




Finanzas en R


Proyecto del curso


Sebastián Egaña Santibáñez 

Nicolás Leiva Díaz 

Enlaces del profesor

 <https://sevana.netlify.app>

 <https://github.com/sebaegana>

 <https://www.linkedin.com/in/sebastian-egana-santibanez/>

Proyecto del curso

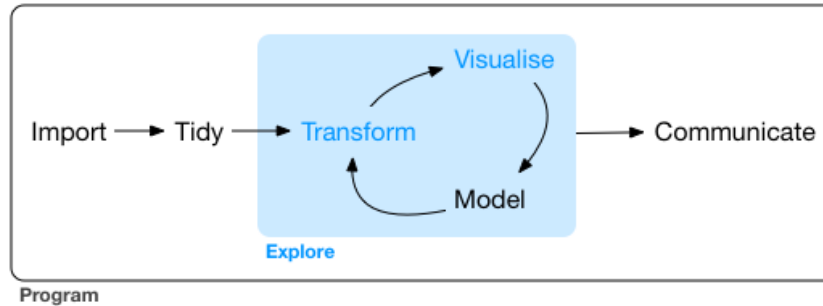
Alcance

Debe desarrollar un proyecto que solucione un problema de negocio en base a código y herramientas en R, ya sea real o hipotético. En cualquiera de las dos situaciones debe realizar una descripción detallada del problema.

Explicación

Como analista, lo que se proveen son soluciones a problemas de negocios, y no “el cálculo de” o “el dato de”. En este sentido, en el contexto de las organizaciones el trabajo debe estar orientado a proveer soluciones de negocio de largo plazo.

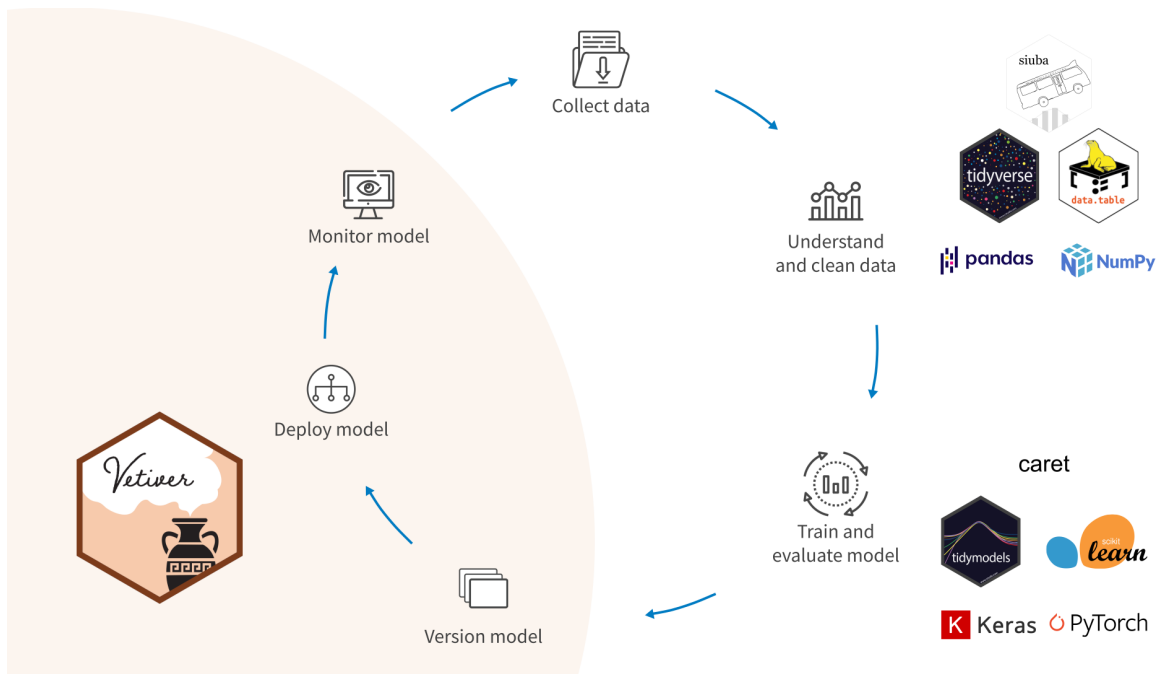
```
# All defaults  
img1_path <- "data-science-explore.png"  
knitr::include_graphics(img1_path)
```



En donde la comunicación corresponde a la solución al problema de negocios.

