Project 1: predictions of the emissions and electricity consumption

Denis Storozhuk, Insy2s

Data link: https://data.seattle.gov/dataset/2016-Building-Energy-Benchmarking/2bpz-gwpy

Agenda

- 0. Brief and goals setting
- 1. Python environment:
    - Data cleaning and transformation
    - Choosing the best model
    - Hyperparameter tuning
- 2. BigQuery environment:
    - BQ ML: ETL data, modelling
    - Vertex AI Auto ML: ETL, modelling
- 3. Summary and key learnings

Brief and goals setting

- Since our goal is to predict emissions (Y0) and electricity consumption (Y1), all the variables of electricity consumption to be deleted from X
- Y emissions - TotalGHGEmissions
- Y electricity - Electricity(kWh)
- Regression problem

Python Environment

Load the dataset

1/ load the latest data from the link, draft data analysis

2/ changing coma in the false object columns and transforming them to numerical (a problem appears while using Jupiter Cloud version)

dropping anomalies - below 0 values for Y columns (TotalGHGEmissions, Electricity(kWh))

Delete non-informative columns

3/ deleting Id and non-informative columns for modelling,

Columns to delete:

OSEBuildingID, DataYear, PropertyName, Address, City, State, TaxParcelIdentificationNumber, YearsENERGYSTARCertified, DefaultData, Comments, Outlier

dropping the columns with electricity and gas consumption data and GHGEmissionsIntensity

'SiteEUI(kBtu/sf)','SiteEUIWN(kBtu/sf)','SourceEUI(kBtu/sf)','SourceEUIWN(kBtu/sf)','SiteEnergyUse(kBtu)','SiteEnergyUseWN(kBtu)','SteamUse(kBtu)','Electricity(kWh)','Electricity(kBtu)','NaturalGas(therms)','NaturalGas(kBtu)','GHGEmissionsIntensity'

Categorical and numerical data transformation

4/ splitting data into numerical and categorical

5/ replacing nans by median/most frequent

6/ implementing automatic Anova approach for treatment of the categorical columns, delete the columns which have no impact on Y (TotalGHGEmissions) (with p_value > 0.05)

7/ selecting numerical columns with weak correlation (threshold < 0.5) to be deleted

8/ implementing OrdinalEncoder for the categorical data

9/ perturbation of the dataset (PS: avoided due to small size of the dataset and problems with reproductivity)

Modelling and hyperparameter tuning

10/ outliers removal (therefore no need to keep the Outlier column which was dropped initially)

11/ scaling/standardizing data (Scaler)

14/ splitting data into X and y (TotalGHGEmissions), yel (Electricity(kWh))

15/ train_test_split of the dataset (check random_state to be equal for the two splits)

16/ checking performance of a set of the different models with fit/predict (KFold/cross_val_score can't be used due to a small size of a dataset and problems with reproductivity)

17/ retrieving features importance

18/ hyperparameter tuning to improve the model performance

Final list of the columns

Final list of X columns: BuildingType, PrimaryPropertyType, Neighborhood, PropertyGFATotal, PropertyGFABuilding(s), ListOfAllPropertyUseTypes, LargestPropertyUseType, LargestPropertyUseTypeGFA, SecondLargestPropertyUseType, ThirdLargestPropertyUseType, NaturalGas(therms)

Y columns: TotalGHGEmissions, Electricity(kWh)

