el proceso de entrenamiento fácilmente.

4. Cómo hacer predicciones(Resumen)

Una vez entrenado, el modelo está listo para recibir nuevos datos de los empleados y predecir automáticamente el canal de notificación más adecuado para cada uno.

El sistema ofrece dos formas de hacer predicciones :

- Predicción unitaria: Se deben ingresar los datos de un empleado a través de un formulario y el sistema predice el canal de comunicación que prefiere dicho empleado. Además de la preferencia, se muestra una breve descripción del canal y

sus principales ventajas.

- Predicción múltiple: Si se desea predecir las preferencias de varios empleados a la vez, se puede cargar un archivo “.x/sx” (tabla en excel) que contenga filas con los datos (nombre, apellido, edad, antigüedad y departamento) de todos ellos. Luego, el sistema procesa ese archivo y mostrará una tabla con una columna adicional que indica el canal sugerido para cada caso.

En ambos métodos, el sistema:

- Toma como entrada la edad, la antigüedad y el departamento del empleado (para predecir no tiene en cuenta las variables nombre y apellido).
- Analiza esa información según los patrones que aprendió durante el entrenamiento.
- Devuelve una sugerencia automática: Si prefiere recibir los avisos por medio de Correo electrónico, Slack o Notificación Push.

Esta funcionalidad permite que las empresas envíen comunicaciones internas de manera más personalizada y efectiva, aumentando así la probabilidad de que los mensajes sean recibidos, leídos y valorados por cada persona.