Más en el detalle, debemos tener la siguiente visión dentro de nuestros desarrollos:

```
# All defaults  
img2_path <- "ml_ops_cycle.png"  
knitr::include_graphics(img2_path)
```



Considere las siguientes explicaciones para el desarrollo de su modelo:

1. Collect data: corresponde a los procesos de extracción de datos, considerando toda las distintas fuentes utilizadas.
2. Understand and clean data: se relaciona con los procesos de limpieza y transformación de datos del proceso.

(Aparte): ambos procesos anteriores pueden considerar la unión de tablas.

3. Train and evaluate model: proceso relacionado con el entrenamiento del modelo y evaluación de métricas relevantes para el modelo.
4. Version model: en caso de existir versiones del modelo
5. Deploy model: despliegue y entrega del modelo
6. Monitor model: revisión constante de métricas del modelo, como también de la operatividad del mismo

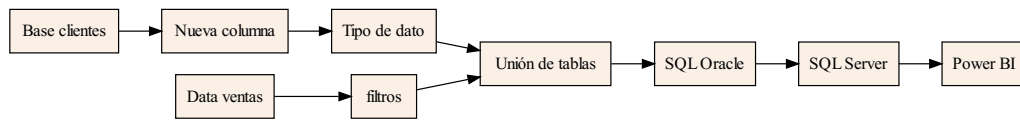
Veamos el siguiente ejemplo para el punto 1 y 2, para un modelo que no considera modelamiento avanzado:

```
DiagrammerR::grViz("digraph {
graph [layout = dot, rankdir = LR]
```

```
# define the global styles of the nodes. We can override these in box if we wish
node [shape = rectangle, style = filled, fillcolor = Linen]

rec1 [label = 'Base clientes']
rec2 [label = 'Nueva columna']
rec3 [label = 'Tipo de dato']
rec4 [label = 'Unión de tablas']
rec5 [label = 'SQL Oracle']
rec6 [label = 'Data ventas']
rec7 [label = 'filtros']
rec8 [label = 'SQL Server']
rec9 [label = 'Power BI']

# edge definitions with the node IDs
rec1 -> rec2 -> rec3 -> rec4 -> rec5 -> rec8 -> rec9
rec6 -> rec7 -> rec4
}"))
```



Entregables

Dentro de su trabajo, debe considerar los siguientes entregables:

1. Descripción de la solución: debe explicar qué soluciona, a quiénes impacta (clientes, usuarios, etc.), cómo espera solucionar el problema, periodicidad del desarrollo y alcance (que considera y que no considera)
2. Planificación del trabajo: considere el desarrollo de un pequeño descriptor, como también alguna herramientas para visualizar la planificación (carta Gantt)
3. MVP: desarrolle un producto mínimo viable para su solución. Esto incluye el código relacionado
4. Documentación: considere la necesidad de dejar constancia del trabajo desarrollado para su proyecto. Considere la utilización de alguna herramientas para generar diagramas como Bizagi o Visio. Como referencia se pueden utilizar las **Model card** de [Vetiver](#), las cuales deben ser complementadas con los apartados de limpieza y transformación de datos
5. Despliegue: describa cómo espera llevar a producción su solución
6. Monitoreo: describa cómo espera realizar el monitoreo de su modelo

Fechas de entrega

Parte 1

Considera los puntos 1 y 2 Fecha: 11 de septiembre

Parte 2

Considera el punto 3 Fecha: 28 de septiembre

Parte 3

Considera todos los puntos anteriores Fecha: 05 de octubre