Regression problem

Summary of the results for test data

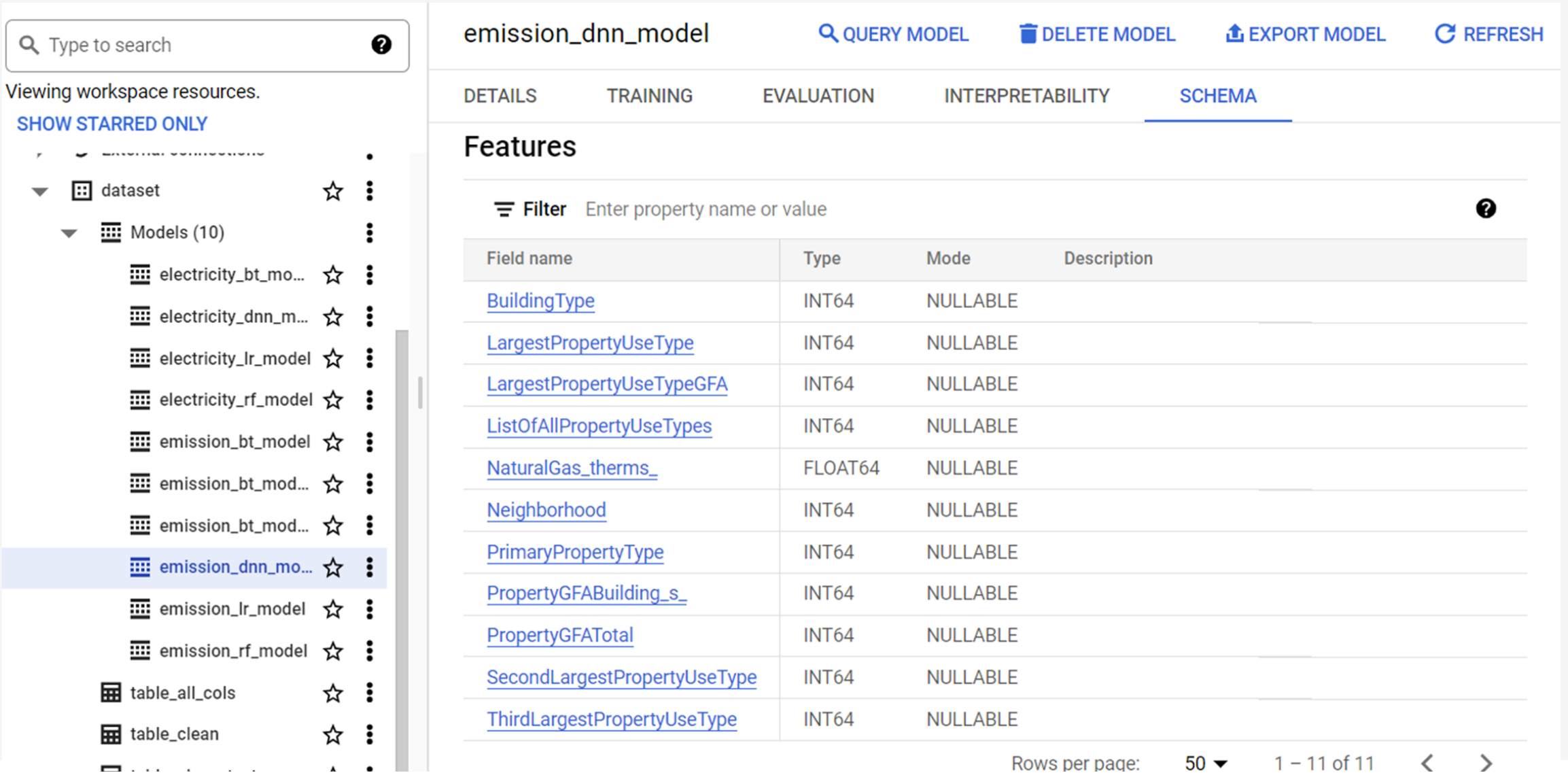
Model / Y	R2	RMSE
Python: Random Forest (TotalGHGEmissions)	0.89	62.3
Python: XGB (Electricity(kWh))	0.88	950 769

