### 

|  |  |
| --- | --- |
|  | **Actividades de capacitación** |

***Información para la presentación de propuesta de Cursos y talleres con modalidad virtual***

1.- Nombre de la Actividad (Sintético): **Introducción a la Programación en Python para Aprendizaje por Refuerzo**

2.- Objetivos

* + Brindar una introducción sólida a la programación en Python con énfasis en las herramientas necesarias para el Aprendizaje por Refuerzo.
  + Familiarizar a los participantes con estructuras de datos, manejo de librerías numéricas y procesamiento de datos.
  + Introducir bibliotecas de aprendizaje automático y entornos de simulación utilizados en RL (Scikit-learn, TensorFlow, OpenAI Gym).

3.- Breve Fundamentación: Este curso es una condición previa para los participantes del curso "Introducción al Aprendizaje por Refuerzo desde una Perspectiva Estadística", asegurando que posean las competencias esenciales en programación y el manejo de herramientas computacionales en Python. Se enfoca en desarrollar una base sólida en programación estructurada y orientada a objetos, manipulación y procesamiento de datos, así como en la visualización de información.

Además, se introduce a los participantes en el ecosistema científico de Python, incluyendo el uso de bibliotecas fundamentales como NumPy, Pandas y Matplotlib, esenciales para la manipulación y análisis de datos. También se familiariza a los estudiantes con herramientas avanzadas como Scikit-learn y TensorFlow para el desarrollo de modelos de aprendizaje automático. Finalmente, se presenta OpenAI Gym para la experimentación con entornos simulados en Aprendizaje por Refuerzo, asegurando una transición fluida al curso principal.

4.- Contenidos

* + **Módulo 1: Fundamentos de Python**
    - Introducción al lenguaje Python: historia, características y aplicaciones.
    - Sintaxis básica y estructura de un programa en Python.
    - Estructuras de control: condicionales (if, elif, else) y bucles (for, while).
    - Tipos de datos primitivos: enteros, flotantes, cadenas de texto, booleanos.
    - Estructuras de datos avanzadas: listas, diccionarios, tuplas, conjuntos.
    - Operaciones y manipulación de datos con estructuras de datos.
    - Buenas prácticas de codificación y estilo en Python (PEP8).
  + **Módulo 2: Programación Funcional y Orientada a Objetos en Python**
    - **Funciones en Python:** Definición, parámetros, valores de retorno y funciones anónimas (lambda).
    - **Decoradores:** Uso y creación de decoradores para modificar el comportamiento de funciones.
    - **Programación funcional:** Uso de funciones de orden superior (map, filter, reduce), expresiones lambda y listas por comprensión.
    - **Clases y objetos:** Conceptos de programación orientada a objetos (POO), creación de clases y objetos, encapsulación, herencia y polimorfismo.
    - **Métodos y atributos:** Métodos de instancia, clase y estáticos.
    - **Manejo de excepciones en POO:** Uso de try, except, finally y la creación de excepciones personalizadas.
    - **Patrones de diseño básicos en Python:** Singleton, Factory, y Strategy aplicados a la POO.
  + **Módulo 3: Manipulación de Datos con Pandas y NumPy**
    - **Lectura y escritura de datos:** Carga de datos desde archivos CSV, Excel, JSON y bases de datos SQL.
    - **Manipulación de DataFrames con Pandas:** Filtrado, indexación, operaciones con columnas y filas.
    - **Operaciones avanzadas con Pandas:** Agrupamiento (groupby), combinación de datasets (merge, concat), manejo de valores nulos y duplicados.
    - **Operaciones con arrays NumPy:** Creación y manipulación de arrays, indexación y slicing, operaciones vectorizadas y broadcasting.
    - **Análisis estadístico con Pandas y NumPy:** Cálculo de estadísticas descriptivas, correlaciones y transformaciones de datos.
    - **Optimización de rendimiento:** Uso de apply vs. vectorization, técnicas para mejorar la eficiencia en el procesamiento de datos.
  + **Módulo 4: Visualización de Datos con Matplotlib y Seaborn**
    - **Introducción a la visualización de datos:** Importancia y aplicaciones en ciencia de datos.
    - **Uso de Matplotlib:** Creación de gráficos de líneas, dispersión, histogramas y gráficos de barras.
    - **Personalización de gráficos:** Etiquetas, títulos, leyendas, escalas y estilos.
    - **Uso de Seaborn:** Gráficos avanzados, mapas de calor, violin plots y gráficos de distribución.
    - **Visualización interactiva:** Uso de plotly y seaborn para gráficos dinámicos.
    - **Casos prácticos:** Análisis exploratorio con gráficos y dashboards básicos.
  + **Módulo 5: Introducción a Scikit-learn y TensorFlow**
    - **Conceptos básicos de aprendizaje automático:** Definición, tipos de aprendizaje (supervisado, no supervisado y por refuerzo), importancia y aplicaciones.
    - **Scikit-learn:**
      * Instalación y configuración.
      * Preprocesamiento de datos (StandardScaler, MinMaxScaler, OneHotEncoder).
      * Modelos básicos de clasificación y regresión (LinearRegression, LogisticRegression, DecisionTreeClassifier).
      * Evaluación de modelos (train\_test\_split, cross\_val\_score, confusion\_matrix).
    - **Introducción a TensorFlow:**
      * Conceptos fundamentales de redes neuronales artificiales.
      * Creación de modelos de aprendizaje profundo con Keras.
      * Entrenamiento, evaluación y ajuste de hiperparámetros.
      * Implementación de redes neuronales simples y convolucionales.
    - **Comparación entre Scikit-learn y TensorFlow:** Casos de uso y cuándo elegir cada herramienta.
  + **Módulo 6: Introducción a OpenAI Gym y Simulaciones**
    - **Conceptos básicos de OpenAI Gym:** Instalación y configuración del entorno.
    - **Uso de entornos simulados:** Introducción a los espacios de observación y acción.
    - **Implementación de agentes básicos:** Definición de políticas aleatorias y heurísticas simples.
    - **Interacción con entornos de OpenAI Gym:** Ejecución de episodios y recolección de datos.
    - **Evaluación del desempeño de los agentes:** Cálculo de recompensas y métricas de rendimiento.
    - **Extensión a entornos avanzados:** Integración con TensorFlow y PyTorch para la implementación de agentes inteligentes.
    - **Ejercicios prácticos:** Creación de un agente de control en un entorno de OpenAI Gym.

5.- Metodología.

* + Clases teórico-prácticas con ejemplos y ejercicios guiados.
  + Uso de Google Colaboratory para la práctica de código, permitiendo ejecución en la nube sin necesidad de instalación local.
  + Desarrollo de notebooks interactivos con ejercicios autocontenidos y explicaciones paso a paso.
  + Uso de datasets para ejercicios prácticos, promoviendo la aplicación de los conceptos en problemas del mundo real.
  + Integración con bibliotecas avanzadas y herramientas colaborativas como Google Drive y GitHub para facilitar el acceso y la gestión del código.
  + Proyectos aplicados en cada módulo, con enfoque en resolución de problemas y aplicación de técnicas específicas de programación en Python.

6.- Duración del Curso:

* **Cantidad total de clases:** 6
* **Cantidad de horas reloj por clase:** 3 horas
* **Cantidad total de horas reloj del Curso:** 18 horas

7.- Beneficiarios del Curso:

* Estudiantes, investigadores y profesionales interesados en Aprendizaje por Refuerzo, Inteligencia Artificial y Ciencia de Datos.
* Profesionales de áreas como estadística, matemáticas, ingeniería, economía y ciencias computacionales que deseen adquirir habilidades en programación para su aplicación en modelos de aprendizaje automático.
* Estudiantes de nivel universitario o posgrado que necesiten desarrollar competencias en Python como requisito previo para abordar algoritmos avanzados en Aprendizaje por Refuerzo.
* Requisitos:
  + Conocimientos básicos de matemática, incluyendo álgebra lineal, probabilidad y estadística.
  + Familiaridad con conceptos de optimización y modelos estadísticos.
  + No es necesario conocimiento previo en Python, pero se recomienda tener nociones básicas de programación en cualquier lenguaje.

8.- Requisitos de Aprobación:

* **Asistencia:** Se requiere una asistencia mínima del 75% a las clases para asegurar la participación activa en el curso.
* **Ejercicios prácticos:** Los participantes deberán completar ejercicios semanales obligatorios, que incluyen problemas de programación en Python y análisis de datos utilizando las bibliotecas presentadas.
* **Proyecto final:** Se deberá entregar un proyecto integrador en el que los estudiantes apliquen los conocimientos adquiridos en un problema práctico, el cual será evaluado en función de su correcta implementación, claridad en el código y presentación de resultados.
* **Participación en foros y debates:** Se valorará la participación en discusiones y la resolución de dudas en el entorno colaborativo de Google Colaboratory.
* **Evaluaciones parciales:** Se incluirán pruebas de autoevaluación al finalizar cada módulo para verificar la comprensión de los temas antes de continuar con los siguientes.

9.- Precisar cupos máximos y mínimos de los participantes.

* Mínimo: 15 participantes
* Máximo: 50 participantes

10.- Aclarar características de la certificación:

Se entregará certificado de aprobación a los participantes que cumplan con los requisitos de evaluación.

11.- Bibliografía (adjuntar con el programa si es necesario).

**Chollet, F. (2021). Deep Learning with Python (2ª ed.). Manning Publications.** Esta segunda edición ofrece una introducción práctica al aprendizaje profundo con Keras y TensorFlow, abordando redes neuronales desde conceptos básicos hasta arquitecturas avanzadas.

**Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.** Considerado una referencia fundamental en el campo, este libro proporciona una comprensión profunda de los conceptos y técnicas del aprendizaje profundo.

**McKinney, W. (2022). Python para análisis de datos: Manipulación de datos con pandas, NumPy y Jupyter (3ª ed.). Anaya Multimedia.** Esta tercera edición, actualizada para Python 3.10 y pandas 1.4.0, ofrece casos prácticos que permiten resolver una amplia variedad de problemas de datos de manera efectiva. Es una introducción práctica y moderna a las herramientas de ciencia de datos que ofrece Python

**Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2ª ed.). MIT Press.** Esta segunda edición actualiza y amplía la cobertura de la teoría y práctica del aprendizaje por refuerzo, incluyendo desarrollos recientes en el campo.

**VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.** Aunque publicado en 2016, este recurso sigue siendo esencial para la ciencia de datos con Python, cubriendo bibliotecas clave como NumPy, Pandas, Matplotlib, Scikit-learn y técnicas avanzadas de análisis.

***Documentación oficial de bibliotecas de Python***

**Matplotlib Developers. (2024).** Matplotlib Documentation. Disponible en: <https://matplotlib.org/stable/contents.html>

**NumPy Developers. (2024).** NumPy Documentation. Disponible en: <https://numpy.org/doc/stable/>

**OpenAI Developers. (2024).** OpenAI Gym Documentation. Disponible en: <https://www.gymlibrary.dev/>. Una guía detallada sobre el uso de Python para análisis de datos, incluyendo manipulación de datos con Pandas y NumPy, técnicas de procesamiento y estrategias para mejorar el rendimiento en análisis de datos.

**Pandas Developers. (2024).** Pandas Documentation. Disponible en: <https://pandas.pydata.org/docs/>

**Scikit-learn Developers. (2024).** Scikit-learn Documentation. Disponible en: <https://scikit-learn.org/stable/documentation.html>

**TensorFlow Developers. (2024).** TensorFlow Documentation. Disponible en: <https://www.tensorflow.org/api_docs>

12.- Recursos materiales y equipos tecnológicos que se necesitan para su dictado.

