

Процес роботи

НАЛАШТУВАННЯ ОТОЧЕННЯ

1. Активуємо всі рекомендовані API

Get started with Vertex AI

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more about Vertex AI](#)



Tutorials

Try an interactive tutorial to learn how to train, evaluate, and deploy a Vertex AI AutoML or custom-trained model.

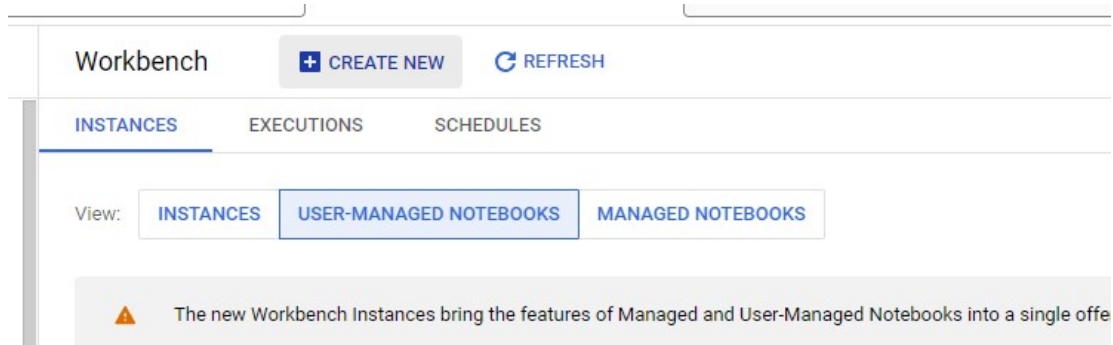
[VIEW TUTORIALS](#)

▼ SHOW API LIST

2. API активовані

Notifications		
✓ Enable service: notebooks.googleapis.com	nk-cloud-course-IAV-21	Just now
✓ Enable service: dataflow.googleapis.com	nk-cloud-course-IAV-21	Just now
✓ Enable service: compute.googleapis.com	nk-cloud-course-IAV-21	Just now
✓ Enable service: aiplatform.googleapis.com	nk-cloud-course-IAV-21	1 minute ago
✓ Enable service: artifactregistry.googleapis.com	nk-cloud-course-IAV-21	1 minute ago
✓ Enable service: visionai.googleapis.com	nk-cloud-course-IAV-21	1 minute ago
✓ Enable service: datacatalog.googleapis.com	nk-cloud-course-IAV-21	1 minute ago
✓ Enable service: dataplex.googleapis.com	nk-cloud-course-IAV-21	1 minute ago
✓ Enable service: dataform.googleapis.com	nk-cloud-course-IAV-21	1 minute ago
✓ Enable service: storage-component.googleapis.com	nk-cloud-course-IAV-21	1 minute ago

3. Створюємо новий інстанс Workbench



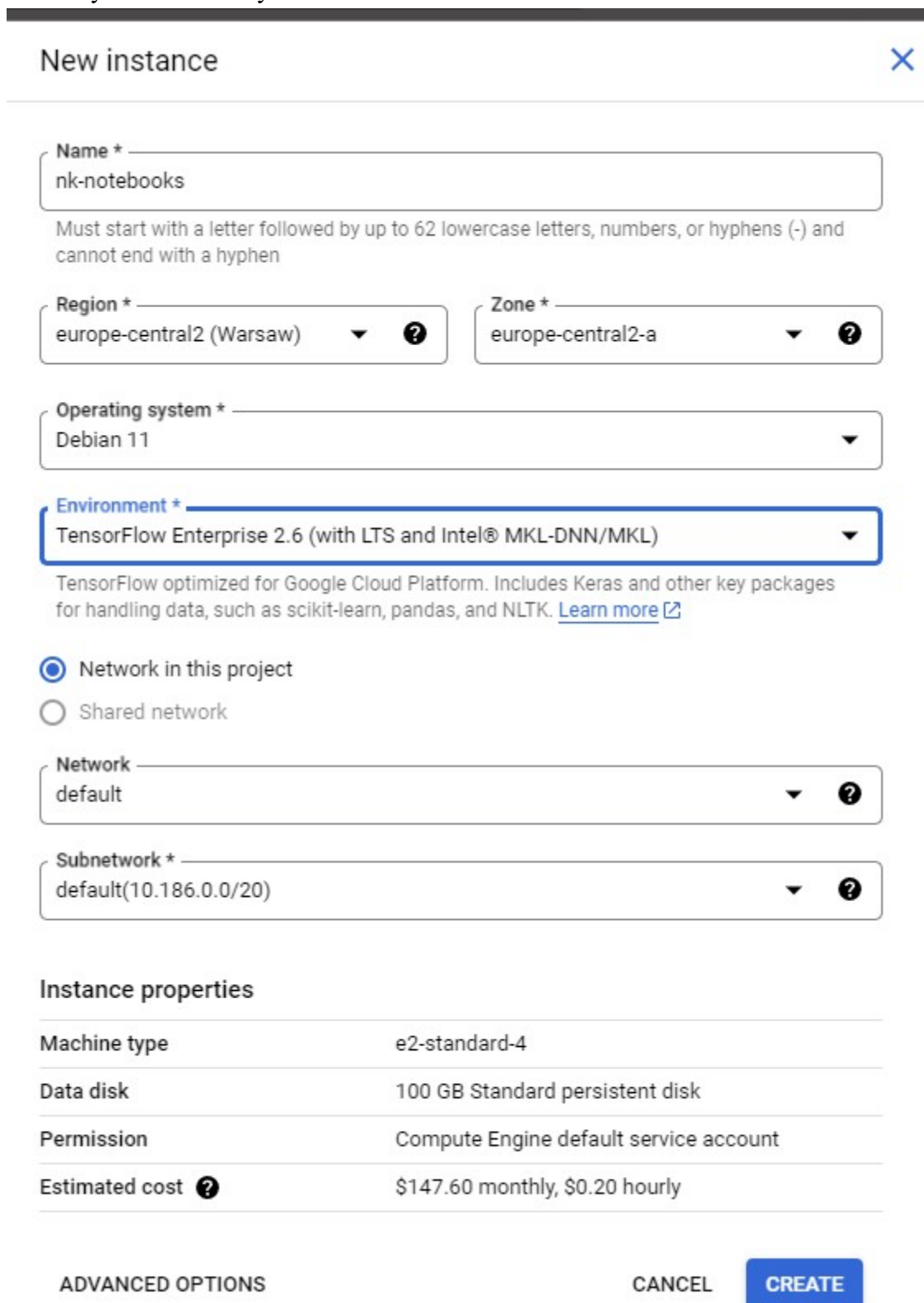
Workbench **+ CREATE NEW** REFRESH

INSTANCES EXECUTIONS SCHEDULES

View: INSTANCES **USER-MANAGED NOTEBOOKS** MANAGED NOTEBOOKS

The new Workbench Instances bring the features of Managed and User-Managed Notebooks into a single offer

4. Налаштування інстансу



New instance

Name *
nk-notebooks
Must start with a letter followed by up to 62 lowercase letters, numbers, or hyphens (-) and cannot end with a hyphen

Region *
europe-central2 (Warsaw)

Zone *
europe-central2-a

Operating system *
Debian 11

Environment *
TensorFlow Enterprise 2.6 (with LTS and Intel® MKL-DNN/MKL)
TensorFlow optimized for Google Cloud Platform. Includes Keras and other key packages for handling data, such as scikit-learn, pandas, and NLTK. [Learn more](#)

☒ Network in this project
☐ Shared network

Network
default

Subnetwork *
default(10.186.0.0/20)

Instance properties

Machine type	e2-standard-4
Data disk	100 GB Standard persistent disk
Permission	Compute Engine default service account
Estimated cost	\$147.60 monthly, \$0.20 hourly

ADVANCED OPTIONS CANCEL CREATE

5. Створений інстанс

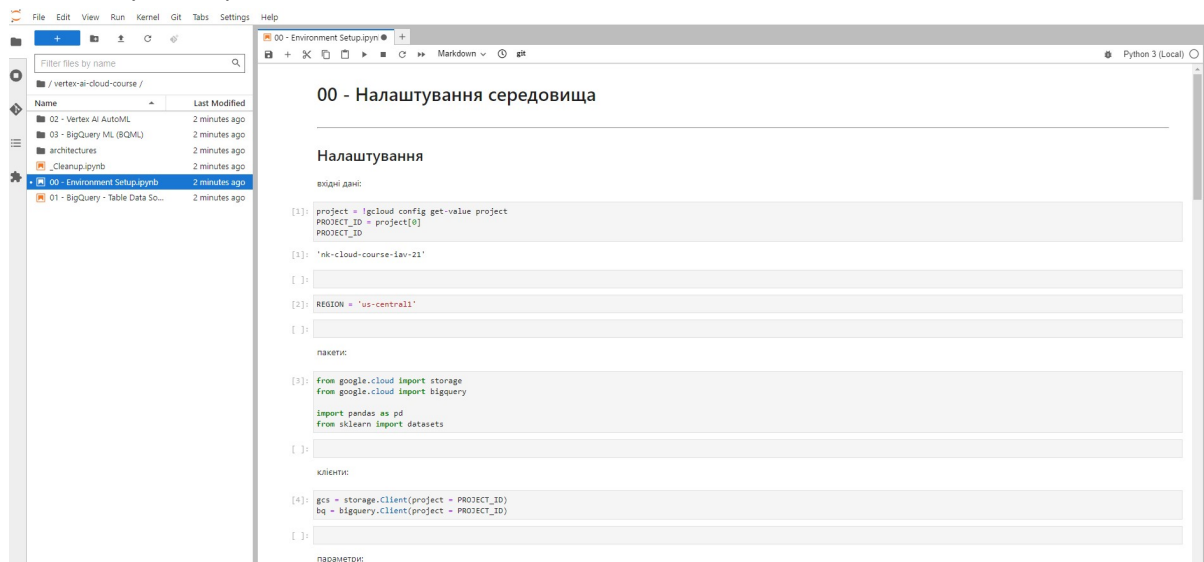
Filter											?	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Notebook name ↑	Zone	Auto upgrade	Environment	Machine Type	GPUs	Owner	Created	Labels		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	nk-notebooks	OPEN JUPYTERLAB	europe-central2-a	—	TensorFlow/2.6	Efficient Instance: 4 vCPUs, 16 GB RAM	None	186934415911-compute@developer.gserviceaccount.com	Apr 1, 2024, 2:48:40 PM	goog-caip-notebook	⋮

6. Клонуємо репозиторій

```
(base) jupyter@nk-notebooks:~$ git clone https://github.com/Aranaur/vertex-ai-cloud-course.git
Cloning into 'vertex-ai-cloud-course'...
remote: Enumerating objects: 562, done.
remote: Counting objects: 100% (87/87), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 562 (delta 49), reused 69 (delta 31), pack-reused 475
Receiving objects: 100% (562/562), 102.78 MiB | 24.51 MiB/s, done.
Resolving deltas: 100% (89/89), done.
(base) jupyter@nk-notebooks:~$
```

НАЛАШТУВАННЯ СЕРЕДОВИЩА

1. Налаштовуємо середовище



2. Створення бакету

Створити кошик для зберігання

Перевіряє, чи вже існує кошик, і створює його, якщо він відсутній:

- GCS Python Client

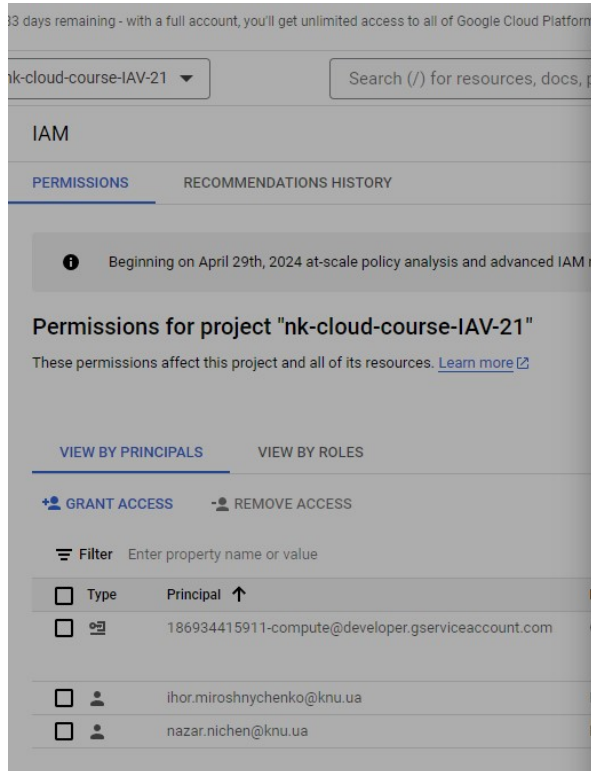
```
[6]: if not gcs.lookup_bucket(BUCKET):
    bucketDef = gcs.bucket(BUCKET)
    bucket = gcs.create_bucket(bucketDef, project=PROJECT_ID, location=REGION)
    print(f'Створення кошика: {gcs.lookup_bucket(BUCKET).name}')
else:
    bucketDef = gcs.bucket(BUCKET)
    print(f'Кошик вже створений: {bucketDef.name}')
```

Створення кошика: nk-cloud-course-iav-21

```
[7]: print(f'Ознайомитися зі сховищем у консолі можна тут: \nhttps://console.cloud.google.com/storage/browser/{PROJECT_ID};tab=objects&project={PROJECT_ID}')
```

Ознайомитися зі сховищем у консолі можна тут:
<https://console.cloud.google.com/storage/browser/nk-cloud-course-iav-21;tab=objects&project=nk-cloud-course-iav-21>

3. Додаємо роль



Edit access to "nk-cloud-course-IAV-21"

Principal 186934415911-compute@developer.gserviceaccount.com

Project nk-cloud-course-IAV-21

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role Editor

View, create, update, and delete most Google Cloud resources. See the list of included permissions.

+ ADD IAM CONDITION

Role Storage Object Admin

Grants full control over objects, including listing, creating, viewing, and deleting objects.

+ ADD IAM CONDITION

+ ADD ANOTHER ROLE

SAVE TEST CHANGES CANCEL

4. Встановлюємо KFP

Встановіть KFP

Якщо ви отримуєте помилку після якогось кроку, повторіть його. Іноді залежні види вирішуються.

- Встановлення Kubeflow Pipelines SDK

```
[16]: !pip install kfp -U -q
```

```
[15]: !pip install google-cloud-pipeline-components -U -q
```

5. Оновлюємо пакет AIPlatform

Оновити пакет AIPlatform:

Пакет `google-cloud-aiplatform` часто оновлюється. Оновіть його, щоб отримати найновішу функціональність.

- [aiplatform Python Client](#)
- [GitHub Repo для api-common-protos](#)

```
[17]: !pip install googleapis-common-protos -U -q
```

```
[ ]:
```

```
[18]: !pip install google-cloud-aiplatform -U -q
```

```
[ ]:
```

```
[19]: from google.cloud import aiplatform
aiplatform.__version__
```

```
[19]: '1.46.0'
```

ІМПОРТУВАННЯ ДАНИХ

1. Створення набору даних і таблиці з CSV файлу з Google Cloud Storage

Create table

Source

Create table from

Google Cloud Storage

Select file from GCS bucket or [use a URI pattern](#) *



nk-cloud-course-iav-21/PS_20174392719_1491204439457_log.csv

File format

CSV

☐ Source Data Partitioning

Destination

Project *

nk-cloud-course-iav-21

Dataset *

nk_fraud_detection

Table *

transactions

Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, a

Table type

Native table

Schema

☐ Auto detect



Edit as text



Partition and cluster settings

Partitioning

No partitioning

CREATE TABLE

CANCEL

2. Семпл даних

transactions

QUERYSHARECOPYSNAPSHOTDELETEDEXPORT

REF

SCHEMA		DETAILS		PREVIEW	LINEAGE	DATA PROFILE		DATA QUALITY					
Row	step	type	amount	nameOrig	oldbalanceOrig	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud		
1	256	DEBIT	759.91	C459353451	80493.5	79733.59	C1011243187	1327439.0	1328198.91	0	0		
2	256	DEBIT	4172.55	C2091244856	83814.1	79641.55	C79242915	594701.71	598874.27	0	0		
3	256	DEBIT	4753.26	C1280637881	76813.42	72060.16	C1159010125	223510.42	228263.68	0	0		
4	256	DEBIT	6635.54	C1538351083	94590.85	87955.31	C882500755	662454.34	669089.87	0	0		
5	256	DEBIT	7598.01	C1941156974	10200.0	2601.99	C1787826365	323148.38	330746.39	0	0		
6	256	DEBIT	6050.1	C1073144857	112272.87	106222.77	C707218751	1066849.62	1072899.72	0	0		
7	256	DEBIT	11100.55	C1994815725	11408.0	307.45	C2007266923	54198.17	65298.72	0	0		
8	256	DEBIT	3408.9	C882177036	101452.33	98043.42	C1349040375	14749.03	18157.93	0	0		
9	256	DEBIT	42069.65	C970826484	3711.0	0.0	C656082643	186945.2	229014.85	0	0		
10	256	DEBIT	3456.04	C736976470	55767.0	52310.96	C997795367	37729.0	41185.04	0	0		
11	256	CASH_IN	231702.91	C1496727695	51747.0	283449.91	C438845176	1850109.38	1618406.47	0	0		
12	256	CASH_IN	48783.96	C2110673707	3305826.89	3354610.85	C1300365380	857496.6	808712.64	0	0		
13	256	CASH_IN	107728.8	C193048229	9375671.2	9483400.0	C567451688	671708.18	563979.38	0	0		
14	256	CASH_IN	233290.84	C990280038	162.0	233452.84	C475228670	188791.88	0.0	0	0		
15	256	CASH_IN	121136.93	C78295515	436106.13	557243.06	C792855360	570013.23	448876.3	0	0		
16	256	CASH_IN	398902.95	C168834899	7074.0	405976.95	C1004476143	258111.42	0.0	0	0		
17	256	CASH_IN	69790.17	C1426743506	0.0	69790.17	C1382895402	190909.64	121119.47	0	0		
18	256	CASH_IN	627516.44	C1999165047	10314841.55	10942358.0	C1641635888	1938582.69	1311066.25	0	0		
19	256	CASH_IN	334645.16	C128678492	5142.0	339787.16	C1779666818	56250.94	383882.61	0	0		
20	256	CASH_IN	75147.71	C1817340285	600061.97	675209.68	C1497198716	1832669.2	1757521.49	0	0		
21	256	CASH_IN	97716.67	C220634407	687724.15	785440.82	C285196065	142017.44	44300.77	0	0		
22	256	CASH_IN	528361.02	C663124494	6951714.01	7480075.02	C509119222	696945.91	168584.89	0	0		
23	256	CASH_IN	493699.21	C2124325427	12057.0	505756.21	C36223390	1329529.08	835829.87	0	0		
24	256	CASH_IN	330366.72	C365231682	22003.0	352369.72	C885827198	0.0	0.0	0	0		
25	256	CASH_IN	51390.1	C40839335	3205255.73	3256645.83	C1031223422	2698561.37	2647171.27	0	0		

Results per page: 501 - 50 of 6362620

НАЛАШТУВАННЯ ДЖЕРЕЛА ДАНИХ

1. Налаштування

Налаштування

вхідні дані:

[1]: project = !gcloud config get-value project
PROJECT_ID = project[0]
PROJECT_ID

[1]: 'nk-cloud-course-iav-21'

[2]: REGION = 'us-central1'
EXPERIMENT = '01'
SERIES = '01'

source data
BQ_PROJECT = PROJECT_ID
BQ_DATASET = 'nk_fraud_detection'
BQ_TABLE = 'transactions'

Data source for this series of notebooks: Described above
BQ_SOURCE = 'nk-cloud-course-iav-21.nk_fraud_detection'

пакести:

[3]: from google.cloud import bigquery
from google.cloud import storage

клієнти:

[4]: bq = bigquery.Client(project = PROJECT_ID)
gcs = storage.Client(project = PROJECT_ID)

параметри:

[5]: BUCKET = PROJECT_ID

[]:

2. Тестовий семпл через клієнт

Отримання та перегляд вибірки з таблиці:

Примітка: Оператор `LIMIT 5` обмежує кількість рядків, що повертаються BigQuery, до 5, але BigQuery все одно виконує повне сканування таблиці. Якщо у вас таблиця більша за 1 ГБ і ви хочете обмежити кількість рядків, що скануються для швидкого перегляду, то ефективнішою буде заміна `LIMIT 5` на `TABLESAMPLE SYSTEM (1 PERCENT)`. Для таблиць розміром менше 1 ГБ програма все одно повертатиме повну таблицю. Докладніше про [тут](#)

```
[6]: query = f"""
SELECT *
FROM `{BQ_PROJECT}`.{BQ_DATASET}`.{BQ_TABLE}` TABLESAMPLE SYSTEM (1 PERCENT)
#LIMIT 5
"""
bq.query(query = query).to_dataframe()
```

```
[6]:
```

	step	type	amount	nameOrig	oldbalanceOrig	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	256	DEBIT	9188.06	C2000977273	105395.00	96206.94	C942954195	46905.24	56093.30	0	0
1	256	DEBIT	12440.64	C1395724234	18512.00	6071.36	C1970400978	36996.65	49437.29	0	0
2	256	DEBIT	6115.72	C1562392919	192854.00	186738.28	C1455016120	822845.67	828961.39	0	0
3	256	DEBIT	1781.32	C43505966	21441.00	19659.68	C1901141963	111854.73	113636.06	0	0
4	256	CASH_IN	185135.37	C1892734251	50698.00	235833.37	C798921761	59894.57	0.00	0	0
...
374109	255	TRANSFER	438431.81	C1131496788	0.00	0.00	C1502424827	1813683.76	2252115.57	0	0
374110	255	TRANSFER	78124.96	C156320073	154824.00	76699.04	C1630298588	0.00	78124.96	0	0
374111	255	TRANSFER	721889.60	C1386514082	20364.02	0.00	C130693448	4789805.91	5511695.51	0	0
374112	255	TRANSFER	255158.69	C1633378	0.00	0.00	C986326114	2638691.06	2893849.75	0	0
374113	255	TRANSFER	369635.83	C276312102	497714.00	128078.17	C1359805956	0.00	369635.83	0	0

374114 rows x 11 columns

3. Перегляд даних у BigQuery

Перегляд даних у BigQuery

Для перегляду даних можна використовувати додаткові SQL-запити. У цьому розділі показано переміщення таблиці до фрейму даних Pandas для локального перегляду у Python:

Примітка:

Цей запит вибирає лише один стовпець. Це означає, що BigQuery сканує менше даних, оскільки не обробляє інші стовпці.

```
[9]: query = f"""
SELECT isFraud
FROM `{BQ_PROJECT}`.{BQ_DATASET}`.{BQ_TABLE}`
"""
df = bq.query(query = query).to_dataframe()
```

```
[ ]:
```

```
[ ]:
```

```
[11]: df['isFraud'].value_counts()
```

```
[11]: 0    6354407
      1     8213
      Name: isFraud, dtype: Int64
```

```
[ ]:
```

```
[ ]:
```

```
[12]: df['isFraud'].value_counts(normalize=True)
```

```
[12]: 0    0.998709
      1    0.001291
      Name: isFraud, dtype: Float64
```

```
[ ]:
```



4. Підготовка даних до аналізу

Підготовка даних для аналізу

Створіть заздалегідь підготовлену версію даних з тестовим/тренувальним розбиттям за допомогою SQL DDL:

```
[13]: query = f"""
CREATE TABLE IF NOT EXISTS `{BQ_PROJECT}`.`{BQ_DATASET}`.`{BQ_TABLE}`_prepped` AS
WITH add_id AS(SELECT *, GENERATE_UUID() transaction_id FROM `{BQ_PROJECT}`.`{BQ_DATASET}`.`{BQ_TABLE}`)
SELECT *,
CASE
    WHEN MOD(ABS(FARM_FINGERPRINT(transaction_id)),10) < 8 THEN "TRAIN"
    WHEN MOD(ABS(FARM_FINGERPRINT(transaction_id)),10) < 9 THEN "VALIDATE"
    ELSE "TEST"
END AS splits
FROM add_id
"""
job = bq.query(query = query)
job.result()
```

```
[13]: <google.cloud.bigquery.table._EmptyRowIterator at 0x7fcfae2f7e80>
```

```
[ ]:
```

```
[14]: (job.ended-job.started).total_seconds()
```

```
[14]: 7.116
```

```
[ ]:
```

```
[15]: if job.estimated_bytes_processed:
    print(f'{job.estimated_bytes_processed/1000000} MB')
```

```
625.99573 MB
```

```
[ ]:
```

Перегляньте розподіл між навчальною та тестовою вибірками:

```
[16]: query = f"""
SELECT splits, count(*) as Count, 100*count(*) / (sum(count(*) OVER()) as Percentage
FROM `{BQ_PROJECT}`.`{BQ_DATASET}`.`{BQ_TABLE}`_prepped`
GROUP BY splits
"""
bq.query(query = query).to_dataframe()
```


[16]:

	splits	Count	Percentage
0	VALIDATE	636938	10.010625
1	TRAIN	5090391	80.004636
2	TEST	635291	9.984739

[]:

Отримати підмножину даних до фрейму даних Pandas:

[17]:

```
query = f"""
SELECT *
FROM `{BQ_PROJECT}`.{BQ_DATASET}`.{BQ_TABLE}_prepped`
LIMIT 5
"""
data = bq.query(query = query).to_dataframe()
```

[]:

[18]:

```
data.head()
```

[18]:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud	transaction_id	split
0	256	DEBIT	9426.95	C1548168480	29945.00	20518.05	C407349438	468269.88	477696.83	0	0	506439ae-464b-459a-964b-f41b3e97acea	TRAIN
1	256	DEBIT	9647.87	C1738691985	80776.00	71128.13	C536995054	310670.76	320318.63	0	0	96fa5652-99cc-4daf-ba3f-cbb4fd0e678f	TRAIN
2	256	DEBIT	3332.05	C25949445	112891.00	109558.95	C41788738	3305665.25	3308997.31	0	0	8b3a16c7-f1e0-40ca-b220-13b0535f207d	TRAIN
3	256	DEBIT	2429.53	C1546127826	106222.77	103793.24	C1017031904	232310.90	234740.43	0	0	fac9c895-dd08-4743-90cb-5a60786cabb3	TEST
4	256	DEBIT	2704.30	C398991382	1627.73	0.00	C1702465296	2340019.02	2342723.32	0	0	228b1a76-2942-4fe8-ab04-1e2ccac66803	TRAIN

ПОБУДОВА МОДЕЛІ

1. Вкладка Analyze для зареєстрованого набору даних

← 02a

SOURCEANALYZE

Analyze

Properties

Created

Apr 01, 2024 8:16 PM

Dataset format

BigQuery

Dataset location(s)

[bq://nk-cloud-cour...nsactions_prepped](#)

Encryption type

Google-managed

Summary

Total columns: 13

Total rows: -

INTEGER

3 (23.08%)

STRING

5 (38.46%)

FLOAT

5 (38.46%)

Filter

Enter property name or value

Column name ↑	BigQuery type	BigQuery mode	Missing % (count) ?	Distinct values
amount	FLOAT	NULLABLE	-	-
isFlaggedFraud	INTEGER	NULLABLE	-	-
isFraud	INTEGER	NULLABLE	-	-
nameDest	STRING	NULLABLE	-	-
nameOrig	STRING	NULLABLE	-	-

2. Дашборд датасетів

Datasets + CREATE

Managed datasets contain data used to train a machine learning model. [Learn more](#)

Region

us-central1 (Iowa)

Filter

Enter a property name

<input type="checkbox"/>	Name	ID	Status	Region	Type
<input type="checkbox"/>	02a	2327478099513442304	✓ Ready	us-central1	Tabular

3. Створена модель у процесі навчання

Training

+ TRAIN NEW MODEL

REFRESH

TRAINING PIPELINES

CUSTOM JOBS

HYPERPARAMETER TUNING JOBS

NAS JOBS

PERSISTENT JOBS

Training pipelines are the primary model training workflow in Vertex AI. You can use training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps like adding a dataset or uploading the model to Vertex AI for prediction serving. [Learn more](#)

Region

us-central1 (Iowa)

Filter

Enter a property name

Name	ID	Status	Job type	Model type
02a	7499144324815257600	<div>Pending</div>	Training pipeline	<div>Tabular classification</div>

4. Навчена модель

Filter

Enter a property name

Name	ID	Status	Job type	Model type
02a	7499144324815257600	<div>Finished</div>	Training pipeline	<div>Tabular classification</div>

5. Результати

Threshold target

Filter

All labels

0

1

Evaluation details

Confidence threshold

0.5

All labels

PR AUC	1
ROC AUC	1
Log loss	0.001
Micro-average F1	0.99965525
Macro-average F1	0.9241352
Micro-average precision	100%
Micro-average recall	100%
Total items	0
Training items	0
Validation items	0
Test items	0

To evaluate your model, set the confidence threshold to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some [example scenarios](#) to learn how evaluation metrics can be used.

Precision-recall curve

ROC curve



РОЗГОРТАННЯ МОДЕЛІ

1. Розгортання моделі під ендпоінтом

Deploy your model

Endpoints are machine learning models made available for online prediction requests. Endpoints are useful for timely predictions from many users (for example, in response to an application request). You can also request batch predictions if you don't need immediate results.

DEPLOY TO ENDPOINT

	Name	ID	Status	Models	Deployment
	02a	955191930537377792	 Deploying model	0	—

ПРОГНОЗУВАННЯ

1. Налаштування

Прогнозування

вхідні дані:

```
[1]: project = !gcloud config get-value project
PROJECT_ID = project[0]
PROJECT_ID
```

```
[1]: 'nk-cloud-course-iav-21'
```

```
[19]: REGION = 'us-central1'
DATASET = 'nk-cloud-course-iav-21'
DATABASE = 'nk_fraud_detection'
TABLE = 'transactions'
NOTEBOOK = '02a'

# Model Training
VAR_TARGET = 'isFraud'
VAR_OMIT = 'transaction_id nameOrig nameDest isFlaggedFraud splits' # додайте більше змінних через пробіл
```

пакети:

```
[20]: from google.cloud import aiplatform

from google.cloud import bigquery
from google.protobuf import json_format
from google.protobuf.struct_pb2 import Value
import json
import numpy as np
```

клієнти:

```
[21]: aiplatform.init(project=PROJECT_ID, location=REGION)
bigquery = bigquery.Client()
```

параметри:

```
[22]: DIR = f"temp/{NOTEBOOK}"
```

середовище:

```
[23]: !rm -rf {DIR}
!mkdir -p {DIR}
```

2. Підготовка спостережень для прогнозування

Підготуйте спостереження для прогнозування

```
[24]: pred = bigquery.query(query = f"SELECT * FROM {DATASET}.{DATABASE}.{TABLE}_prepped WHERE splits='TEST' LIMIT 10").to_dataframe()

[25]: pred.head(4)
```

	step	type	amount	nameOrig	oldbalanceOrig	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	256	DEBIT	32898.66	C1298651968	12243.00	0.00	C1494799126	232802.90	265701.57	0	
1	256	CASH_IN	233336.04	C2060327049	46645.00	279981.04	C1502126152	0.00	0.00	0	
2	256	CASH_IN	89350.50	C96593198	937026.89	1026377.39	C972585449	1039371.68	950021.18	0	
3	256	CASH_IN	48489.96	C1075655098	2214.00	50703.96	C1917872398	6710518.09	6662028.13	0	

```
[26]: #newob = pred[pred.columns[~pred.columns.isin(VAR_OMIT.split()+[VAR_TARGET, 'splits'])]].to_dict(orient='records')[0]
newob = pred[pred.columns[~pred.columns.isin(VAR_OMIT.split()+[VAR_TARGET])]].to_dict(orient='records')[0]
#newob

Потрібно розуміти формат змінних, які очікує прогноз. AutoML може перетворити тип деяких змінних. Наступні комірки отримують модель з кін
```

```
[31]: newob['step'] = str(newob['step'])

[32]: instances = [json_format.ParseDict(newob, Value())]
parameters = json_format.ParseDict({}, Value())
```

3. Отримання прогнозів: Клієнт Python

Отримання прогнозів: Клієнт Python

```
[14]: endpoint = aiplatform.Endpoint.list(filter=f'display_name={NOTEBOOK}')[0]
      endpoint.display_name

[14]: '02a'

[33]: prediction = endpoint.predict(instances=instances, parameters=parameters)
      prediction

[33]: Prediction(predictions=[{'classes': ['0', '1'], 'scores': [0.9996766448020935, 0.000323380867484957]}], deployed_model_id='6731454276826038272', metadata={'source_name': 'projects/186934415911/locations/us-central1/models/2507696851398950912', explanations=None})

[34]: prediction.predictions[0]['classes'][np.argmax(prediction.predictions[0]['scores'])]

[34]: '0'
```

4. Отримання прогнозів: REST

Отримання прогнозів: REST

```
[35]: with open(f'{DIR}/request.json', 'w') as file:
      file.write(json.dumps({"instances": [newob]}))

[36]: !curl -X POST \
      -H "Authorization: Bearer $(gcloud auth application-default print-access-token) \
      -H "Content-Type: application/json; charset=utf-8" \
      -d @${DIR}/request.json \
      https://{REGION}-aiplatform.googleapis.com/v1/{endpoint.resource_name}:predict

{
  "predictions": [
    {
      "scores": [
        0.99967664480209351,
        0.000323380867484957
      ],
      "classes": [
        "0",
        "1"
      ]
    }
  ],
  "deployedModelId": "6731454276826038272",
  "model": "projects/186934415911/locations/us-central1/models/2507696851398950912",
  "modelDisplayName": "02a",
  "modelVersionId": "1"
}
```


5. Отримання прогнозів: gcloud (CLI)

Отримання прогнозів: gcloud (CLI)

```
!gcloud beta ai endpoints predict {endpoint.name.rsplit('/',1)[-1]} --region={REGION} --json-request={DIR}/request.json  
Using endpoint [https://us-central1-prediction-aiplatform.googleapis.com/]  
[{'classes': ['0', '1'], 'scores': [0.9996766448020935, 0.000323380867484957]}]
```

6. Інтерпретація результатів

Інтерпретація результатів

Гайд

- <https://cloud.google.com/vertex-ai/docs/predictions/interpreting-results-automl#tabular>

```
[41]: explanation = endpoint.explain(instances=instances, parameters=parameters)
```

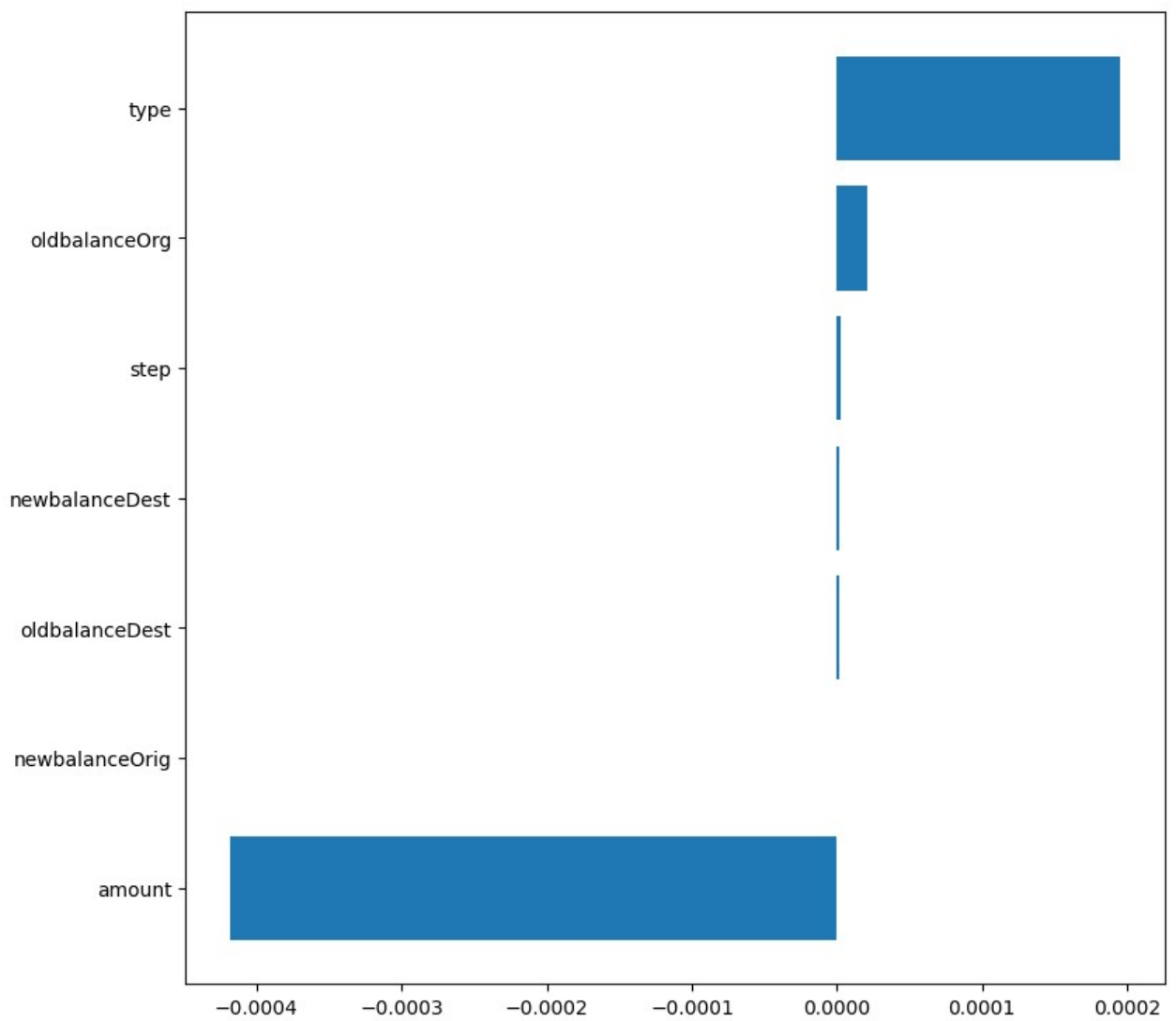
```
[42]: explanation.predictions
```

```
[42]: [{'classes': ['0', '1'], 'scores': [0.9996766448020935, 0.000323380867484957]}]
```

```
[43]: print("attribution:")  
print("baseline output",explanation.explanations[0].attributions[0].baseline_output_value)  
print("instance output",explanation.explanations[0].attributions[0].instance_output_value)  
print("output_index",explanation.explanations[0].attributions[0].output_index)  
print("output display value",explanation.explanations[0].attributions[0].output_display_name)  
print("approximation error",explanation.explanations[0].attributions[0].approximation_error)
```

```
attribution:  
baseline output 0.9998730421066284  
instance output 0.9996766448020935  
output_index [0]  
output display value 0  
approximation error 0.0013967822445738335
```

```
[44]: import matplotlib.pyplot as plt  
features = []  
scores = []  
for k in explanation.explanations[0].attributions[0].feature_attributions:  
    features.append(k)  
    scores.append(explanation.explanations[0].attributions[0].feature_attributions[k])  
features = [x for _, x in sorted(zip(scores, features))]  
scores = sorted(scores)  
fig, ax = plt.subplots()  
fig.set_size_inches(9, 9)  
ax.barh(features, scores)  
fig.show()
```

АНАЛІЗ ТА ВИСНОВКИ

- Модель навчалась на датасеті 640 MB приблизно 2.25 годин.
- Показано дуже добре результати з precision = 99.4% та recall = 74% при порозі впевненості 0.5.
- Найбільш важливими фічерами виявились oldbalanceOrg та newbalanceOrig.
- Розгнута модель точно класифікувала транзакції