**REPO Y CURSO INTRODUCTORIO UTILIZADO PARA APRENDER Y TOMAR IDEAS:**

REPO <https://github.com/marcopeix/MachineLearningModelDeploymentwithStreamlit>

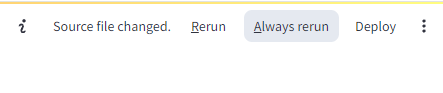
CURSO UDEMY:<https://www.udemy.com/course/machine-learning-model-deployment-with-streamlit/>

**COMO CORRER APP WEB DE STREAMLIT LOCALMENTE**

* pararse en la carpeta donde está el script con el código de streamlit
* abrir consola con el env que tiene instalado streamlit
* correr la siguiente linea de comando: **streamlit run your\_script.py**
* listo, automáticamente se debería abrir una pestaña en el navegador preteterminado en el local host: <http://localhost:8501/>

**RERUN DEL FRONT AL MOMENTO DE HACER UN CAMBIO EN EL CÓDIGO QUE LO GENERA Y GUARDAR ESE CAMBIO**

Cuando cambia el código se puede configurar stremalit para que siempre haga rerun de la app y guardar cambios en el script.py y así mostrar siempre la pantalla del código guardado



**CÓMO HACER DEPLOY DE LAS APPS – DOCU OFICIAL STREAMLIT**

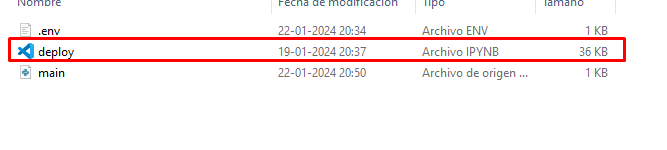
Link con lecturas de interés de cómo hacerlo en diferentes servicios de diferentes nubes

<https://docs.streamlit.io/knowledge-base/tutorials/deploy>

**CÓMO HACER DEPLOY EN CLOUD RUN – (seguir misma lógica que deploy de flask app en cloud run)**

Tanto para flask como para streamlit tengo desarrollado un código para hacer el deploy en un cloud run. Correr este notebook ghace todo el proceso, solo hay que definir los parámetros y luego correr los códigos e intermanente creará el dockerfile y el cloudbuild necesarios para hacer el deploy en un cloud run

Correr notebook deploy



**Fuentes:**

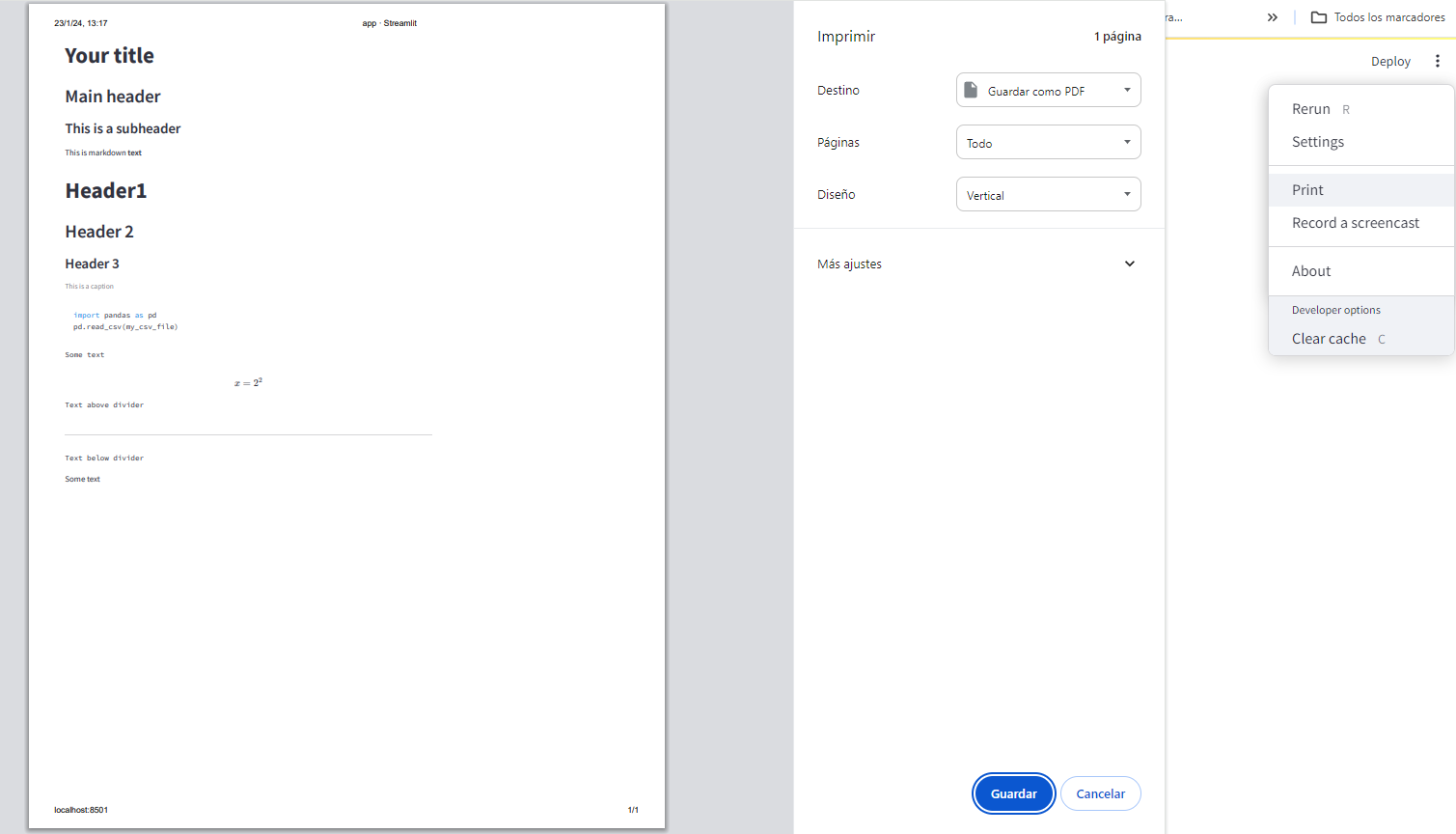
- deploy streamlit app in cloud run: https://medium.com/@faizififita1/how-to-deploy-your-streamlit-web-app-to-google-cloud-run-ba776487c5fe

- deploy streamlit app into google app engine: https://dev.to/whitphx/how-to-deploy-streamlit-apps-to-google-app-engine-407o

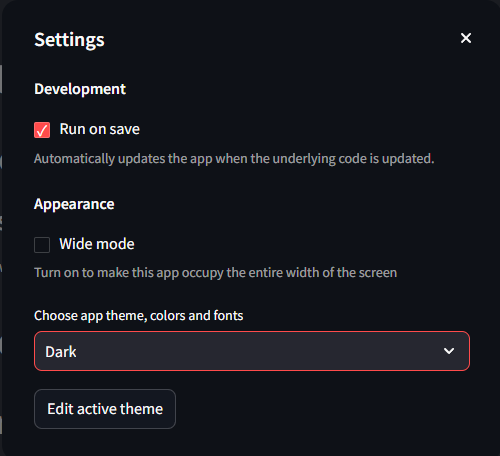
- deploy a flask app into a cloud run: my previous codes

**PERMITE IMPRIMIR LA PÁGINA WEB COMO UN PDF DE FORMA SIMPLE**

Entonces puedo diseñar una web de forma muy simple y además guardarla como un pdf (pareciera que sin perder las dimensiones) de forma extremadamente fácil



**OPCIONES DE CONFIGURACIÓN DE LA APP DESDE EL UI**

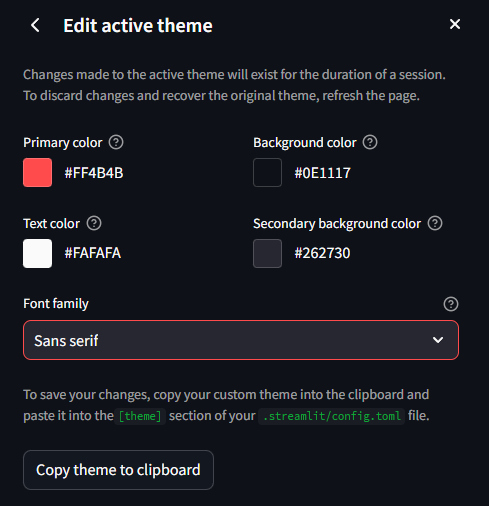


- development: run on save. Permite actualizar automáticamente la web al hacer un cambio en el código python que lo genera. En lugar de tener que cerrar el localhost y volver a abrirlo

- Opciones de apariencia:

- Permite ajustar la pantalla a que el texto ocupe toda la pantalla o esté centrado

- Permite migrar entre modo light y modo dark solo con un click!!. Además permite editar la tonalidad de los colores light y dark también solo cambiando los códigos de los colores desde la misma app



**RESUMEN:**

- cambiar temas light y dark con un solo click. Cambiar tonalidad de los colores del tema con un par de clicks

- tiene un nivel de customización suficiente para hacer una web para mostrar resultados y probar modelos de ml. Obviamente no tiene el nivel de customización que permite un desarrollo desde cero pero el tradeoff que ofrece entre tiempo de desarrollo (y curva de aprendizaje) y grado de personalización de la pantalla es muy aceptable para realizar pruebas rápidas y mostrar visualmente resultados

- con el nivel de personalización que ofrece es más que suficiente para mostrar la inferencia de modelos de ml y a la vez para que el usuario haga pruebas de la inferencia de los modelos

**FALTA VER:**

- CONEXIÓN A CLOUD

- REFRESH AUTOMÁTICO DE LA PANTALLA

- REFRESH AUTOMÁTICO DE LA PANTALLA CON UN OBSERVER DESDE BIGQUERY

- CUANTO DURA LA APP SIN CAERSE DESDE:

- CLOUD RUN (ME IMAGINO QUE POCO :/ )

- CORRIENDO EN EL LOCALHOST

- CORRIENDO EN EL LOCAL EN ALGUNA MAQUINA VIRTUAL O ALGO . APACHE/HAMACHI/O ALGO QUE SE UTILIZÓ EN EL RAMO DE TI

- VER LOS COSTOS DE TENER LA APP EN UN COMPUTE ENGINE U OTRA FORMA DE DESPLIEGUE QUE HAGA QUE NO SE CAIGA LA APP