BigQuery ML

ETL data

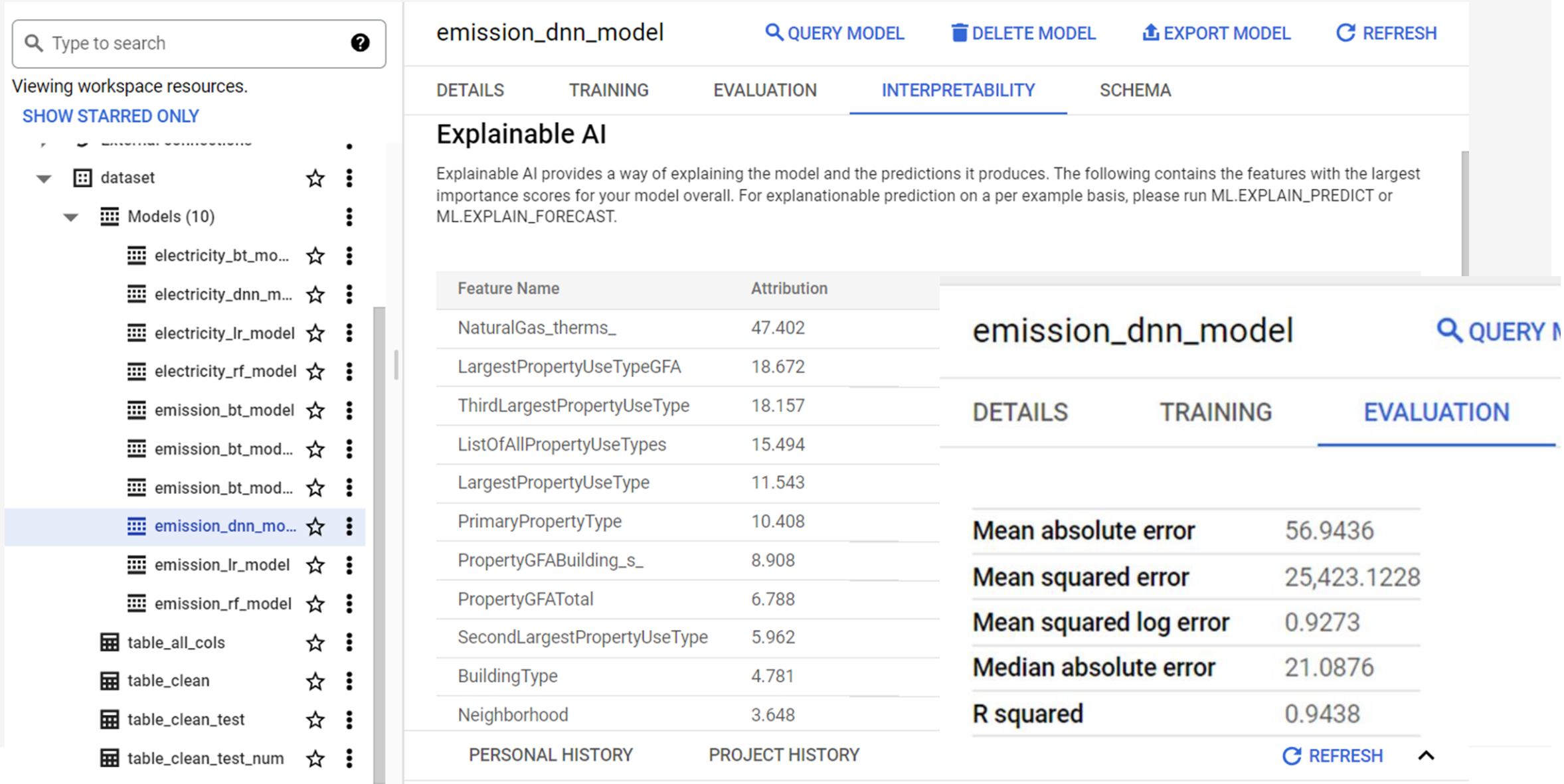
- Though BQ ML automatically treats nans, I encountered problems with categorical columns (String data type in BQ). The models showed bad performance
- Therefore I did manual data transformation with ML.LABEL_ENCODER for categorical columns (String data type in BQ) for BQ ML
- Additional data transformation (optional since it didn’t improve the results) – replace nans with median/most frequent (PERCENTILE_CONT, APPROX_TOP_COUNT)

Manual encoding for string columns

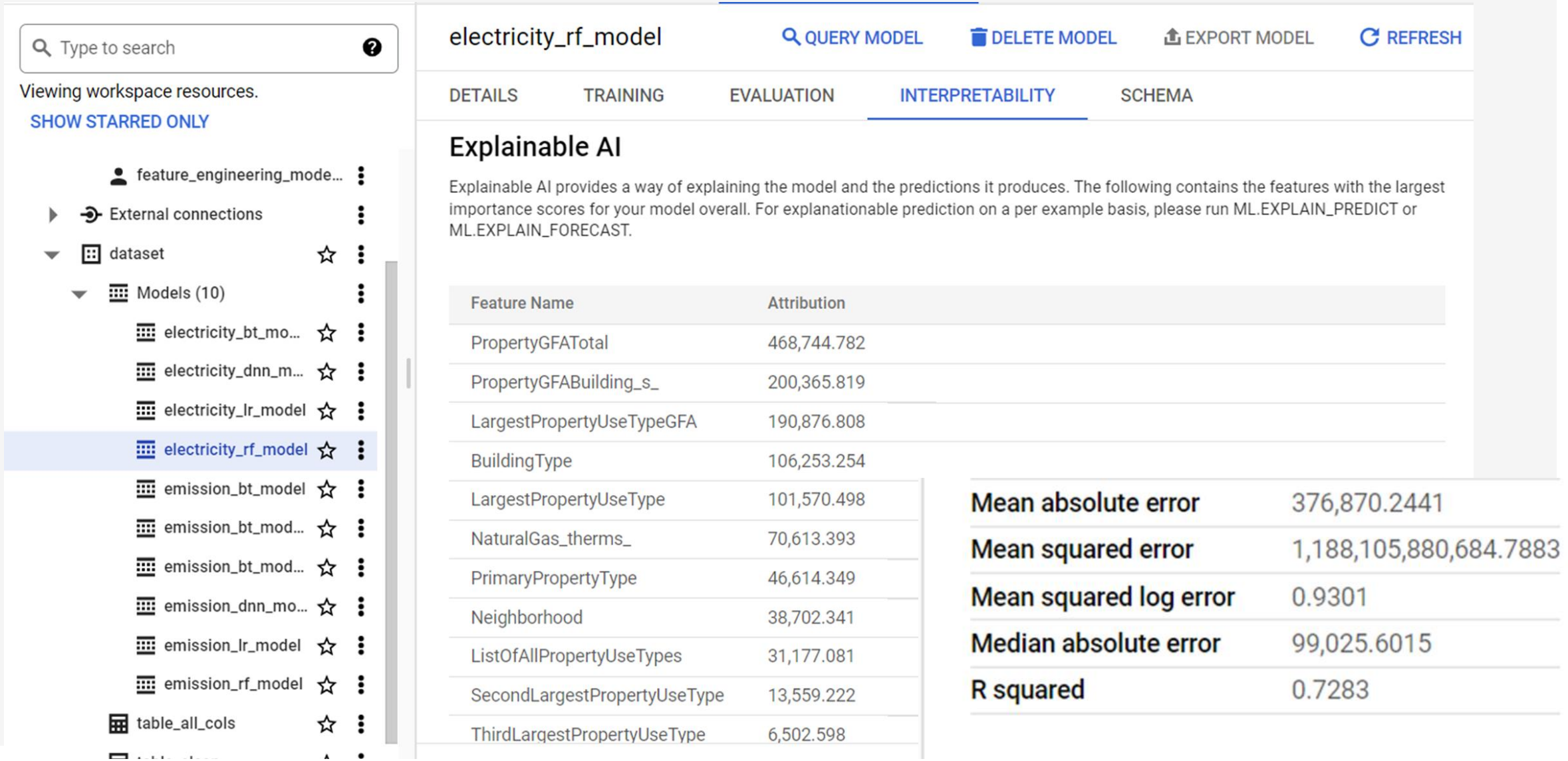


DNN model training (TotalGHGEmissions)

