



U N I V E R S I D A D
Panamericana

Alumno: Adriana Leticia Martinez Estrada

Trabajo: Proyecto final Github Pull Requests

Materia: Grandes Datos

Profesor: Jaime Ulises Jiménez Cardoso

Fecha de entrega: 6 de marzo de 2024

Resumen Ejecutivo	4
Desarrollo de la Solución	4
Aplicación del Modelo	5
Beneficios y Clientela Potencial	5
Visión General del Desarrollo	5
Solución Actual	5
Limitaciones de la Solución	5
Propósito, Uso y Alcances	5
Revisión y uso de datos	6
Orígenes de Datos y Control de Datos	6
Preparación de Datos	6
Limpieza y Tratamiento de Datos	6
Integridad de los Datos	6
Limitaciones de Datos	7
Proceso de Desarrollo	7
Metodología	7
Pruebas	7
Resultados y Conclusiones	8
Resultados Obtenidos	8
Resultados del Modelo de Machine Learning	8
Prueba datos Testeo	9
Herramientas y Beneficios	9
Conclusiones Clave	10
Trabajo Futuro	10
Referencias	11
Anexos	12
Querys	12
Creación de Tabla externa HIVE	12
Creación de tabla Auxiliar	12
Exportación de información de HIVE a Bucket	12
Creación De Vista en BigQuery	13
Creación de tabla auxiliar para segmentación de información en BigQuery	13
Generación de tablas de entrenamiento y pruebas	13
Entrenamiento de Modelo BigQueryML	14
Evaluación de Modelo BigQueryML	14
Comandos configuración de Herramientas	14
Máquina Virtual	14
Conexión máquina local a GCP	15
Transferencia de claves Kaggle	15
Creación Bucket	15
Descarga de información a través del API de Kaggle	15
Transferencia de indormación de VM a Bucket	15
Creación de Cluster	15

Carga de Información filtrada de Bucket a BigQuery	16
Exportación de modelo	16
Descarga los archivos del modelo exportado en un directorio temporal	16
Creación de un subdirectorio de la versión	16
Extracción de la imagen de Docker	16
Ejecución del contenedor de Docker	16
Ejecución de la predicción	16
Repositorio Remoto	16
Google Site GitHub Pull Requests	17
Diccionarios de datos	17
Tabla original	17
Tabla para modelo	18
Evidencias	19
Almacenamiento en Bucket	19
BigQuery	20
Visualizaciones Looker	22

GitHub Pull Requests

Resumen Ejecutivo

En esta era digital, el desarrollo de software no es solo una tarea aislada de programadores individuales; se ha transformado en una vasta red de colaboración global. Plataformas como GitHub han emergido como epicentros de este paradigma, permitiendo a desarrolladores de todo el mundo contribuir a proyectos comunes, compartir código y fomentar la innovación. GitHub, un repositorio de código y una plataforma que facilita el versionado y la colaboración en proyectos de software, ofreciendo herramientas esenciales para el manejo eficaz de proyectos complejos.

Uno de los conceptos fundamentales en GitHub es el "pull request" (PR), que es una solicitud enviada por contribuyentes para que los cambios que han implementado en su versión del proyecto sean revisados y potencialmente integrados (merge) en la base de código principal. Los PRs son el corazón de la colaboración en GitHub, permitiendo que el código sea discutido, revisado y mejorado antes de su incorporación final, asegurando así la calidad y la coherencia del software.[1]

Este proyecto se centra en el análisis de Pull Requests de GitHub para explorar cómo las contribuciones individuales afectan la calidad general del desarrollo de software. Con el uso de un conjunto de datos extenso, superior a 20 GB, proveniente de GHTorrent —una base de datos que indexa la actividad pública de GitHub—, buscamos identificar patrones y tendencias en las contribuciones que podrían indicar la calidad de los aportes.

La meta es aplicar técnicas de Machine Learning para desarrollar un modelo capaz de predecir la presencia de errores en los PRs. Este modelo no solo podría ser una herramienta invaluable para los mantenedores de proyectos, facilitando la revisión y aceptación de contribuciones, sino que también ofrecería insights sobre prácticas de desarrollo que maximizan la calidad del código.

Desarrollo de la Solución

Utilizando Google Cloud Storage, Dataproc y BigQuery, se extrajeron y transformaron datos significativos de cinco archivos con un peso total de 69 GB, aplicando un proceso ETL detallado para preparar un conjunto de datos para el análisis y la modelación. Las herramientas de Big Data facilitaron este proceso, permitiendo la manipulación eficiente de un gran volumen de información.

Aplicación del Modelo

Se implementó un modelo en BigQuery ML, que, basado en expresiones regulares, identifica y clasifica los comentarios de los commits por la presencia de términos clave. Esta clasificación ayudará a predecir si un pull request contiene errores, lo que puede interpretarse como un indicador de la necesidad de mejora.

Beneficios y Clientela Potencial

El proyecto no solo aporta a la mejora del proceso de revisión de código, sino que también beneficia a los gestores de repositorios y líderes de proyecto al proporcionar un método sistemático para evaluar la calidad de las contribuciones así como la cantidad de las mismas. Los equipos de desarrollo que buscan optimizar sus flujos de trabajo de revisión y contribución son también clientes ideales de este producto.

Visión General del Desarrollo

Solución Actual

Se aborda el desafío de analizar y clasificar las contribuciones a repositorios en GitHub mediante Pull Requests (PRs), utilizando un conjunto de datos de gran volumen (aproximadamente 69 GB). A través de un proceso de ETL (Extracción, Transformación y Carga), hemos procesado y almacenado con éxito estos datos en Google Cloud Storage. Utilizando las capacidades de procesamiento de Dataproc y la consulta de datos de BigQuery, hemos creado vistas y tablas para la segmentación de los datos, y hemos desarrollado un modelo predictivo con BigQuery ML para evaluar la calidad de los PRs basándonos en comentarios y otros metadatos relevantes.

