**EXTRAS 1**: 6 únicas cosas que necesitas aprender para optar por Ciencias de Datos.

* **Tecnología**:
  + Python.
  + SQL.
  + R.
  + Tableau.
  + Tensorflow.
* **Conocimiento**:
  + Machine Learning.
  + Data Science.
  + Estadística.
  + Big Data.
* **No** **técnicas**:
  + Metodologías de analytics.
  + Ingles.
  + Insights (conclusiones de negocios).
  + Management.
  + Visualización.

Entonces, se pueden optar a este trabajo dominando:

**PYTHON (o R) + DATA SCIENCE + MACHINE LEARNING + ESTADÍSTICA + SQL + TABLEAU**

**EXTRAS 2**: Capacidades de negocio a aprender.

* Conceptos: Ingresos, Costes, ROI, Rentabilidad.
* Como hacer Business Case.
* Metodologías de proyectos.
* Como comunicar resultados.
* Portafolio:
  + Generación de insights.
  + Optimización de procesos.
  + Modelización predictiva.
  + Segmentación.
  + Forecasting.

**EXTRAS 3**: Estadísticas.

* **Estadística** **descriptiva**: técnicas que permiten analizar datos para entenderlos y resumirlos.
  + 4 escalas de medida, ¿Cuáles son y cuando aplicarlas?
  + Análisis de variables categóricas: por ejemplo, las más comunes frecuencias, moda, gráficos de barra, gráficos de sectores.
  + Variables de tipo continuas: por ejemplo, media, mediana, desviación típica, varianza, gráfico histograma.
  + Correlación o análisis de correlación y diferenciar entre la correlación vs causalidad.
* **Estadística inferencial**: tomar los resultados de las estadísticas descriptivas y someter a pruebas para ver si esos datos pueden ser válidos.
  + Distribuciones.
  + Distribución Normal.
  + Teorema del límite central.
  + Margen de error y nivel de confianza.
  + Alpha y pvalor.
  + Contraste de hipótesis.
* **Conceptos avanzados**:
  + Normalidad, que las variables se distribuyan normalmente.
  + Homocedasticidad de las varianzas.
  + Linealidad vs no linealidad.
  + Multicolinealidad.
  + Bootstrapping.

**INSTALACIÓN HERRAMIENTAS DE TRABAJO**:

* Instalar **Anaconda** Individual Edition.
* Se usará **Anaconda Prompt**.
* Se usará **pip** para instalar paquetes.
  + Actualizar el pip:

**> python -m pip install --upgrade pip**

* + Instalar paquetes:

**> pip install <paquete>**

**> pip install <paquete>==<version>**

* + Actualizar paquete:

**> pip install --upgrade <paquete>**

* + Ver paquetes instalados:

**> pip list**

* + Eliminar paquete:

**> pip uninstall <paquete>**

* Los paquetes necesarios son:
  + Pandas: <https://pandas.pydata.org/>

**> pip install pandas**

* + Numpy: <https://numpy.org/>

**> pip install numpy**

* + Matplotlib: <https://matplotlib.org/>

**> pip install matplotlib**

* + IPython: <https://ipython.org/>

**> pip install ipython**

* + Scikit-learn: <https://scikit-learn.org/>

**> pip install scikit-learn**

* + Tensorflow: <https://www.tensorflow.org/>

**> pip install tensorflow**

* Utilizar Google Colab.
* Las 5 etapas del análisis de datos son:
  + **Etapa 1**: Enmarcar el problema y hacer las preguntas adecuadas. Ejemplo, ¿Cuál es el objetivo de la empresa? Y ¿Qué queremos estimar o predecir?
  + **Etapa 2**: Adquirir y preparar los datos. Ejemplo, ¿Qué recursos tenemos para obtener datos?, ¿Qué información es relevante?, Limpiar y filtrar los datos para su posterior análisis.
  + **Etapa 3**: Explorar los datos. Ejemplo, Visualizar los datos, Localizar en los gráficos posibles tendencias, correlaciones o patrones.
  + **Etapa 4**: Modelizar y evaluar los datos. Ejemplo, Utilizar algún algoritmo innovador (según el problema) para crear el modelo, y Evaluar el modelo.
  + **Etapa 5**: Comunicar los resultados y/o puesta en producción. Ejemplo, ¿Qué resultados hemos obtenido?, ¿Qué hemos aprendido? Y ¿Los resultados tienen sentido?
* En cualquier etapa se puede avanzar y regresar.

**INTRODUCCIÓN AL ANÁLISIS PREDICTIVO Y AL MACHINE LEARNING**:

* **DataSet** son ficheros que contiene datos.
* El **análisis retrospectivo** permite analizar la historia, descubrir momentos claves, y encontrar fallos para evitar que no se repitan.
* El **análisis predictivo** es el enfoque que ayuda a darle una visión al dato, permite predecir el futuro.
* El **modelo predictivo** predice un resultado basado en parámetros de entrada, los cuales el modelo opera, predice o decide, de modo que en futuro tomar una decisión.
* La **estadística** es el eje fundamental sobre el cual se trabaja el modelo de decisión de aprendizaje estadístico.
* Los **algoritmos supervisados** son una serie de algoritmos donde los datos históricos tiene un valor de salida, aparte de los datos de entrada.
* Los **algoritmos no supervisados** son los que trabajan sin variables de salida precedente de los datos históricos, es decir, no necesito datos históricos para generar mi propio modelo.
* Entre las herramientas estadísticas tenemos a R y Python.
* Los datos históricos se dividirán en dos conjuntos, de entrenamiento (80%) y de validación (20%).
* Data Scientist = Matemáticas + Programación + Business:

**COMPUTER**

**SCIENCE**

**MATH &**

**STATISTICS**

**BUSINESS**

**ANALYSIS**

**MACHINE**

**LEARNING**

**DATA**

**SCIENTIST**

**DATA**

**ENGINEER**

**SOFTWARE**

**ENGINEER**

* Los modelos suelen elaborar una función matemática como resultado.
* Una empresa nunca sabe lo que quiere, solo sabe que quiere mejorar.
* Para un buen modelo predictivo las 4 grandes áreas a conocer son: Estadística, Algoritmos, Herramientas y técnicas, y el Contexto empresarial.
* Regla de Pareto indica que las cosas importantes se dividen en 80 – 20, 80% para la limpieza de datos y 20% para la modelización.
* Investigar:
  + Probabilidad.
  + Reglas de asociación.
  + Publicidad CTR.
  + Regresión logística.

**LIMPIEZA DE DATOS**:

* **DataSet** son los ficheros que utilizamos para realizar el análisis.
* Si en Jupyter no se activa el autocompletado con el tabulador, hay que colocar en la primera:
  + **%config IPCompleter.greedy=True**
* La ruta de los archivos en Windows se hace con “**/**”. Si no quiere cambiar la barra se puede colocar una **r** antes de la ruta ‘**r”c:\user…”**’.
* Hay que documentar cada DataSet, describiendo en general, cada columna y el separador.
* Para leer un fichero usamos **read\_csv** que pertenece a pandas:

Read\_csv(filepath=”[RUTA]”, sep=”[SEPARADOR]”, dtype={“NOMBRE\_1”:TIPO1, …, ”NOMBRE\_N”:TIPO\_N}, header=[NRO\_FILA], names={NUEVO\_NOMBRE\_1, …, NUEVO\_NOMBRE\_N}, skiprows=[FILA\_INICIO\_LECTURA], index\_col=[NUMERO\_COLUMNA], skip\_blank\_lines=[TRUE|FALSE] , na\_filter=[TRUE|FALSE])

Donde:

* + **filepath**: ruta donde se encuentra el fichero, puede ser absoluta o relativa.
  + **Sep**: el separador que tiene el fichero, por defecto considera que es una coma. También acepta expresiones regulares.
  + **Dtype**: permite formatear las columnas, indicando el nombre y su nuevo tipo de datos, ejemplo, una fecha que venga como cadena. Por defecto es None.
  + **Header**: permite especificar que fila es la cabecera, por defecto es la primera fila. Viene establecido en None.
  + **Name**: permite establecer los nombres de las columnas. Por defecto está en None.
  + **Skiprows**: permite saltar filas para comenzar con su lectora, ejemplo, valor 12 quiere decir que salta 12 filas y comienza en la 13. Por defecto está en None.
  + **index\_col**: permite indicar columnas que formen parte del identificador, por defecto es None.
  + **Skip\_blank\_lines**: valor booleano que permite ignorar las filas que tienen NaN, no colocara NaN sino valores en blanco. Por defecto está en False.
  + **Na\_filter**: elimina todas las filas que tienen valores desconocidos (NaN). Por defecto está en False.
  + **Encoding**: interpretación de caracteres especiales.

Ejemplo:





* Separar la ruta del fichero para no colocarla siempre:
  + La forma 1 genera 2 variables, mainpath contiene la ruta completa que no debería cambiar, y filename la parte de la ruta que si cambia. Luego se une en fullpath que se pasa al data.



* + La forma 2 utiliza la librería OS que une a través de la función path y su método join une las variables corrigiendo problemas si es que existen, por ejemplo espacios.



* Para conocer las columnas, es decir las cabeceras se hace:



* Para generar un DataFrame de 2 archivos, uno con los datos y otro con las cabeceras:

Texto

Descripción generada automáticamente

* Con read\_csv lo que hace panda es leer el dataset completo y si es muy grande se puede llenar la memoria del equipo. Por eso se puede usar la función open para generar el DataFrame.
* Si se quiere agregar información usar la opción “a” en vez de “w”, ya que la segunda elimina los datos y la primera posiciona el cursor al final.
* Para cargar datos a través de la función **Open**, se hace:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Para leer y escribir en un fichero:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Para leer desde una URL se hace:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Para leer ficheros XLS y XLSX se hace:

Imagen que contiene Texto

Descripción generada automáticamente

* Ejemplo: r.status debe ser 200, eso quiere decir que salió todo bien.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

* Generalizamos el ejercicio anterior en una función:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

* Funciones básicas de resumen, estructura, dimensiones y cabecera:

Texto, Correo electrónico

Descripción generada automáticamente

* ¿Por qué faltan valores en los dataset?







* Las variables dummy:
  + Una variable ficticia es una variable binaria que indica si una variable categórica separada toma un valor especifico.



* + Luego de generar las variables dummies, se elimina la columna original y se agregan las nuevas.



* + Para hacer todo de inmediato, crear una función.

Texto

Descripción generada automáticamente

* Visualización básica de un dataset

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* + Scatter Plot:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

* + Historiógrafa de frecuencias:

Gráfico, Histograma

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

* + Boxplot, diagrama de caja y bigotes:
    - Representa los 3 cuartiles, valores mínimos y máximo.
    - La parte baja de la caja es el percentil 25 (25%).
    - Al medio es el percentil 50 (50%).
    - La parte superior de la caja es el percentil 75 (75%).
    - Por debajo de la caja hay 1/4 de los datos.
    - Por encima de la caja hay 25% de los datos.
    - En la mitad sería la mediana.
    - El valor modal es el que más aparece.
    - El describe() se puede entender en este gráfico.

Texto

Descripción generada automáticamente

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

* + - La resta entre el primer cuartil y el tercer cuartil se conoce como rango intercuartílico (IQR) que es la altura de la caja.
    - Los bigotes se colocan a 1.4 veces el rango intercuartílico.
    - Si se pasan de los bigotes se pintan como bolitas.



**OPERACIONES DE MANEJO DE DATOS**

* Data Wrangling – cirugía de datos: a veces denominado **data munging**, es el proceso de transformar y mapear datos de un dataset raw (en bruto) en otro formato con la intención de hacerlo más apropiado y valioso para una variedad de propósitos posteriores, como el análisis. Un **Data Wrangling** es una persona que realiza estas operaciones de transformación. Esto puede incluir munging, visualización de datos, agregación de datos, entrenamiento de un modelo estadístico, así como muchos otros potenciales. La oscilación de datos como proceso generalmente sigue un conjunto de pasos generales que comienzan extrayendo los datos en forma cruda del origen de datos, dividiendo los datos en bruto usando algoritmos (por ejemplo, clasificación) o analizando los datos en estructuras de datos predefinidas, y finalmente depositando el contenido resultante en un sistema de almacenamiento (o silo) para su otro futuro.
* Los array de Python comienzan en el índice 0, es decir, si quiero 10 elementos, sería desde array[0:10], esto entregaría los valores de los índices del 0 al 9.
* Buscar un subconjunto de datos de un dataset: