

Syllabus: Fundamentos de Machine Learning para IOAI

Programa de Entrenamiento para Competencias Internacionales

Versión 1.0 | Enero 2024 Público objetivo: Estudiantes escolares sin experiencia previa en programación.

Información General del Curso

Detalle	Especificación
Duración	10 clases de 2 horas (20 horas totales)
Prerrequisitos	Conocimientos básicos de matemáticas escolares
Modalidad	Práctica, con sesiones en Google Colab
Lenguaje	Python 3.8+
Evaluación	Ejercicios en clase (40%), Proyecto final (60%)

Objetivos del Primer Módulo (10 Clases)

Al finalizar este módulo, los estudiantes serán capaces de:

- Programar en Python usando condicionales, bucles y funciones.
- Manipular y limpiar datos básicos con Pandas.
- Crear gráficos simples con Matplotlib.
- Entrenar y evaluar un modelo básico de clasificación con Scikit-learn.

Plan de Clases Detallado

Clase 1: Primeros Pasos con Python y Google Colab (3h)

Objetivo: Familiarizarse con el entorno de trabajo y ejecutar código básico.

Contenido y Actividades (Todo Práctico):

1. **Configuración (30 min):** Crear cuenta en Google, acceder a Colab, crear un nuevo Notebook, entender celdas de código y texto.
2. **Hola Mundo (30 min):** Usar `print()`, crear variables (`nombre = "Ana"`), tipos de datos básicos (texto, número), y operaciones aritméticas simples.
3. **Ejercicios varios (1h):** Pequeños ejercicios: pedir nombre y año de nacimiento para calcular edad, formatear salida, y practicar conversión de tipos usando `print()` e `input()`.

Clase 2: Tomando Decisiones con `if`, `elif`, `else` (3h)

Objetivo: Controlar el flujo del programa usando comparaciones y lógica.

Contenido y Actividades:

1. **Fundamentos (45 min):** Sintaxis de `if`. Operadores de comparación (`==`, `>`, `<`, `!=`). Condición `else`. Ejercicios guiados: “¿Puedo votar?” (edad ≥ 18).
2. **Decisiones Complejas (45 min):** Uso de `elif`. Operadores lógicos `and` y `or`. Ejercicios: Clasificar notas (A, B, C, D) o determinar acceso a un descuento (estudiante Y edad < 25).
3. **Ejercicios varios (30 min):** Pequeños ejercicios: crear un sistema simple de login que verifique un usuario y contraseña predefinidos y muestre mensajes personalizados.

Clase 3: Repetición con Bucles `for` y `while` (3h)

Objetivo: Automatizar tareas repetitivas y trabajar con listas de datos.

Contenido y Actividades:

1. **Bucle `for` (1h):** Iterar sobre una lista (`for amigo in ["Juan", "Ana"]:`). Usar `range()` para generar secuencias numéricas. Ejercicios: sumar números en una lista, imprimir la tabla de multiplicar de un número.
2. **Bucle `while` (45 min):** Concepto de condición de parada. Diferencia clave con `for`. Ejercicios: adivinar un número secreto con intentos ilimitados, sumar números hasta que el usuario ingrese 0.
3. **Comparación (15 min):** ¿Cuándo usar `for` y cuándo `while`? Discusión con ejemplos.

Clase 4: Organizando Código con Funciones (3h)

Objetivo: Escribir código reusable y mantenible mediante funciones.

Contenido y Actividades:

1. **Funciones Básicas (1h):** Definir una función con `def`. Llamar a una función. Parámetros y argumentos. Retornar un valor con `return`. Ejercicios: función para saludar, función para calcular el área de un círculo.
2. **Funciones en la Práctica (1h):** Refactorizar el código de los ejercicios de las clases 2 y 3 para usar funciones. Crear una función que reciba una lista y devuelva su promedio.

Clase 5: Datos en Tablas con Pandas I - Exploración (3h)

Objetivo: Cargar datos reales y realizar una exploración inicial.

Contenido y Actividades:

1. **Introducción a DataFrames (45 min):** ¿Qué es un DataFrame? Cargar un archivo CSV (`pd.read_csv`). Dataset práctico: “titanic.csv” o “iris.csv”. Ver las primeras filas (`.head()`), dimensiones (`.shape`).
2. **Exploración Básica (1h 15min):** Obtener estadísticas con `.describe()`. Contar valores únicos con `.value_counts()`. Seleccionar columnas. Introducción a NaN (valores faltantes) y cómo contarlos con `.isna().sum()`.