Limitaciones de la Solución

La solución depende de la precisión de las expresiones regulares para identificar comentarios significativos dentro de los PRs y de esta forma generar una categorización inicial. La identificación de errores, buen trabajo y nuevas características está sujeta a las limitaciones inherentes al análisis de texto simple y puede no capturar la complejidad o el contexto completo de las conversaciones en GitHub.

Propósito, Uso y Alcances

El propósito de esta herramienta es proporcionar a los gestores de repositorios y a los equipos de desarrollo insights prácticos y accionables para mejorar la colaboración y la calidad del código en proyectos de software. Los indicadores clave y el modelo predictivo buscan identificar patrones de éxito y áreas de mejora en las contribuciones de código, con el objetivo de incrementar la eficiencia y efectividad de la revisión y aceptación de PRs. El modelo y las visualizaciones resultantes ofrecen una perspectiva valiosa que puede mejorar

las prácticas de desarrollo de software en comunidades de código abierto y en entornos profesionales.

La herramienta tiene un alcance amplio, ofreciendo beneficios a pequeños proyectos independientes así como a grandes organizaciones que gestionan múltiples contribuciones en diversas bases de código. Con su API expuesta, la herramienta puede facilitar la integración en flujos de trabajo existentes.

Revisión y uso de datos

Orígenes de Datos y Control de Datos

La base de datos sobre la que se ha trabajado fue extraída de Kaggle, específicamente del conjunto de datos de GitHub Pull Requests provisto por GHTorrent. Esta extracción se llevó a cabo a través de una máquina virtual configurada en Google Cloud Compute Engine, conectada a la API de Kaggle, donde se ejecutó el comando de descarga y se procesaron los archivos correspondientes.

Preparación de Datos

Los datos, una vez extraídos y descomprimidos en la máquina virtual, fueron almacenados en un Bucket de Google Cloud Storage, designado para su almacenamiento intermedio y accesibilidad. Esta etapa fue esencial para preparar el terreno para el procesamiento y análisis subsiguiente.

Limpieza y Tratamiento de Datos

Utilizando Dataproc y Hive, los datos se sometieron a un riguroso proceso de ETL. Se crearon tablas externas que permitieron la agrupación y organización de los distintos archivos del dataset. Se realizó una limpieza de datos donde se filtraron aquellos registros no esenciales para el análisis, como pull requests sin comentarios, sin repositorio específico o sin lenguaje de programación definido.

Integridad de los Datos

La integridad de los datos se mantuvo mediante consultas de verificación en Hive, asegurándose de que solo los datos válidos y relevantes fueran procesados y analizados. Esto fue vital para garantizar que la información que avanzaba a través del pipeline fuera precisa y confiable.

Limitaciones de Datos

Las limitaciones en el conjunto de datos se abordaron desde el principio. Al centrarnos solo en los registros que proporcionaban información completa, se excluyeron aquellos que podrían introducir ruido o sesgo en el análisis. No obstante, las limitaciones en términos de la variedad y la complejidad de los comentarios en el texto de los commits podrían influir en la capacidad del modelo para generalizar y captar la sutileza de los comentarios humanos.

Proceso de Desarrollo

Metodología

Esta solución se abordó mediante una metodología iterativa, con la flexibilidad de las herramientas de GCP como eje central. La solución implicó el uso de la API de Kaggle para extraer un conjunto de datos significativo de GitHub Pull Requests, proporcionados por GHTorrent y disponibles en Kaggle. Una máquina virtual en GCP sirvió como el punto de partida para la descarga y descompresión de datos, que luego fueron transferidos a Google Cloud Storage. Esta secuencia de acciones no solo evidenció la capacidad de las herramientas de GCP para manejar grandes volúmenes de datos, sino que también mostró la interoperabilidad entre diversos servicios en la nube.

El cluster en Dataproc se utilizó para el procesamiento ETL, aprovechando Hive para realizar la limpieza y organización de los datos, preparándolos para su análisis posterior. Con la data ya procesada, se cargó a BigQuery para realizar análisis más detallados y para alimentar las visualizaciones en Looker Studio.

Una vez en BigQuery, se crearon vistas y tablas con la información necesaria para el entrenamiento y prueba de un modelo predictivo de regresión lineal, utilizando BigQuery ML para evaluar la probabilidad de errores en los commits basándose en los comentarios de los pull requests. El enfoque no solo se limitó al análisis de datos, sino que también se extendió al despliegue de un modelo de Machine Learning a través de un API.

El API fue desarrollado para funcionar inicialmente en un entorno local, permitiendo la ejecución de predicciones y la integración de los resultados en un sitio web creado con Google Sites, haciendo los resultados accesibles y fácilmente interpretables por los usuarios finales. El sitio web actúa como una interfaz de usuario amigable que demuestra el valor práctico del proyecto.

Pruebas

Las pruebas jugaron un papel fundamental en el proyecto. Se emplearon para validar cada paso del proceso ETL, garantizar la integridad de los datos en BigQuery, y confirmar la eficacia del modelo de ML. La precisión del modelo, su recuerdo y su puntuación F1 se calculó utilizando un conjunto de datos de prueba para evaluar su rendimiento y afinar sus parámetros.

Las pruebas no solo se limitaron a la validación de modelos, sino que también se extendieron a la integración del API y su funcionalidad dentro del sitio web. Se realizaron pruebas para verificar la respuesta del API a diversas solicitudes y asegurar que los resultados fueran consistentes y confiables.

