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.

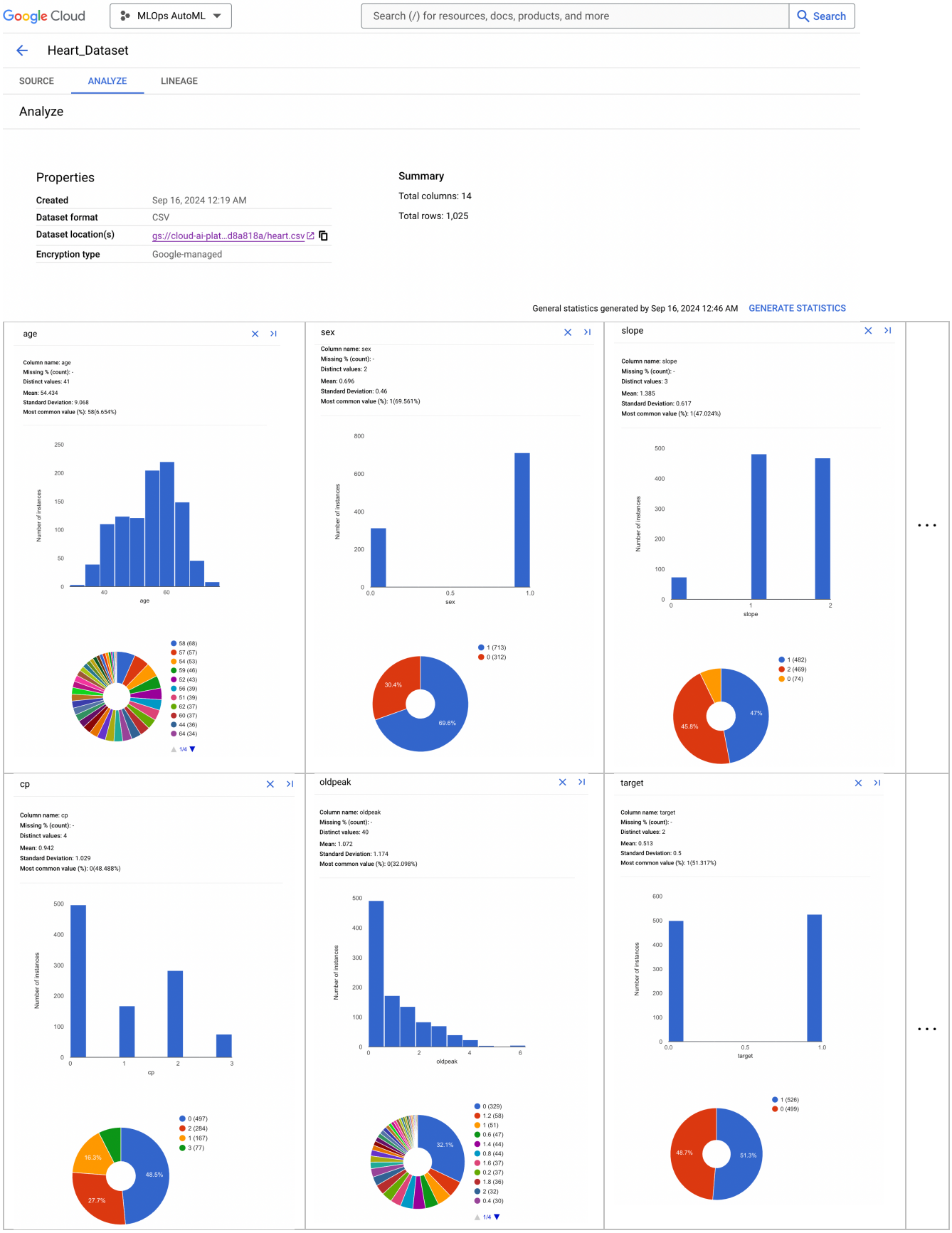
```
Data types:
age          object
sex          object
cp           object
trestbps     object
chol         object
fbs          object
restecg      object
thalach      object
exang        object
oldpeak      object
slope        object
ca           object
thal         object
target       object
dtype: object
```

```
Missing values:
age          1
sex          1
cp           1
trestbps     1
chol         1
fbs          1
restecg      1
thalach      1
exang        1
oldpeak      1
slope        1
ca           1
thal         1
target       1
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', 'a
ge_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

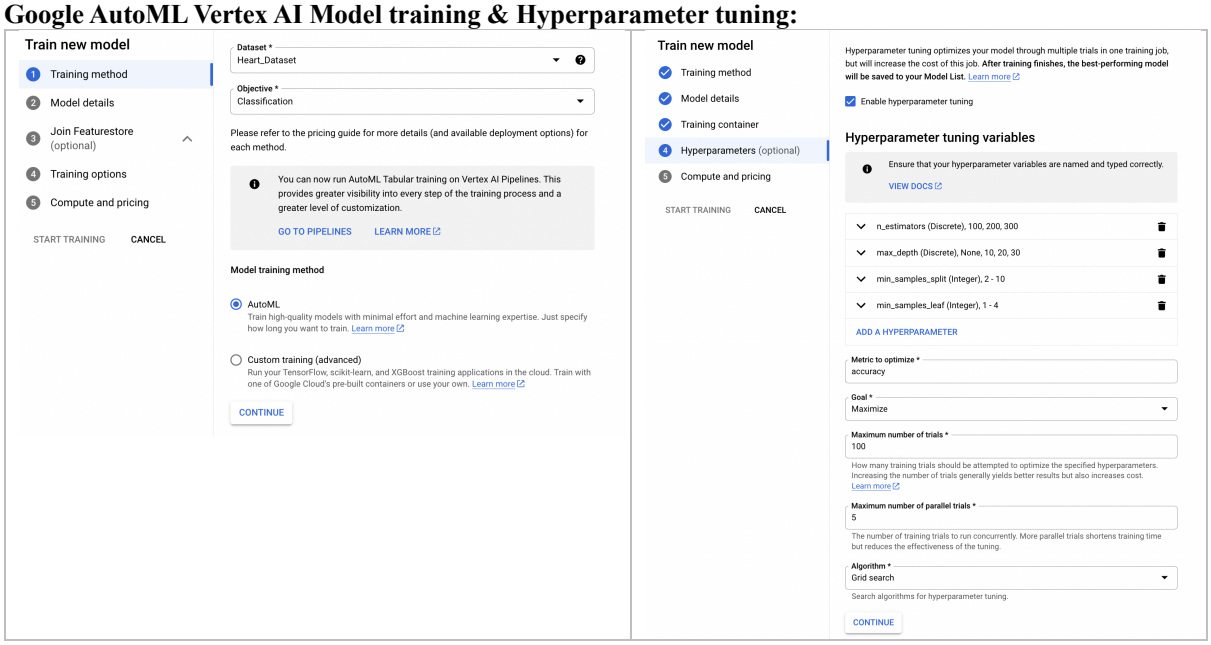


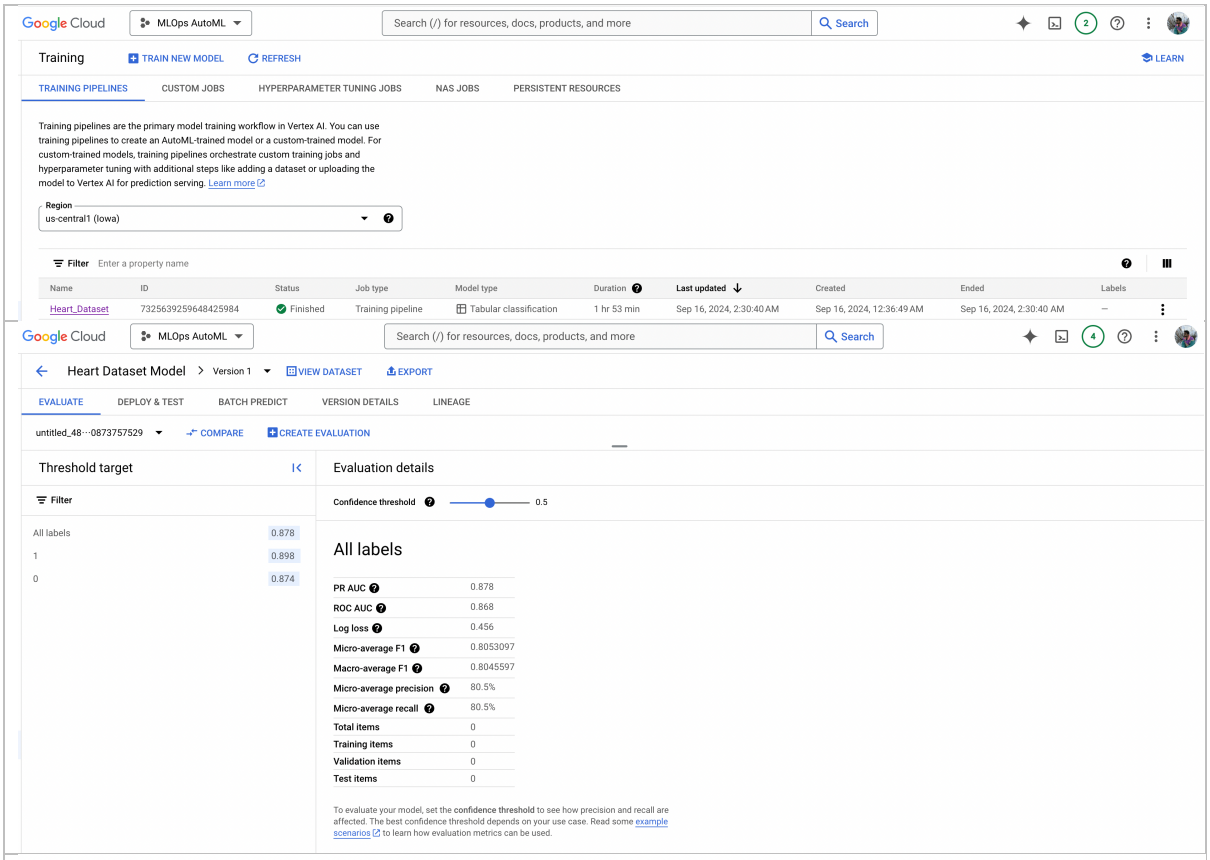
Model Selection, Training, and Hyperparameter Tuning:

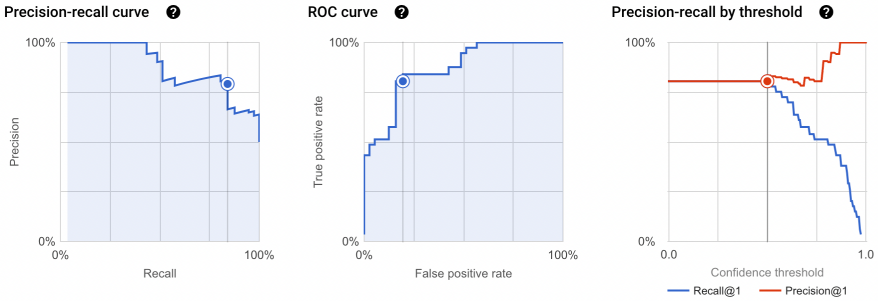
We performed hyperparameter tuning and evaluation for multiple machine learning models using RandomizedSearchCV. It iterates over a dictionary of models and their respective hyperparameters, fits each model to the training data, predicts on the test data, calculates the accuracy, and stores the best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







Confusion matrix

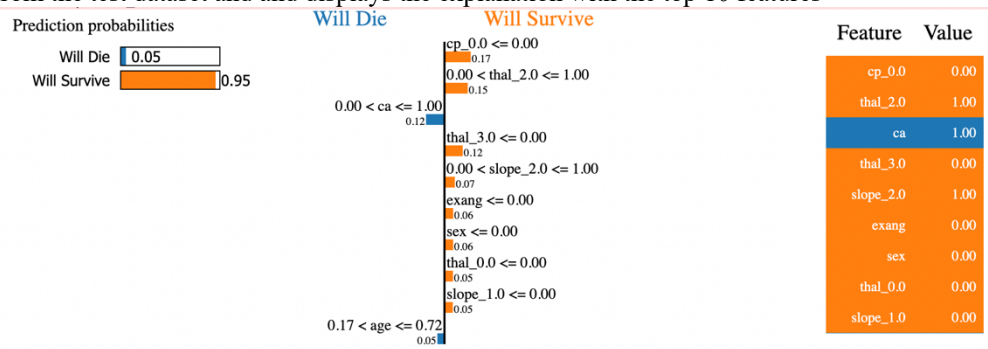
Item counts

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: 1	0	Dropped
1	83%	17%	0%
0	22%	78%	0%

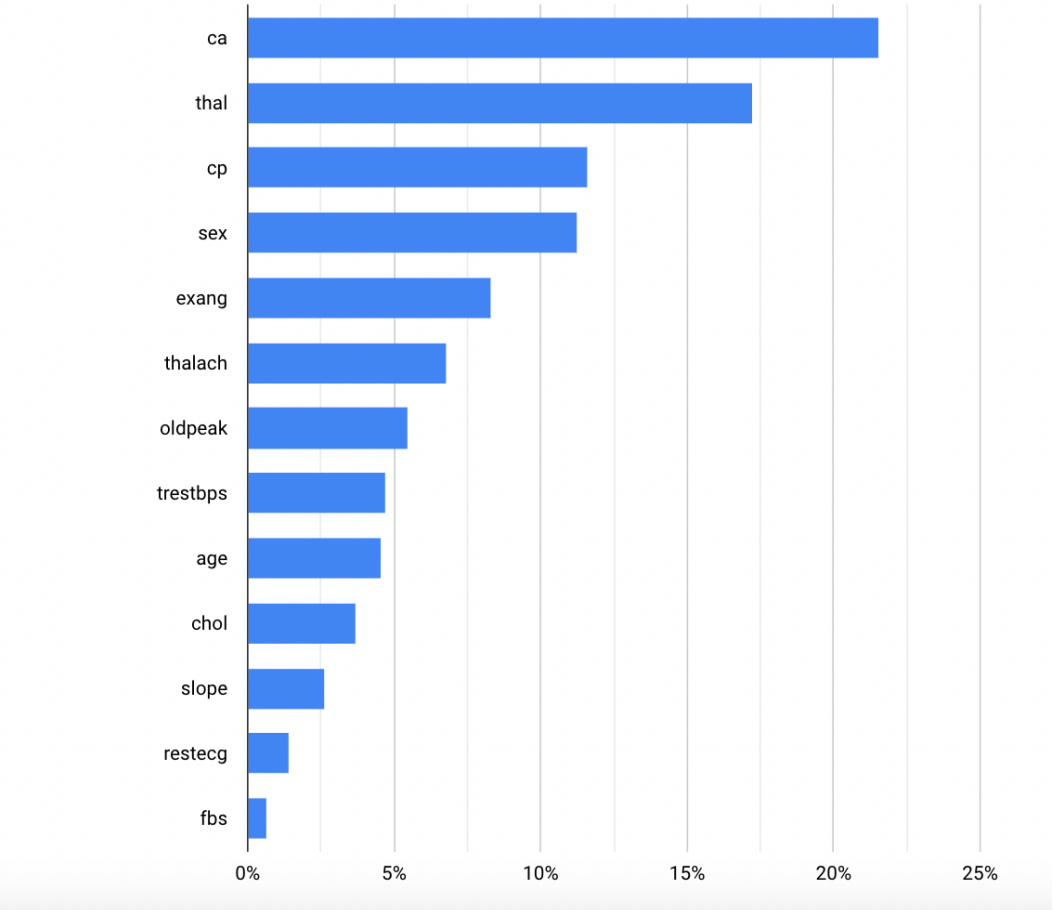
Explainable AI (XAI) Implementation:

This project uses LIME library to explain the prediction of a machine learning model for a specific instance from the test dataset and 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**.

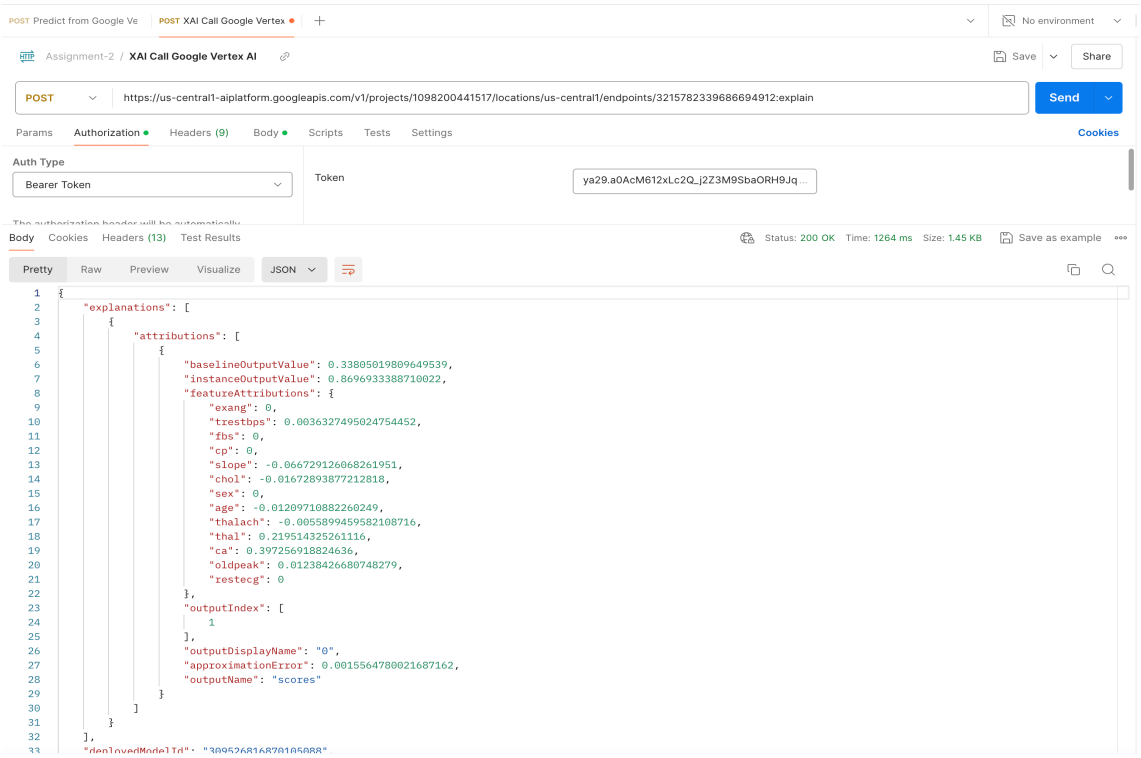
Call Vertex AI explain API to get the explanation for given prediction:

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}"
```



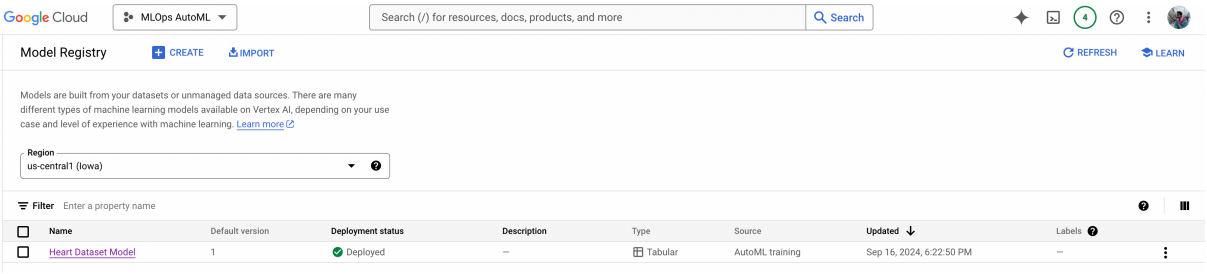
baselineOutputValue: This is the predicted output value for a baseline instance, often an average or neutral input used as a reference for comparison. For screenshot attached, it's 0.338.

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.

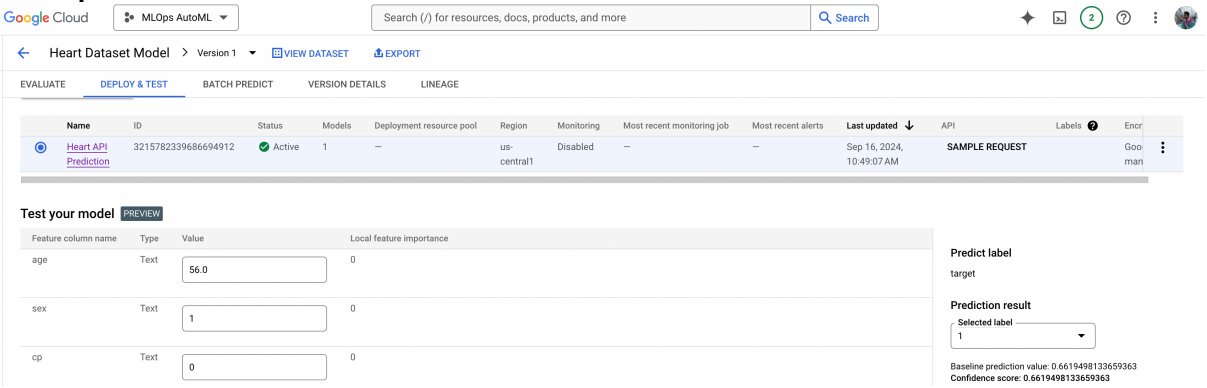
featureAttributions: This shows the contribution of each feature to the model's prediction.

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**



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



Monitoring & Logs: