

MLOps Assignment 2

Group Number: 35

Group Member:

Manglam Kumar (2022ac05260)

Saurabh Kumar (2022ac05293)

Rajiv Kumar (2022ac05147)

Neeraj Kumar (2022ac05468)

Krishna Kumar v (2022ac05373)

Repository Link: https://github.com/manglamsingh10/MLOps_Assignment_2

Code Snippets:

https://github.com/manglamsingh10/MLOps_Assignment_2/blob/main/MLOps_Assignment_2_group35.ipynb

Project Description:

This project aims to build a complete end-to-end machine learning (ML) workflow using **Google Cloud Platform Vertex AI AutoML**.

Details:

Data Collection and Preprocessing:

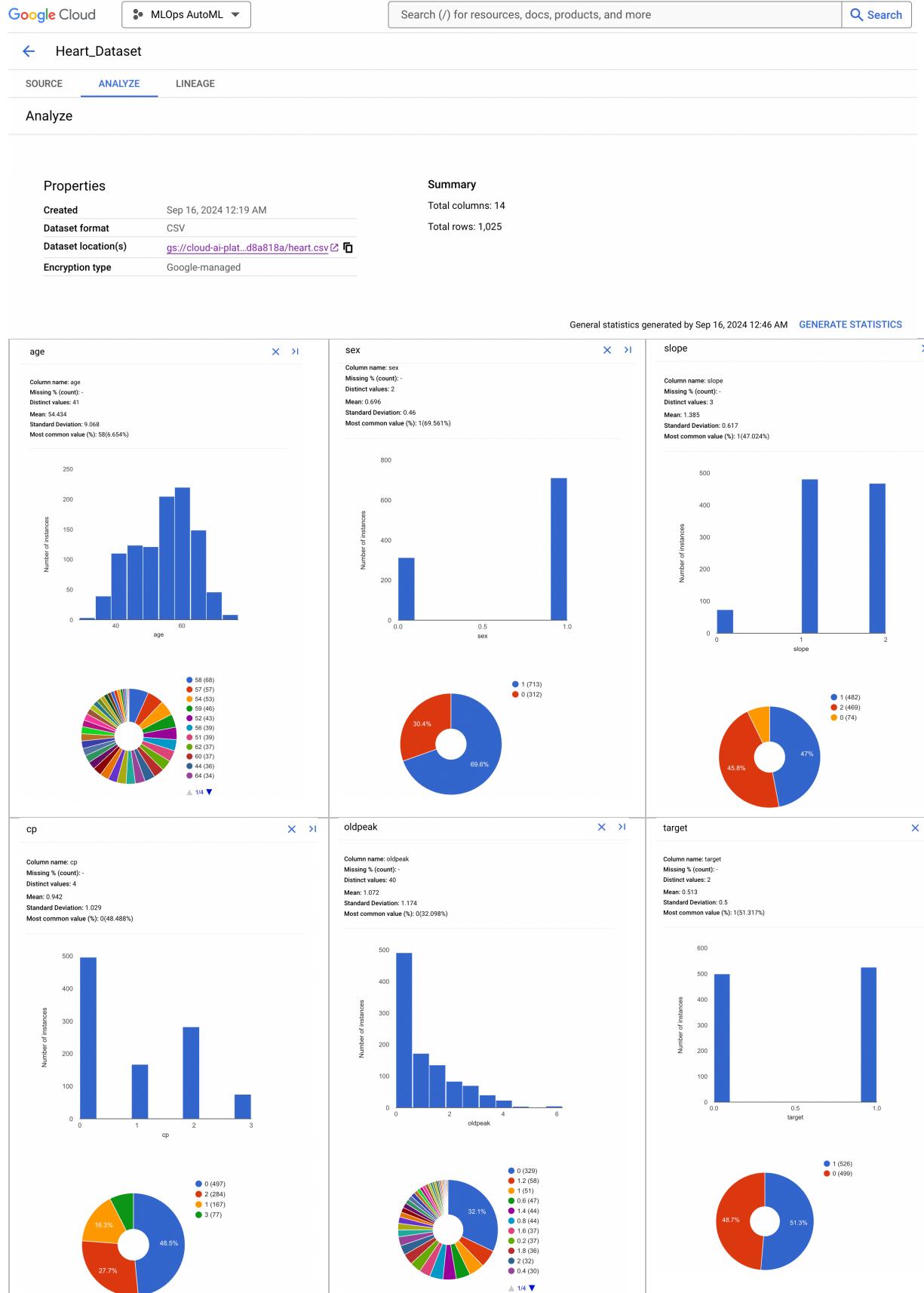
In this step we preprocesses a dataset for a classification task. It converts specific columns to numeric types, handles missing values by dropping rows with NaN values, and creates a new categorical feature for age groups. It then converts categorical variables into dummy variables, scales numerical features using StandardScaler, and splits the data into training and testing sets. Finally, it prints the shape of the pre-processed data and the list of feature names.

Data types:	Missing values:
age object	age 1
sex object	sex 1
cp object	cp 1
trestbps object	trestbps 1
chol object	chol 1
fbs object	fbs 1
restecg object	restecg 1
thalach object	thalach 1
exang object	exang 1
oldpeak object	oldpeak 1
slope object	slope 1
ca object	ca 1
thal object	thal 1
target object	target 1
dtype: object	dtype: int64

Preprocessed data shape: (1025, 27)
Features: ['age', 'sex', 'trestbps', 'chol', 'fbs', 'thalach', 'exang', 'oldpeak', 'ca', 'cp_0.0', 'cp_1.0', 'cp_2.0', 'cp_3.0', 'restecg_0.0', 'restecg_1.0', 'restecg_2.0', 'slope_0.0', 'slope_1.0', 'slope_2.0', 'thal_0.0', 'thal_1.0', 'thal_2.0', 'thal_3.0', 'age_group_0-40', 'age_group_41-50', 'age_group_51-60', 'age_group_60+']

We used Google's Vertex AI AutoML Managed Dataset to Store the data. This automatically do all the preprocessing and visualization for the data.

Google storage Bucket Location: <gs://cloud-ai-platform-9672eb94-cc15-4d36-9868-d4f0fd8a818a/heart.csv>



Model Selection, Training, and Hyperparameter Tuning:

We performed hyperparameter tuning and evaluation for multiple machine learning models using RandomizedSearchCV. It iterates over a dictionary of models and their respective hyperparameters, fits each model to the training data, predicts on the test data, calculates the accuracy, and stores the best model and its accuracy in a results dictionary. Finally, it prints the best hyperparameters, accuracy, and a classification report for each model.

Random Forest – Best params: {'n_estimators': 300, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': 30}

Random Forest – Accuracy: 0.9853658536585366

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	102
1.0	1.00	0.97	0.99	103
accuracy			0.99	205
macro avg	0.99	0.99	0.99	205
weighted avg	0.99	0.99	0.99	205

Gradient Boosting – Best params: {'n_estimators': 300, 'max_depth': 5, 'learning_rate': 0.1}

Gradient Boosting – Accuracy: 0.9853658536585366

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	102
1.0	1.00	0.97	0.99	103
accuracy			0.99	205
macro avg	0.99	0.99	0.99	205
weighted avg	0.99	0.99	0.99	205

SVM – Best params: {'kernel': 'rbf', 'gamma': 'scale', 'C': 10}

SVM – Accuracy: 0.9853658536585366

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	102
1.0	1.00	0.97	0.99	103
accuracy			0.99	205
macro avg	0.99	0.99	0.99	205
weighted avg	0.99	0.99	0.99	205

Best Model: Random Forest

Best Accuracy: 0.9853658536585366

Google AutoML Vertex AI Model training & Hyperparameter tuning:

The screenshot shows two panels of the Google AutoML Vertex AI interface for model training and hyperparameter tuning.

Left Panel (Train new model):

- Training method:** Set to "AutoML".
- Dataset:** Heart_Dataset
- Objective:** Classification
- Model details:** (checkboxes checked)
- Join Featurestore (optional):** (checkbox checked)
- Training options:** (checkbox checked)
- Compute and pricing:** (checkbox checked)
- START TRAINING** and **CANCEL** buttons.

Right Panel (Train new model):

- Training method:** Set to "AutoML".
- Model details:** (checkbox checked)
- Training container:** (checkbox checked)
- Hyperparameters (optional):** (checkbox checked)
- Compute and pricing:** (checkbox checked)
- START TRAINING** and **CANCEL** buttons.

Hyperparameter tuning variables:

- n_estimators:** Discrete values: 100, 200, 300
- max_depth:** Discrete values: None, 10, 20, 30
- min_samples_split:** Integer values: 2 - 10
- min_samples_leaf:** Integer values: 1 - 4

Hyperparameter tuning variables configuration:

- Metric to optimize:** accuracy
- Goal:** Maximize
- Maximum number of trials:** 100
- Maximum number of parallel trials:** 5
- Algorithm:** Grid search

CONTINUE button at the bottom right.

Google Cloud MLOps AutoML Search (/) for resources, docs, products, and more

Training TRAIN NEW MODEL REFRESH

TRAINING PIPELINES CUSTOM JOBS HYPERPARAMETER TUNING JOBS NAS JOBS PERSISTENT RESOURCES

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	Duration	Last updated	Created	Ended	Labels
Heart_Dataset	7325639259648425984	Finished	Training pipeline	Tabular classification	1 hr 53 min	Sep 16, 2024, 2:30:40 AM	Sep 16, 2024, 12:36:49 AM	Sep 16, 2024, 2:30:40 AM	-

Google Cloud MLOps AutoML Search (/) for resources, docs, products, and more

Heart Dataset Model Version 1 VIEW DATASET EXPORT

EVALUATE DEPLOY & TEST BATCH PREDICT VERSION DETAILS LINEAGE

untitled_48...0873757529 COMPARE CREATE EVALUATION

Threshold target Evaluation details

All labels Confidence threshold 0.5

All labels	0.878
1	0.898
0	0.874

All labels

PR AUC	0.878
ROC AUC	0.868
Log loss	0.456
Micro-average F1	0.8053097
Macro-average F1	0.8045597
Micro-average precision	80.5%
Micro-average recall	80.5%
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 Precision-recall by threshold

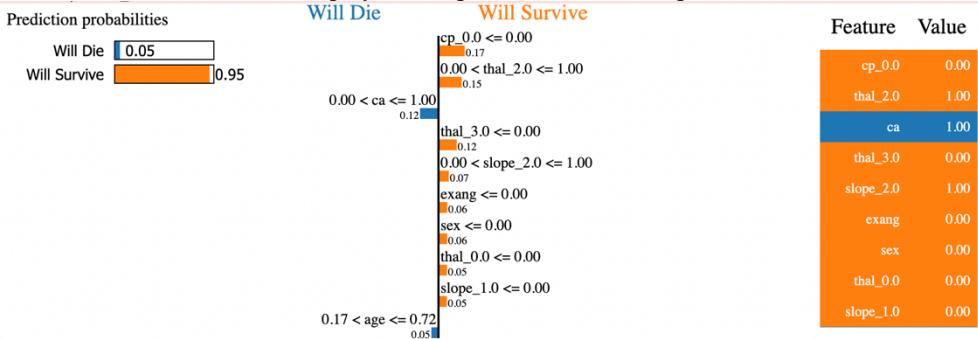
Confusion matrix Item counts

A confusion matrix shows how the model classified each label in the evaluation dataset. The blue, bold cells indicate a correct prediction. A data item is moved to the dropped column if it does not meet the confidence threshold for any label.

True label	Predicted label		Dropped
	1	0	
1	83% (bold)	17%	0%
0	22%	78% (bold)	0%

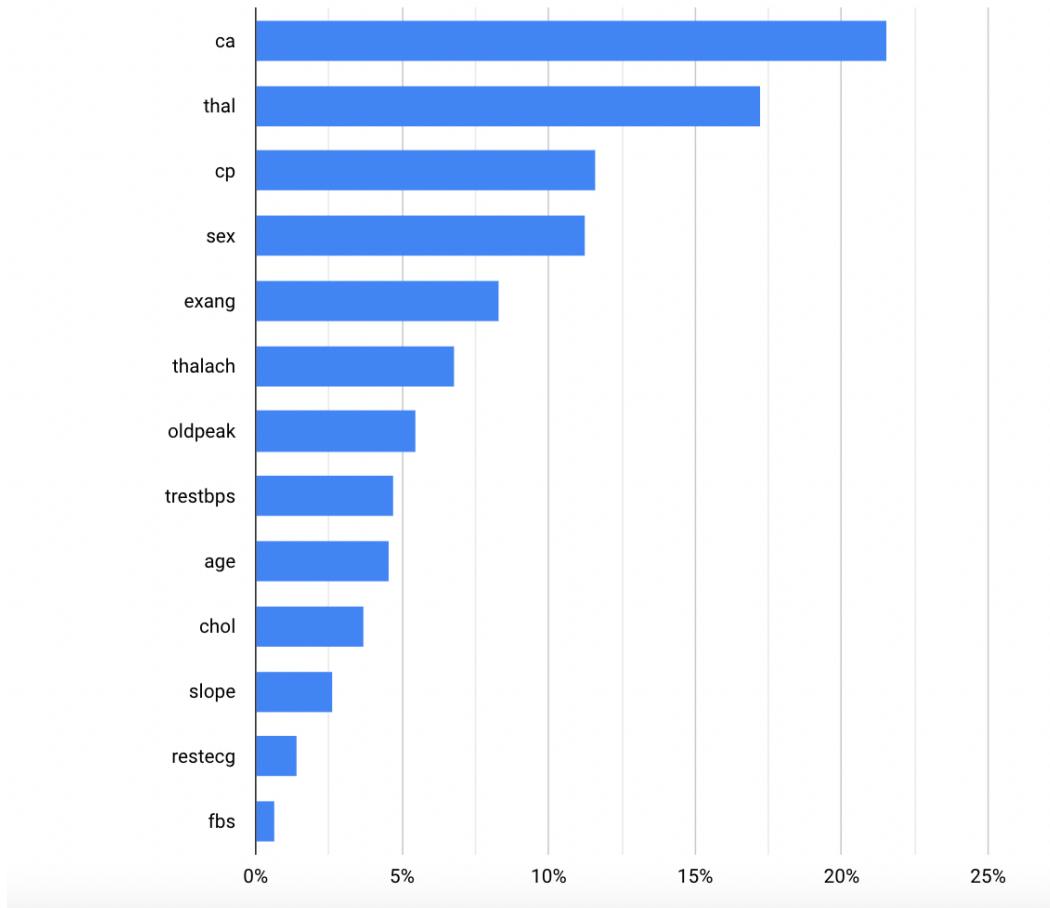
Explainable AI (XAI) Implementation:

This project uses LIME library to explain the prediction of a machine learning model for a specific instance from the test dataset and displays the explanation with the top 10 features



Feature importance

Model feature attribution tells you how important each feature is when making a prediction. Attribution values are expressed as a percentage; the higher the percentage, the more strongly that feature impacts a prediction on average. Model feature attribution is expressed using the Sampled Shapley method. [Learn more ↗](#)



We also used **SHAP method** to explain the output of a prediction with the help of **shapely values**.

Call Vertex AI explain API to get the explanation for given prediction:

Predict API URL: <https://us-central1-aiplatform.googleapis.com/v1/projects/1098200441517/locations/us-central1/endpoints/3215782339686694912:predict>

Explain API URL: <https://us-central1-aiplatform.googleapis.com/v1/projects/1098200441517/locations/us-central1/endpoints/3215782339686694912:explain>

Both of above API used Bearer token for authentication. To generate the token use command ‘`gcloud auth print-access-token`’

```
curl \
-X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
"https://us-central1-aiplatform.googleapis.com/v1/projects/${PROJECT_ID}/locations/us-central1/endpoints/${ENDPOINT_ID}:predict" \
-d "@${INPUT_DATA_FILE}"
```

```
POST Predict from Google Ve | POST XAI Call Google Vertex • +
HTTP Assignment-2 / XAI Call Google Vertex AI
POST https://us-central1-aiplatform.googleapis.com/v1/projects/1098200441517/locations/us-central1/endpoints/3215782339686694912:explain
Send
Params Authorization Headers (9) Body Scripts Tests Settings Cookies
Auth Type Bearer Token Token ya29.a0AcM612xLc2Q_j2Z3M9SbaORH9Jq...
The authorization header will be automatically added to all outgoing requests.
Body Cookies Headers (13) Test Results
Pretty Raw Preview Visualize JSON
1 {
  2   "explanations": [
  3     {
  4       "attributions": [
  5         {
  6           "baselineOutputValue": 0.33805019809649539,
  7           "instanceOutputValue": 0.8696933388710022,
  8           "featureAttributions": [
  9             {
  10               "exang": 0,
  11               "trestbps": 0.0036327495024754452,
  12               "fbs": 0,
  13               "cp": 0,
  14               "slope": -0.066729126068261951,
  15               "chol": -0.01672893877212818,
  16               "sex": 0,
  17               "age": -0.01209710882260249,
  18               "thalach": -0.0055899459582108716,
  19               "thal": 0.219514325261116,
  20               "ca": 0.3972569188246436,
  21               "oldpeak": 0.01238426680748279,
  22               "restecg": 0
  23             },
  24             "outputIndex": [
  25               1
  26             ],
  27             "outputDisplayName": "0",
  28             "approximationError": 0.0015564780021687162,
  29             "outputName": "scores"
  30           ]
  31         }
  32       ],
  33       "denormalizedModelId": "30052681670105088"
  34     }
  35   ]
  36 }
```

baselineOutputValue: This is the predicted output value for a baseline instance, often an average or neutral input used as a reference for comparison. For screenshot attached, it's 0.338.

instanceOutputValue: This is the model's output (prediction) for the specific instance (input) being evaluated. For screenshot attached, it's 0.8697. This indicates how confident the model is in its prediction for this particular case.

featureAttributions: This shows the contribution of each feature to the model's prediction.

Model Deployment Using Cloud Services:

Once the best model is selected, it upload the model file to a Google Cloud Storage bucket, then it imports a machine learning model from a Google Cloud Storage bucket to Google Vertex AI Model Registry

The screenshot shows the Google Cloud Model Registry interface. At the top, there's a search bar and navigation icons. Below it, a table lists a single model entry:

Name	Default version	Deployment status	Description	Type	Source	Updated	Labels
Heart Dataset Model	1	Deployed	--	Tabular	AutoML training	Sep 16, 2024, 6:22:50 PM	--

Once the model is available to Model Registry then it check for precondition and finally deploys it to a **Vertex AI endpoint**.

The screenshot shows the Google Cloud Model Registry interface for the "Heart API Prediction" model. It includes tabs for EVALUATE, DEPLOY & TEST, BATCH PREDICT, VERSION DETAILS, and LINEAGE. Under the EVALUATE tab, there's a "Test your model" section with a "PREVIEW" button. The preview table shows feature values and local feature importance:

Feature column name	Type	Value	Local feature importance
age	Text	56.0	0
sex	Text	1	0
cp	Text	0	0

On the right, there are sections for Predict label (target), Prediction result (Selected label: 1), and Baseline prediction value (0.6619498133659363) and Confidence score (0.6619498133659363).

Monitoring & Logs:

The screenshot shows the Google Cloud Monitoring & Logs interface for the "Heart API Prediction" model. It includes tabs for PERFORMANCE and RESOURCE USAGE. The PERFORMANCE tab displays a line chart of Predictions/second over time, with a sharp peak around 11 AM. The RESOURCE USAGE tab shows a table of deployment logs:

Region	Logs	Model Monitoring
us-central1	View Logs	Disabled

Below the chart, a log entry for "Heart_Dataset (Version 1)" is shown:

1 2024-09-17 02:52:43.349 INFO:tornado.access:200 POST /predict (10.36.1.4) 33 69ms

The screenshot shows a detailed view of the Google Cloud Monitoring & Logs interface. It features a "Log fields" sidebar with filters for RESOURCE TYPE (Vertex AI Endpoint) and SEVERITY (Info). The main area shows a timeline of log entries:

Severity	Time	Log
INFO	Sep 17 1:54:00AM	INFO:tornado.access:200 POST /predict (10.36.1.4) 33 69ms
INFO	Sep 17 1:54:34M	{"@type": "type.googleapis.com/google.cloud.aiplatform.logging.OnlinePredictionLogEntry", "deployedModelId": "309526816878105088", "endpoint": "projects/1098288441517/1...", "resource.labels.endpoint_id": "3215782339686694912", "resource.labels.location": "us-central1"}
INFO	Sep 17 1:54:35M	INFO:tornado.access:200 POST /predict (10.36.1.4) 28 89ms
INFO	Sep 17 1:54:35M	{"@type": "type.googleapis.com/google.cloud.aiplatform.logging.OnlinePredictionLogEntry", "deployedModelId": "309526816878105088", "endpoint": "projects/1098288441517/1...", "resource.labels.endpoint_id": "3215782339686694912", "resource.labels.location": "us-central1"}