Resultados y Conclusiones

Resultados Obtenidos

La implementación de la solución siguió un flujo de trabajo claro y estructurado. Utilizamos Compite engine para la descarga y descompresión de la información, Google Cloud Storage para el almacenamiento inicial de datos así como una herramienta auxiliar para almacenar la información y poder importarla entre las distintas herramientas utilizadas, Dataproc y Hive para el procesamiento y la limpieza de datos, y BigQuery/BigQueryML para el análisis de datos y la creación de modelos predictivos. La visualización de datos se llevó a cabo con Looker Studio, proporcionando paneles intuitivos que resumen nuestras métricas clave y conclusiones. través de un Site de Google para brindar una funcionalidad práctica a esta solución.

A través de este enfoque, pudimos no solo identificar y clasificar comentarios relevantes dentro de los PRs, sino también desarrollar un modelo de Machine Learning que predice la presencia de errores con una precisión, recall, y puntuaciones F1 y AUC significativas. Este modelo se desplegó con las herramientas de AI Platform y Google APIs utilizando un API desarrollado que funcionó inicialmente en un entorno local, y los resultados se hicieron accesibles a través de un sitio web en Google Sites, demostrando la aplicabilidad práctica de nuestra investigación.

Resultados del Modelo de Machine Learning

Esta solución ha culminado en el desarrollo de un modelo de clasificación que demuestra capacidades prometedoras en predecir errores en pull requests de GitHub. La evaluación del modelo reveló una precisión del 79.17%, lo que indica una alta confiabilidad en las predicciones positivas. A pesar de una baja recuperación, lo que sugiere un margen para detectar todos los errores potenciales, el modelo demostró una capacidad significativa para identificar los pull requests más críticos que requieren atención. La pérdida logística mostró una disminución consistente a lo largo de las iteraciones de entrenamiento, señalando un buen ajuste del modelo.

La curva ROC, con un área de 0.6998, y la matriz de confusión respaldan la eficacia del modelo. Estos resultados validan el enfoque analítico adoptado y sientan las bases para aplicaciones más amplias de la tecnología de aprendizaje automático en la mejora del proceso de revisión de software. La tasa de aprendizaje se incrementó progresivamente para permitir que el modelo se adaptara de manera efectiva a la complejidad de los datos. La duración de las iteraciones refleja una optimización eficiente de los recursos computacionales.

La implementación del modelo ha aportado valiosos insights sobre la contribución de las variables a la presencia de errores. Esta información puede guiar a los desarrolladores y a los equipos de proyectos para refinar sus prácticas de código y mejorar la calidad general del software. Además, las visualizaciones desarrolladas ofrecen una manera accesible de comprender el desempeño del modelo y las tendencias de los datos.

Prueba datos Testeo

Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.780698	0.02361515386	0.9252535530	0.04584359546	0.2452104565	0.7105834165

Los resultados de la prueba de testeo del modelo con datos de prueba han reforzado la eficacia de nuestra solución. Alcanzamos una precisión de 0.7807, lo que subraya la habilidad del modelo para identificar correctamente los errores con gran exactitud. Aunque el recall de 0.0236 indica que hay espacio para mejorar en la identificación de todos los casos positivos reales, la exactitud general del modelo fue notable, con un valor de 0.9253.

El f1_score de 0.0458, aunque mejorable, refleja el difícil equilibrio entre precisión y recuperación para nuestro conjunto de datos desafiante. La pérdida logística de 0.2452 demuestra la eficiencia del modelo en términos de las probabilidades pronosticadas, mientras que un área ROC de 0.7106 valida la habilidad del modelo para distinguir entre clases de manera efectiva.

Estos resultados enfatizan la promesa de esta metodología y la aplicabilidad del modelo en entornos reales, abriendo caminos para futuras optimizaciones y aplicaciones prácticas.

Herramientas y Beneficios

Las herramientas seleccionadas para este proyecto ofrecieron beneficios específicos:

- **Compute Engine:** Proporcionó un entorno virtual y en la nube que nos permitió la conectividad con APIs externas para la descarga y descompresión de archivos sin necesidad de una descarga local.
- **Google Cloud Storage:** Ofreció una solución escalable para manejar grandes conjuntos de datos, facilitando el acceso y la colaboración.
- **Dataproc y Hive:** Permitieron el procesamiento eficiente de datos estructurados y semi-estructurados a gran escala.
- **BigQuery y BigQueryML:** Proporcionaron una plataforma poderosa para análisis y modelado predictivo con capacidades de Machine Learning integradas.
- **Looker Studio:** Permitió la visualización interactiva y dinámica de datos para interpretar fácilmente las conclusiones del proyecto.
- **AI Platform y Google APIs (No visto en Clase):** Permitieron la productivización y despliegue del modelo desarrollado

Conclusiones Clave

Esta solución destacó la relevancia del análisis de datos para mejorar la calidad y eficiencia del desarrollo de software. La clasificación automática de PRs por calidad puede servir como una herramienta valiosa para los desarrolladores y equipos de proyectos de software, ayudándoles a identificar áreas de mejora y asegurando mejores prácticas en el desarrollo colaborativo.

Trabajo Futuro

Mirando hacia el futuro, reconocemos que el modelo podría mejorarse aún más con la integración de datos adicionales, como la aceptación de los PRs y el número de revisiones que reciben. Tales métricas podrían ofrecer una comprensión más profunda de la productividad y la calidad en el desarrollo de software. En vez de centrarse únicamente en la presencia de comentarios de errores para medir la productividad, estas métricas adicionales podrían proporcionar una visión más holística del proceso de desarrollo y permitir la creación de un modelo más refinado y preciso.