* **Computadora con acceso a internet:** Se recomienda cualquier computadora con conexión a internet y un navegador web actualizado, ya que todo el desarrollo del curso se realizará en Google Colaboratory, evitando la necesidad de hardware avanzado o configuraciones locales.
* **Instalación de Python 3.x y bibliotecas necesarias:**
  + **NumPy:** Para operaciones numéricas y manipulación de arrays multidimensionales.
  + **Pandas:** Para manipulación y análisis de datos estructurados.
  + **Matplotlib y Seaborn:** Para visualización de datos y generación de gráficos interactivos.
  + **Scikit-learn:** Para aprendizaje automático y modelado estadístico.
  + **TensorFlow:** Para el desarrollo de modelos de aprendizaje profundo.
  + **OpenAI Gym:** Para la implementación y simulación de entornos de aprendizaje por refuerzo.
* **Uso de Google Colaboratory:**
  + Permite la ejecución de código en la nube sin necesidad de instalación local.
  + Compatible con bibliotecas avanzadas de Python.
  + Facilita la colaboración en tiempo real y la integración con Google Drive y GitHub.
  + Se recomienda familiarizarse con Jupyter Notebooks y su entorno de trabajo para optimizar el uso de Google Colab.

13.- Currículum Vitae del / los docente/s encargado / s de la Actividad (adjuntar CV breve).

### ****Currículum Vitae – Dr. Darío Ezequiel Díaz****

**Contacto:**

* **Correo:** drdarioezequieldiaz@gmail.com / ddiaz@indec.gob.ar
* **Ubicación:** Misiones, Argentina
* **Ocupación Actual:** Director de Metodología y Relevamiento Estadístico, Instituto Provincial de Estadística y Censos de Misiones

### ****Formación Académica****

* **Doctorado en Estadística** – Universidad Nacional de Rosario (en curso).
* **Posdoctorado en Retos Poblacionales Actuales** – Universidad Nacional de Córdoba (2022)
* **Doctor en Ciencias Económicas con Mención en Economía** – Universidad Nacional de Córdoba (2014)
* **Maestría en Métodos Cuantitativos para la Gestión y Análisis de Datos en Organizaciones** – Universidad de Buenos Aires **(Todos los cursos y seminarios aprobados, tesis en elaboración).**
* **Maestría en Explotación de Datos y Gestión del Conocimiento – Universidad Austral (Todos los cursos y seminarios aprobados, tesis en elaboración).**
* **Maestría en Econometría – Universidad Torcuato Di Tella (en curso).**
* **Maestría en Políticas Públicas y Desarrollo – FLACSO (Todos los cursos y seminarios aprobados, tesis en elaboración).**
* **Maestría en Políticas Económicas, Sociales y Regionales-Universidad Nacional de Córdoba (en curso)**
* **Licenciado en Economía – Universidad Nacional de Misiones (2008)**

### ****Experiencia Profesional****

* **Director de Metodología y Relevamiento Estadístico** – Instituto Provincial de Estadística y Censos de Misiones (2016 - Actualidad).
* **Docente Universitario** en diversas instituciones, con cursos en estadística, econometría, ciencia de datos y programación en Python.
* **Investigador en Ciencia de Datos y Aprendizaje Automático** en proyectos sobre eficiencia económica y modelado predictivo.
* **Experiencia en Capacitación**: Dictado de cursos y talleres sobre ciencia de datos, análisis de datos en Python y aprendizaje automático.

### ****Habilidades Técnicas****

* **Programación:** Python (Polars, Pandas, NumPy, TensorFlow, Scikit-learn), R, Stata, Eviews, SPSS,
* **Aprendizaje Automático:** Modelos supervisados, no supervisados y aprendizaje por refuerzo.
* **Análisis de Datos:** Procesamiento y modelado de datos con bibliotecas avanzadas.
* **Desarrollo de Modelos Matemáticos:** Aplicaciones en econometría y estadística aplicada.

### ****Publicaciones y Conferencias****

* Autor de múltiples artículos en revistas científicas sobre análisis de eficiencia, aprendizaje automático y modelado económico.
* Ponente en congresos nacionales e internacionales en estadística, economía y ciencia de datos.