Clase 6: Datos en Tablas con Pandas II - Limpieza y Filtrado (3h)

Objetivo: Preparar los datos para el análisis filtrando y limpiando información.

Contenido y Actividades:

1. **Filtrado (1h):** Seleccionar filas que cumplan condiciones (ej: `df[df['edad'] > 18]`). Usar múltiples condiciones con `&` (and) y `|` (or). Ejercicio: filtrar pasajeros del Titanic de primera clase que sobrevivieron.
2. **Limpieza Básica (1h):** Eliminar filas con valores faltantes (`.dropna()`). Rellenar valores faltantes con un número (`.fillna(0)`). Crear nuevas columnas a partir de otras.

Clase 7: Visualización de Datos con Matplotlib (3h)

Objetivo: Comunicar hallazgos a través de gráficos simples.

Contenido y Actividades:

1. **Gráficos de Barras y Línea (1h):** Crear un gráfico de barras para mostrar la cantidad de sobrevivientes por clase en el Titanic. Crear un gráfico de línea para una serie temporal simple (ej: ventas por mes).
2. **Histogramas y Dispersión (1h):** Crear un histograma para ver la distribución de edades. Crear un gráfico de dispersión (`scatter`) para ver la relación entre dos variables (ej: longitud vs ancho del pétalo en Iris).

Clase 8: Introducción al Machine Learning con Scikit-learn (3h)

Objetivo: Entender el flujo de trabajo básico para entrenar un primer modelo.

Contenido y Actividades:

1. **Conceptos Clave (30 min):** ¿Qué es un modelo? Diferencia entre entrenar y predecir. Features (X) y target (y). Train y Test sets.
2. **Primer Modelo: Regresión Logística (1h 30min):**
 - Preparar los datos: seleccionar features y target del DataFrame, eliminar filas con NaN.
 - Dividir en entrenamiento (train) y prueba (test) con `train_test_split`.
 - Crear, entrenar (`model.fit`) y predecir (`model.predict`) con `LogisticRegression`.
 - Evaluar: calcular la **precisión (accuracy)** comparando predicciones con valores reales del test set.

Clase 9: Evaluación de Modelos y Matriz de Confusión (3h)

Objetivo: Ir más allá del “accuracy” para entender cómo y cuándo falla el modelo.

Contenido y Actividades:

1. **Matriz de Confusión (1h):** Entender Verdaderos Positivos/Negativos y Falsos Positivos/Negativos. Calcularla con `confusion_matrix`. Interpretarla en el contexto del problema (ej: predecir supervivencia).
2. **Métricas Derivadas (1h):** Calcular **Precisión (precision)** y **Exhaustividad (recall)** a partir de la matriz. Discutir su importancia (ej: en un modelo médico, es peor un falso negativo que un falso positivo).

Clase 10: Proyecto Integrador Guiado (3h)

Objetivo: Aplicar todo lo aprendido en un flujo de trabajo completo.

Actividad (Práctica Supervisada): Desarrollar, en conjunto con el instructor, un proyecto desde cero usando un dataset nuevo (ej: “diabetes.csv” para predecir diagnóstico).

1. Cargar y explorar los datos.
2. Realizar limpieza y selección de features.
3. Entrenar un modelo de Regresión Logística.
4. Evaluar su desempeño con Accuracy y Matriz de Confusión.
5. Discutir posibles mejoras.

Proyecto Final del Módulo

Consigna: Individualmente, cada estudiante aplicará el flujo completo de las Clases 5-10 a un dataset a elección (aprobado por el instructor).

Entregables:

1. Un Notebook de Google Colab que contenga:
 - Análisis exploratorio con al menos 2 visualizaciones.
 - Limpieza y preparación de los datos.
 - Entrenamiento y evaluación de un modelo de Regresión Logística.
 - Comentarios que expliquen cada paso.
2. Una explicación breve (oral o escrita) de los resultados.

Criterios de Evaluación:

- Correcta aplicación de los conceptos técnicos (40%).
 - Claridad y organización del código (30%).
 - Análisis e interpretación de los resultados (30%).
-

Este syllabus está diseñado para un aprendizaje progresivo y realista, priorizando la comprensión sólida de los fundamentos sobre la velocidad.