La incorporación de estas métricas en el conjunto de datos y su análisis a través de nuestro modelo podría resultar en una herramienta más robusta para predecir la calidad de los PRs, lo que a su vez podría traducirse en una colaboración más efectiva y en una mayor calidad del código producido en proyectos de código abierto y privados por igual. Con el objetivo de mejorar constantemente, planeamos recopilar estos datos adicionales y actualizar nuestro modelo y visualizaciones en consecuencia, asegurando que este trabajo siga siendo relevante y de vanguardia en el campo de la ciencia de datos.

Referencias

- [1] Olmedo, A., Arévalo, G., Cassol, I., Perez, Q., Urtado, C., & Vauttier, S. (2022, April). Pull Requests Integration Process Optimization: An Empirical Study. In *International Conference on Evaluation of Novel Approaches to Software Engineering* (pp. 155-178). Cham: Springer Nature Switzerland.
- [2]<https://towardsdatascience.com/quickly-transfer-a-kaggle-dataset-into-a-google-bucket-ac21aefceb15>
- [3] <https://cloud.google.com/sdk/docs/install?hl=es-419>
- [4] <https://cloud.google.com/compute/docs/instances/transfer-files?hl=es-419>
- [5] <https://cloud.google.com/bigquery/docs/export-model-tutorial?hl=es-419>
- [6] <https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create>
- [7]<https://cloud.google.com/bigquery/docs/create-machine-learning-model?hl=es-419>
- [8] NUNES, T. A. R. (2019). Git dashboard: dashboard de pull requests para o github.

Anexos

Querys

Creación de Tabla externa HIVE

```
CREATE EXTERNAL TABLE IF NOT EXISTS github_pull_requests (
  actor_login STRING,
  actor_id INT,
  comment_id INT,
  comment STRING,
  repo STRING,
  language STRING,
  author_login STRING,
  author_id INT,
  pr_id INT,
  c_id INT,
  commit_date STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'gs://up-pf-ghtorrent-2024-alme/temp_zip_extract';
```

Creación de tabla Auxiliar

```
CREATE TABLE github_pull_requests_clean AS
SELECT actor_login, actor_id, comment_id, comment, repo, language, author_login,
author_id, pr_id, c_id, commit_date
FROM github_pull_requests
WHERE repo IS NOT NULL
AND language IS NOT NULL
AND comment IS NOT NULL
AND TRIM(comment) <> ''
AND comment NOT LIKE '%NULL%'
AND commit_date IS NOT NULL
AND LENGTH(commit_date) = 23;
```

Exportación de información de HIVE a Bucket

```
INSERT OVERWRITE DIRECTORY
'gs://up-pf-ghtorrent-2024-alme/temp_for_bigquery_load/'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
SELECT * FROM github_pull_requests_clean;
```

Creación De Vista en BigQuery

```
CREATE VIEW `bd-final-alme.pf_github.pruebagit_with_flags` AS
SELECT
  actor_login,
  actor_id,
  comment_id,
  comment,
  repo,
  language,
  author_login,
  author_id,
  pr_id,
  c_id,
  commit_date,
  CASE WHEN LOWER(comment) LIKE '%error%' OR LOWER(comment) LIKE '%fix%' OR
  LOWER(comment) LIKE '%bug%' THEN TRUE ELSE FALSE END AS contains_error,
  CASE WHEN LOWER(comment) LIKE '%well%' OR LOWER(comment) LIKE '%good%'
  OR LOWER(comment) LIKE '%great%' OR LOWER(comment) LIKE '%ok%' THEN TRUE
  ELSE FALSE END AS good_job,
  CASE WHEN LOWER(comment) LIKE '%new%' OR LOWER(comment) LIKE '%feature%'
  THEN TRUE ELSE FALSE END AS contains_new_feature
FROM
  `bd-final-alme.pf_github.pruebagit`;
```

Creación de tabla auxiliar para segmentación de información en BigQuery

```
CREATE TABLE `bd-final-alme.pf_github.pruebagit-segment`
AS
SELECT
  actor_login,
  language,
  repo,
  CASE WHEN LOWER(comment) LIKE '%error%' OR LOWER(comment) LIKE '%fix%' OR
  LOWER(comment) LIKE '%bug%' THEN TRUE ELSE FALSE END AS contains_error,
  CASE WHEN LOWER(comment) LIKE '%well%' OR LOWER(comment) LIKE '%good%'
  OR LOWER(comment) LIKE '%great%' OR LOWER(comment) LIKE '%ok%' THEN TRUE
  ELSE FALSE END AS good_job,
  RAND() AS random_number
FROM `bd-final-alme.pf_github.pruebagit`;
```

Generación de tablas de entrenamiento y pruebas

```
CREATE OR REPLACE TABLE `bd-final-alme.pf_github.tabla_entrenamiento` AS
SELECT *
```

```
FROM `bd-final-alme.pf_github.pruebagit-segment`
WHERE random_number < 0.8;
```

```
CREATE OR REPLACE TABLE `bd-final-alme.pf_github.tabla_pruebas` AS
SELECT *
FROM `bd-final-alme.pf_github.pruebagit-segment`
WHERE random_number >= 0.8;
```

Entrenamiento de Modelo BigQueryML

```
CREATE OR REPLACE MODEL `bd-final-alme.pf_github.modelo_contains_error`
OPTIONS(model_type='logistic_reg', input_label_cols=['contains_error']) AS
SELECT
  actor_login,
  language,
  IF(contains_error, 1, 0) AS contains_error
FROM
  `bd-final-alme.pf_github.tabla_entrenamiento`;
```

