

Análisis de datos

Segundo modulo

Poll Avila Arias

Portafolio de Evidencias

Docente:

Josué Zúñiga Hernández

Semana 1

Tema: Configuración de Entornos de trabajo Anaconda y Jupyter

¿Qué aprendí?

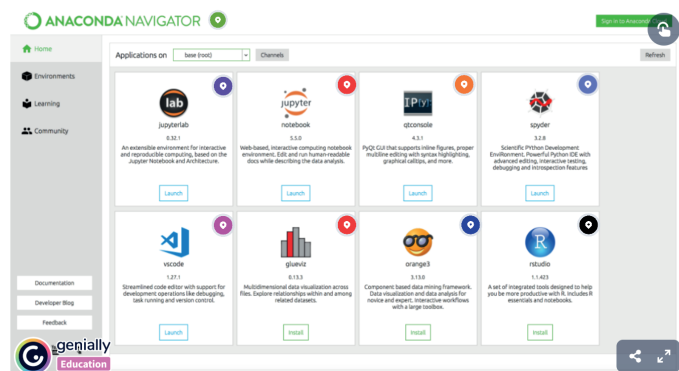
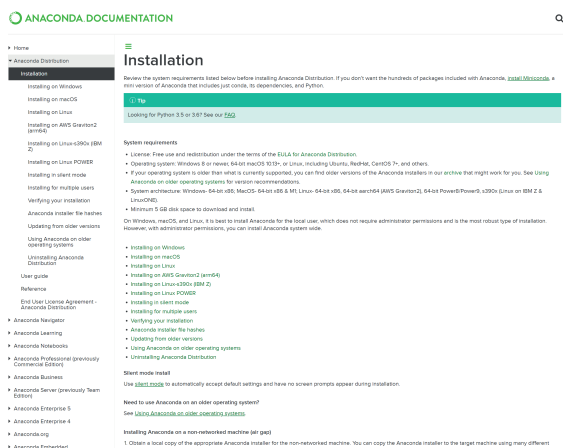
Introducción a entornos en Anaconda y Jupyter para trabajo de altos volúmenes de datos, para manejo, carga y visualización de mismos.

¿Cómo puedo aplicarlo?

Mediante estas herramientas me permitira proyectar los diferentes volúmenes de datos para una adecuada toma de decisiones en los procesos que conlleve.

Evidencias

Uso e instalación del entorno de Anaconda Navigator



Semana 2

Tema: Introducción a la metodología de Data Science

¿Qué aprendí?

Introducción a Metodología CRISP-DM y Metodología BADIR.
Introducción a HTML

¿Cómo puedo aplicarlo?

El conocimiento globalizado de un correcto uso de las metodologías para aplicación de análisis de datos me permitirá llevar a cabo un adecuado análisis, cumpliendo cada una de sus etapas para que el mismo sea eficiente y correcto

Evidencias

Metodologías y conceptos básicos

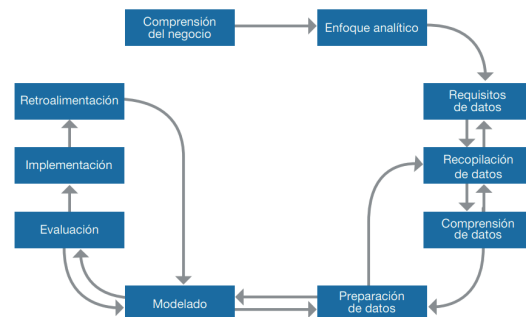
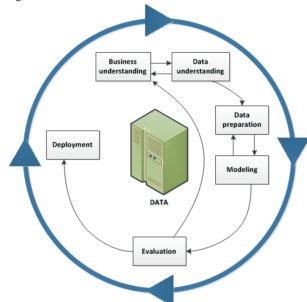
Conceptos básicos de ayuda de CRISP-DM

Última actualización: 2021-08-17

CRISP-DM, que son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para orientar sus trabajos de minería de datos.

- Como **metodología**, incluye descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas.
- Como **modelo de proceso**, CRISP-DM ofrece un resumen del ciclo vital de minería de datos.

Figura 1. Ciclo de vida de minería de datos



Semana 3

Tema: Introducción al Análisis Exploratorio de Datos (EDA)

¿Qué aprendí?

Concepto de EDA y los usos que tiene en librerías como Pandas

¿Cómo puedo aplicarlo?

Con el EDA, se pueden hallar anomalías en los datos, como valores atípicos u observaciones inusuales, revelar patrones, comprender posibles relaciones entre variables y generar preguntas o hipótesis interesantes que se pueden comprobar más adelante mediante métodos estadísticos más formales.

Evidencias

Uso exploratorio con Pandas



Análisis exploratorio
de datos con Pandas

AL MAL TIEMPO, BUENA DATA

© LACRUZ ALPACER

Guía práctica para análisis exploratorio de datos con Pandas

En la publicación anterior sobre [análisis de datos](#) mencioné el análisis exploratorio de datos, el cual nos permitirá definir si los datos que tenemos son suficientes y son los que necesitamos para cumplir con el objetivo planteado. Además, en este análisis podemos identificar si en nuestros datos hay valores nulos, atípicos, los tipos de variables, la distribución de los datos y comportamiento general de la información recopilada.

Python ha sido uno de los lenguajes de programación más utilizados por la comunidad de científicos de datos, tanto para la preparación de datos como para la implementación de modelos de machine learning y deep learning. Este lenguaje nos permite realizar un análisis exploratorio de datos mediante una de sus librerías, Pandas.



IMPORTANT METHODS IN PANDAS PACKAGE

- | | | |
|------------------------------------|---------------------------------|------------------------------|
| • <code>pd.read_csv()</code> | • <code>df.dropna()</code> | • <code>df.head()</code> |
| • <code>pd.read_table()</code> | • <code>df.fillna()</code> | • <code>df.tail()</code> |
| • <code>pd.read_excel()</code> | • <code>df.describe()</code> | • <code>df.info()</code> |
| • <code>pd.read_sql()</code> | • <code>df.sort_values()</code> | • <code>df.describe()</code> |
| • <code>pd.read_json()</code> | • <code>df.groupby()</code> | • <code>df.mean()</code> |
| • <code>pd.read_html()</code> | • <code>df.apply()</code> | • <code>df.median()</code> |
| • <code>pd.read_clipboard()</code> | • <code>df.append()</code> | • <code>df.std()</code> |
| • <code>pd.DataFrame()</code> | • <code>df.join()</code> | • <code>df.corr()</code> |
| • <code>pd.concat()</code> | • <code>df.rename()</code> | • <code>df.count()</code> |
| • <code>pd.Series()</code> | • <code>df.set_index()</code> | • <code>df.max()</code> |
| • <code>pd.date_range()</code> | • <code>df.to_csv()</code> | • <code>df.min()</code> |