Evaluación de Modelo BigQueryML

```
SELECT
  *
FROM
  ML.EVALUATE(MODEL `bd-final-alme.pf_github.modelo_contains_error`, (
    SELECT
      actor_login,
      language,
      IF(contains_error, 1, 0) AS contains_error
    FROM
      `bd-final-alme.pf_github.tabla_pruebas`
  ));
```

Comandos configuración de Herramientas

Máquina Virtual

```
gcloud compute instances create vm-pf-kaggle-alme --project=bd-final-alme
--zone=us-central1-c --machine-type=n2-standard-4
--network-interface=network-tier=PREMIUM,stack-type=IPV4_ONLY,subnet=default
--maintenance-policy=MIGRATE --provisioning-model=STANDARD
--service-account=594778691467-compute@developer.gserviceaccount.com
```

```
--scopes=https://www.googleapis.com/auth/cloud-platform --tags=http-server,https-server
--create-disk=auto-delete=yes,boot=yes,device-name=vm-pf-kaggle-alme,image=projects/ubuntu-os-cloud/global/images/ubuntu-2004-focal-v20240229,mode=rw,size=200,type=projects/bd-final-alme/zones/us-central1-c/diskTypes/pd-balanced --no-shielded-secure-boot
--shielded-vtpm --shielded-integrity-monitoring --labels=goog-ec-src=vm_add-gcloud
--reservation-affinity=any
```

Conexión máquina local a GCP

```
gcloud compute ssh --project bd-final-alme --zone us-central1-c vm-pf-kaggle-alme
```

Transferencia de claves Kaggle

```
gcloud compute scp Downloads/kaggle.json vm-pf-kaggle-alme:~
```

Creación Bucket

```
gsutil mb gs://up-pf-ghtorrent-2024-alme
```

Descarga de información a través del API de Kaggle

```
gcloud compute scp Downloads/kaggle.json vm-pf-kaggle-alme:~
```

Transferencia de información de VM a Bucket

```
gsutil -m cp -r temp_zip_extract gs://up-pf-ghtorrent-2024-alme
```

Creación de Cluster

```
gcloud dataproc clusters create cluster-pf-alme \
  --enable-component-gateway \
  --region us-central1 \
  --zone us-central1-c \
  --master-machine-type n2-standard-2 \
  --master-boot-disk-size 200 \
  --num-workers 2 \
  --worker-machine-type n2-standard-2 \
  --worker-boot-disk-size 200 \
  --image-version 2.0-ubuntu18 \
  --optional-components JUPYTER \
  --bucket up-pf-ghtorrent-2024-alme \
  --project bd-final-alme
```

Carga de Información filtrada de Bucket a BigQuery

```
bq load --source_format=CSV --autodetect \
pf_github.pull_requests_clean \
gs://up-pf-ghtorrent-2024-alme/temp_for_bigquery_load/*
```

Exportación de modelo

```
bq extract -m pf_github.modelo_contains_error gs://up-pf-ghtorrent-2024-alme/pred_model
```

Descarga los archivos del modelo exportado en un directorio temporal

```
mkdir tmp_dir
gsutil cp -r gs://up-pf-ghtorrent-2024-alme/pred_model tmp_dir
```

Creación de un subdirectorio de la versión

```
mkdir -p serving_dir/pred_model/1
cp -r tmp_dir/pred_model/* serving_dir/pred_model/1
rm -r tmp_dir
```

Extracción de la imagen de Docker

```
docker pull tensorflow/serving
```

Ejecución del contenedor de Docker

```
docker run -p 8500:8500 --network="host" --mount
type=bind,source=`pwd`/serving_dir/pred_model,target=/models/pred_model -e
MODEL_NAME=pred_model -t tensorflow/serving &
```

Ejecución de la predicción

```
curl -d '{"instances": [{"actor_login": "adrianaleticiamartinez", "language": "PHP"}]}' -X POST
http://localhost:8501/v1/models/pred\_model:predict
```

Despliegue Online

Creación de modelo

```
MODEL_NAME="PRED_MODEL"
gcloud ai-platform models create $MODEL_NAME
```

Creación de versión

```
MODEL_DIR="gs://up-pf-ghtorrent-2024-alme/pred_model"
VERSION_NAME="v1"
FRAMEWORK="TENSORFLOW"
gcloud ai-platform versions create $VERSION_NAME --model=$MODEL_NAME
--origin=$MODEL_DIR --runtime-version=1.15 --framework=$FRAMEWORK
--region=us-central1 --machine-type=n1-standard-2
```


Repositorio Remoto

Se incluye en un repositorio remoto con acceso público el código, comandos y Querys utilizados

https://github.com/adrianaleticiamartinez/mcd_bigdata

Google Site GitHub Pull Requests

Se incluye en un repositorio remoto con acceso público el código, comandos y Querys utilizados

<https://sites.google.com/up.edu.mx/githubpullrequests>

Diccionarios de datos

Tabla original

Campo	Tipo de Datos	Descripción
actor_login	STRING	El nombre de usuario del actor que realizó la acción en el Pull Request.
actor_id	INTEGER	El ID del actor que realizó la acción en el Pull Request.
comment_id	INTEGER	El ID único asociado al comentario en el Pull Request.
comment	STRING	El contenido del comentario asociado al Pull Request.
repo	STRING	El nombre del repositorio asociado al Pull Request.
language	STRING	El lenguaje de programación asociado al repositorio.
author_login	STRING	El nombre de usuario del autor del Pull Request.
author_id	INTEGER	El ID del autor del Pull Request.
pr_id	INTEGER	El ID único asociado al Pull Request.
c_id	INTEGER	El ID único asociado al commit (compromiso) relacionado con el Pull Request.
commit_date	TIMESTAMP	La fecha y hora en que se realizó el commit relacionado con el Pull Request.

Tabla para modelo

Campo	Tipo de Datos	Descripción
actor_login	STRING	El nombre de usuario del actor que realizó la acción en el Pull Request.
language	STRING	El lenguaje de programación asociado al repositorio.
repo	STRING	El nombre del repositorio asociado al Pull Request.
contains_error	BOOLEAN	Indica si el comentario asociado al Pull Request contiene palabras clave relacionadas con errores (<code>error</code> , <code>fix</code> , <code>bug</code>).
good_job	BOOLEAN	Indica si el comentario asociado al Pull Request contiene palabras clave que reconocen el buen trabajo realizado (<code>well</code> , <code>good</code> , <code>great</code> , <code>ok</code>).
random_number	FLOAT	Un número aleatorio generado con el fin de segmentar los datos entre los conjuntos de entrenamiento y prueba.

Evidencias

Almacenamiento en Bucket

Archivos descompresión

up-pf-ghtorrent-2024-ahme

Ubicación

us (varias regiones en Estados Unidos)

Clase de almacenamiento

Standard

Acceso público

Sujeto a LCA de objeto

Protección

Ninguna

OBJETOS

CONFIGURACIÓN

PERMISOS

PROTECCIÓN

CICLO DE VIDA

OBSERVABILIDAD

INFORMES DE INVENTARIO

Depósitos > up-pf-ghtorrent-2024-ahme > temp_zip_extract

SUBIR ARCHIVOS

SUBIR CARPETA

CREAR CARPETA

TRANSFERIR LOS DATOS

ADMINISTRAR CONSERVACIONES

EDITAR LA RETENCIÓN

DESCARGAR

BORRAR

Filtrar solo por prefijo de nombre

Filtro Filtrar objetos y carpetas

Mostrar datos borrados

<input type="checkbox"/>	Nombre	Tamaño	Tipo	Fecha de creación	Clase de almacenamiento	Última modificación	Acceso público	Historial de versiones	Encript
<input type="checkbox"/>	ghtorrent-2019-01-07.csv	16.4 GB	text/csv	4 mar 2024 13:53:44	Standard	4 mar 2024 13:53:44	No público	—	Admin
<input type="checkbox"/>	ghtorrent-2019-02-04.csv	18.5 GB	text/csv	4 mar 2024 13:54:12	Standard	4 mar 2024 13:54:12	No público	—	Admin
<input type="checkbox"/>	ghtorrent-2019-03-11.csv	18.5 GB	text/csv	4 mar 2024 13:54:12	Standard	4 mar 2024 13:54:12	No público	—	Admin
<input type="checkbox"/>	ghtorrent-2019-04-15.csv	16.7 GB	text/csv	4 mar 2024 13:53:51	Standard	4 mar 2024 13:53:51	No público	—	Admin

Archivos para carga en BigQuery

← Detalles del bucket

ACTUALIZAR

APRENDIZAJE

us (varias regiones en Estados Unidos)

Standard

Sujeto a LCA de objeto

Ninguna

OBJETOS

CONFIGURACIÓN

PERMISOS

PROTECCIÓN

CICLO DE VIDA

OBSERVABILIDAD

INFORMES DE INVENTARIO

Depósitos > up-pf-ghtorrent-2024-ahme > temp_for_bigquery_load

SUBIR ARCHIVOS

SUBIR CARPETA

CREAR CARPETA

TRANSFERIR LOS DATOS

ADMINISTRAR CONSERVACIONES

EDITAR LA RETENCIÓN

DESCARGAR

BORRAR

Filtrar solo por prefijo de nombre

Filtro Filtrar objetos y carpetas

Mostrar datos borrados

<input type="checkbox"/>	Nombre	Tamaño	Tipo	Fecha de creación	Clase de almacenamiento	Última modificación	Acceso público	Historial de versiones
<input type="checkbox"/>	000000_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000001_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000002_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000003_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000004_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000005_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000006_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000007_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000008_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000009_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000010_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000011_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000012_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000013_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000014_0	896 MB	application/octet-stream	4 mar 2024 19:48:44	Standard	4 mar 2024 19:48:44	No público	—
<input type="checkbox"/>	000015_0	896 MB	application/octet-stream	4 mar 2024 19:48:45	Standard	4 mar 2024 19:48:45	No público	—
<input type="checkbox"/>	000016_0	896 MB	application/octet-stream	4 mar 2024 19:48:45	Standard	4 mar 2024 19:48:45	No público	—

[illegible]

Sets de entrenamiento y prueba

pruebagit-segment

CONSULTA

COMPARTIR

COPIAR

INSTANTÁNEA

BORRAR

EXPORTAR

ACTUALIZAR

ESQUEMA

DETALLES

VISTA PREVIA

LINAJE

PERFIL DE DATOS

CALIDAD DE LOS DATOS

Fila	actor_login	language	repo	contain	good_jc	random_number
1	seanh	Python	h	false	false	0.77453025...
2	seanh	Python	h	true	false	0.68930523...
3	seanh	Python	h	false	false	0.69670720...
4	seanh	Python	h	false	false	0.16725925...
5	seanh	Python	h	false	false	0.17146121...
6	seanh	Python	h	true	false	0.32505893...
7	seanh	Python	h	false	false	0.99518137...
8	seanh	Python	h	false	false	0.35985055...
9	dwolfe	Python	h	false	true	0.21611548...
10	sheetaluk	Python	h	false	false	0.52602958...
11	sheetaluk	Python	h	false	false	0.48307146...
12	sheetaluk	Python	h	false	false	0.20014810...
13	sheetaluk	Python	h	false	true	0.01715119...
14	sheetaluk	Python	h	false	true	0.03176970...
15	hmstepanek	Python	h	false	false	0.72405249...
16	hmstepanek	Python	h	false	false	0.75960810...
17	hmstepanek	Python	h	false	false	0.81584421...
18	hmstepanek	Python	h	false	false	0.13327882...
19	lyzadanger	Python	h	false	false	0.31321314...
20	lyzadanger	Python	h	false	false	0.12041304...
21	lyzadanger	Python	h	false	true	0.07568985...
22	lyzadanger	Python	h	false	false	0.15499929...
23	lyzadanger	Python	h	false	false	0.23325229...

Resultados por página: 50 1 – 50 de 277912818

Historial de trabajos

Se copió el ID de recurso en el portapapeles.

ACTUALIZAR

pruebagit-segment

Sin título

Model error

tabla_entrenamiento

Sin título 2

tabla_pruebas

anon76

tabla_entrenamiento

CONSULTA

COMPARTIR

COPIAR

INSTANTÁNEA

BORRAR

EXPORTAR

ESQUEMA

DETALLES

VISTA PREVIA

LINAJE

PERFIL DE DATOS

CALIDAD DE LOS DATOS

Fila	actor_login	language	repo	contain	good_jc	random_number
1	seanh	Python	h	false	false	0.78891595...
2	seanh	Python	h	true	true	0.54959925...
3	seanh	Python	h	false	false	0.03344011...
4	seanh	Python	h	false	false	0.77401493...
5	seanh	Python	h	false	false	0.51116669...
6	seanh	Python	h	false	false	0.18009198...
7	seanh	Python	h	false	false	0.58610283...
8	seanh	Python	h	false	false	0.77145978...
9	seanh	Python	h	false	false	0.23804835...
10	seanh	Python	h	false	false	0.71931207...
11	seanh	Python	h	false	true	0.04777233...
12	seanh	Python	h	false	false	0.62347790...
13	seanh	Python	h	false	false	0.41416700...
14	seanh	Python	h	false	false	0.19041898...
15	seanh	Python	h	false	false	0.21996113...
16	seanh	Python	h	false	false	0.12803574...
17	seanh	Python	h	false	false	0.09995410...
18	sheetaluk	Python	h	false	false	0.64707544...
19	hmstepanek	Python	h	false	false	0.42542999...
20	hmstepanek	Python	h	false	false	0.07374992...
21	hmstepanek	Python	h	false	true	0.06582197...
22	hmstepanek	Python	h	false	false	0.21873104...
23	hmstepanek	Python	h	false	false	0.06273161...

Resultados por página: 50 1 – 50 de 222330457

Historial de trabajos

pruebagit-segment

Sin título

Model error

tabla_entrenamiento

Sin título 2

tabla_pruebas

anon76

tabla_pruebas

CONSULTA

COMPARTIR

COPIAR

INSTANTÁNEA

BORRAR

EXPORTAR

ESQUEMA

DETALLES

VISTA PREVIA

LINAJE

PERFIL DE DATOS

CALIDAD DE LOS DATOS

Fila	actor_login	language	repo	contain	good_jc	random_number
1	seanh	Python	h	false	false	0.96002854...
2	seanh	Python	h	false	false	0.83899766...
3	seanh	Python	h	false	false	0.84672378...
4	seanh	Python	h	false	false	0.95057341...
5	seanh	Python	h	true	false	0.82939667...
6	seanh	Python	h	false	false	0.94581418...
7	seanh	Python	h	false	false	0.91071338...
8	seanh	Python	h	false	false	0.81081816...
9	seanh	Python	h	false	false	0.95372954...
10	seanh	Python	h	false	false	0.97497122...
11	seanh	Python	h	false	true	0.94668438...
12	seanh	Python	h	true	true	0.82045183...
13	seanh	Python	h	false	false	0.84983803...
14	seanh	Python	h	false	true	0.80660521...
15	hmstepanek	Python	h	false	true	0.94271023...
16	hmstepanek	Python	h	false	false	0.85446372...
17	hmstepanek	Python	h	false	true	0.80824516...
18	hmstepanek	Python	h	false	false	0.98026273...
19	hmstepanek	Python	h	false	false	0.80858604...
20	hmstepanek	Python	h	false	false	0.82167058...
21	hmstepanek	Python	h	false	false	0.91577051...
22	hmstepanek	Python	h	false	false	0.94942919...
23	hmstepanek	Python	h	false	false	0.98844240...

Resultados por página: 50 1 – 50 de 55582361

Historial de trabajos

Visualizaciones Looker



Modelo Regresión BigQueryML

modelo_contains_error

DETALLES ENTRENAMIENTO EVALUACIÓN ESQUEMA

Ubicación de los datos
US

Detalles del modelo

ID de modelo

bo-f9a1-afme-pf_github_modelo_contains_error

Descripción

Fecha de creación

6 mar 2024, 7:41:23 p.m. UTC-6

Vencimiento del modelo

Nunca

Fecha de modificación

6 mar 2024, 7:41:29 p.m. UTC-6

Ubicación de los datos

US

Tipo de modelo

LOGISTIC_REGRESSION

Tipo de pérdida

Media de la pérdida logística

Datos de entrenamiento

TABLA DE DATOS DE ENTRENAMIENTO TEMPORAL

Datos de evaluación

TABLA DE DATOS DE EVALUACIÓN TEMPORAL

Opciones de entrenamiento

Las opciones de entrenamiento son parámetros opcionales que se agregaron en la secuencia de comandos para agregar este modelo.

Cantidad máxima de iteraciones permitidas

20

Iteraciones reales

7

Regularización L1

0.00

Regularización L2

0.00

Interrupción anticipada

verdadero

Progreso mínimo relativo

0.01

Estrategia de tasa de aprendizaje

Búsqueda lineal

Tasa de aprendizaje

0.10

Intervalo de la búsqueda

Historial de trabajos

modelo_contains_error

DETALLES ENTRENAMIENTO EVALUACIÓN ESQUEMA

Ver como

Grafos

Tabla

Pérdida

Duración (segundos)

Tasa de aprendizaje

Historial de trabajos

modelo_contains_error

CONSULTAR MODELO BORRAR MODELO EXPORTAR MODELO ACTUALIZAR

DETALLES ENTRENAMIENTO EVALUACIÓN ESQUEMA

Métricas agregadas

Umbral

0.5000

Precisión

0.7917

Recuperación

0.0249

Exactitud

0.9256

Puntuación F1

0.0483

Pérdida logística

0.2456

Área bajo la ROC

0.6998

Umbral de puntuación

Umbral de clase positiva

0.0167

Clase positiva

1

Clase negativa

0

Precisión

0.0758

Recuperación

1.0000

Exactitud

0.0758

Puntuación F1

0.1409

Precisión-recuperación por umbral

Curva de precisión-recuperación

Curva ROC

Matriz de confusión

Las matrices de confusión muestran cómo el modelo clasificó cada etiqueta de recurso del conjunto de datos que se evaluó. Las celdas azules en negrita indican que la predicción fue correcta. Los datos se transfieren a la columna de elementos eliminados si no satisfacen el umbral de confianza de cualquier etiqueta.

Historial de trabajos

Matriz de confusión

Recuentos de elementos

Las matrices de confusión muestran cómo el modelo clasificó cada etiqueta de recurso del conjunto de datos que se evaluó. Las celdas azules en negrita indican que la predicción fue correcta. Los datos se transfieren a la columna de elementos eliminados si no satisfacen el umbral de confianza de cualquier etiqueta.

Etiqueta de confianza

Etiqueta predicha

1

0

1

100%

0%

0

100%

0%

Matriz de confusión

Resultados prueba datos test

anon769dbed...	CONSULTA	COMPARTIR	COPIAR	INSTANTÁNEA	BORRAR	EXPORTAR
ESQUEMA	DETALLES	VISTA PREVIA	LINAJE	PERFIL DE DATOS	CALIDAD DE LOS DATOS	
Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.78069804...	0.02361515...	0.92525355...	0.04584359...	0.24521045...	0.71058341...

Modelo Exportado

up-pf-ghtorrent-2024-alme

Ubicación

us (varias regiones en Estados Unidos)

Clase de almacenamiento

Standard

Acceso público

Sujeto a LCA de objeto

Protección

Ninguna

OBJETOS

CONFIGURACIÓN

PERMISOS

PROTECCIÓN

CICLO DE VIDA

OBSERVABILIDAD

INFORMES DE INVENTARIO

Depósitos > up-pf-ghtorrent-2024-alme > pred_model

SUBIR ARCHIVOS

SUBIR CARPETA

CREAR CARPETA

TRANSFERIR LOS DATOS

ADMINISTRAR CONSERVACIONES

EDITAR LA RETENCIÓN

DESCARGAR

BORRAR

Filtrar solo por prefijo de nombre

Filtrar objetos y carpetas

Mostrar datos borrados

<input type="checkbox"/>	Nombre	Tamaño	Tipo	Fecha de creación	Clase de almacenamiento	Última modificación	Acceso público	Historial de versiones	
<input type="checkbox"/>	assets/	—	Carpeta	—	—	—	—	—	
<input type="checkbox"/>	explanation_metadata.json	438 B	application/octet-stream	6 mar 2024 20:50:41	Standard	6 mar 2024 20:50:41	No público	—	
<input type="checkbox"/>	fingerprint.pb	47 B	application/octet-stream	6 mar 2024 20:50:41	Standard	6 mar 2024 20:50:41	No público	—	
<input type="checkbox"/>	saved_model.pb	1 MB	application/octet-stream	6 mar 2024 20:50:41	Standard	6 mar 2024 20:50:41	No público	—	
<input type="checkbox"/>	variables/	—	Carpeta	—	—	—	—	—	

Pruebas API

```
[ev]http server.cc : 245] NET LOG: Entering the event loop ...
curl -d '{"instances": [{"actor_login": "caalador", "language": "PHP"}]}' -X POST http://localhost:8501/v1/models/pred_model:predict
{
  "predictions": [
    {
      "predicted_contains_error": ["0"],
      "contains_error_values": ["1", "0"],
      "contains_error_probs": [0.021372578703077991, 0.97862742129692204]
    }
  ]
}
[g0268379@cloudshell:~ (bd-final-alme)]$
```

Configuración versión

Pre-built container settings

Python version *

3.7

Select the Python version you used to train the model

Framework

TensorFlow

Framework version

1.15.0

ML runtime version *

1.15

Model URI *

gs:// up-pf-ghtorrent-2024-alme/pred_model

EXPLORAR

Cloud Storage path to the entire SavedModel directory. [Learn more](#)

Online prediction deployment

Scaling

Auto scaling

Cantidad mínima de nodos

1

Keeping a minimum number of nodes running all the time will avoid dropping requests due to nodes initialization after the service has scaled down. This setting can increase cost, as you pay for the nodes even when no predictions are served.

Machine type *

n1-standard-4, 4 vCPUs, 15 GB memory

Tipo de acelerador

?

Recuento de aceleradores

?

Service account

Specifies the service account for resource access control.

GUARDAR

BORRAR

CANCELAR