Evaluation and FI (TotalGHGEmissions)



Evaluation and FI (Electricity(kWh))

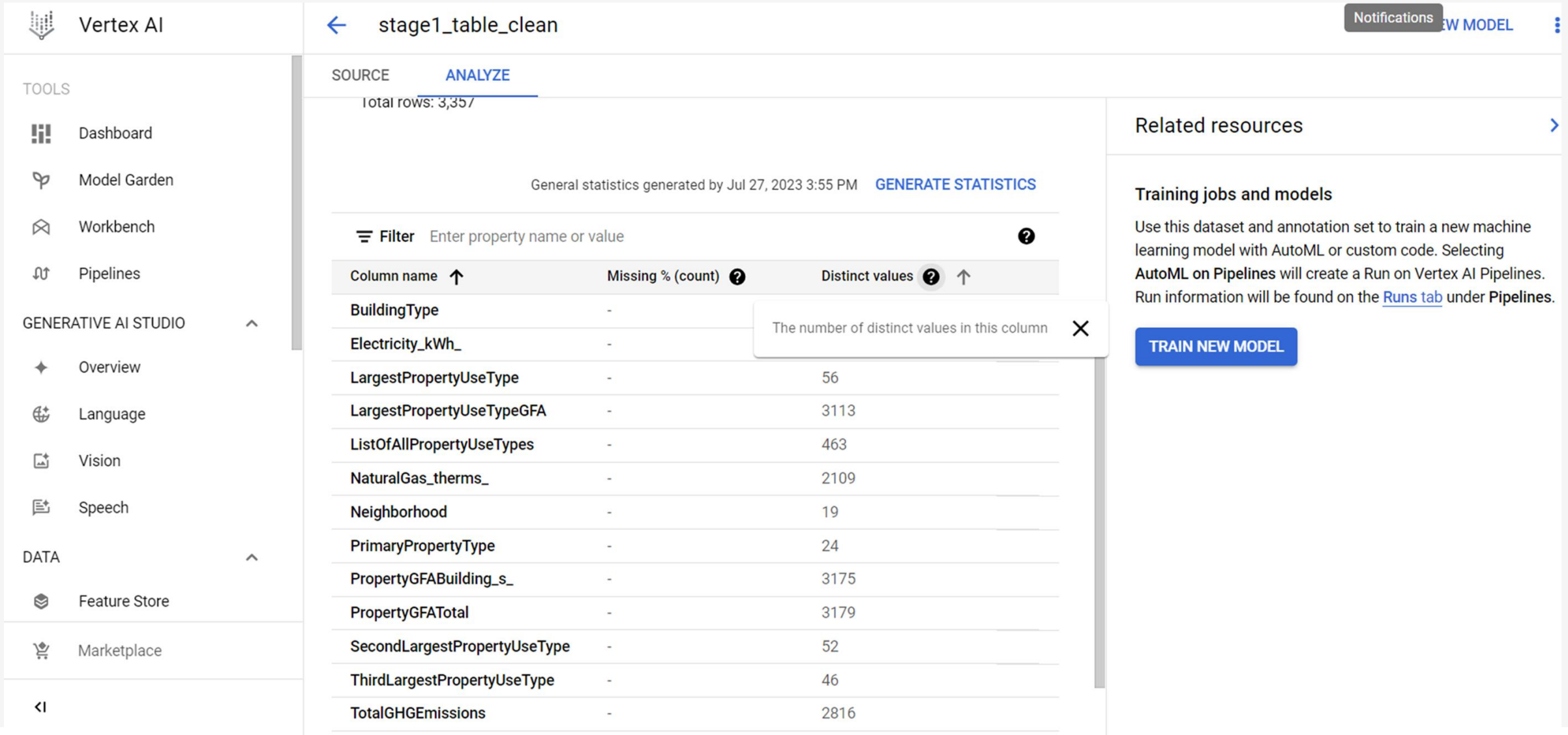


Summary of the results for test data

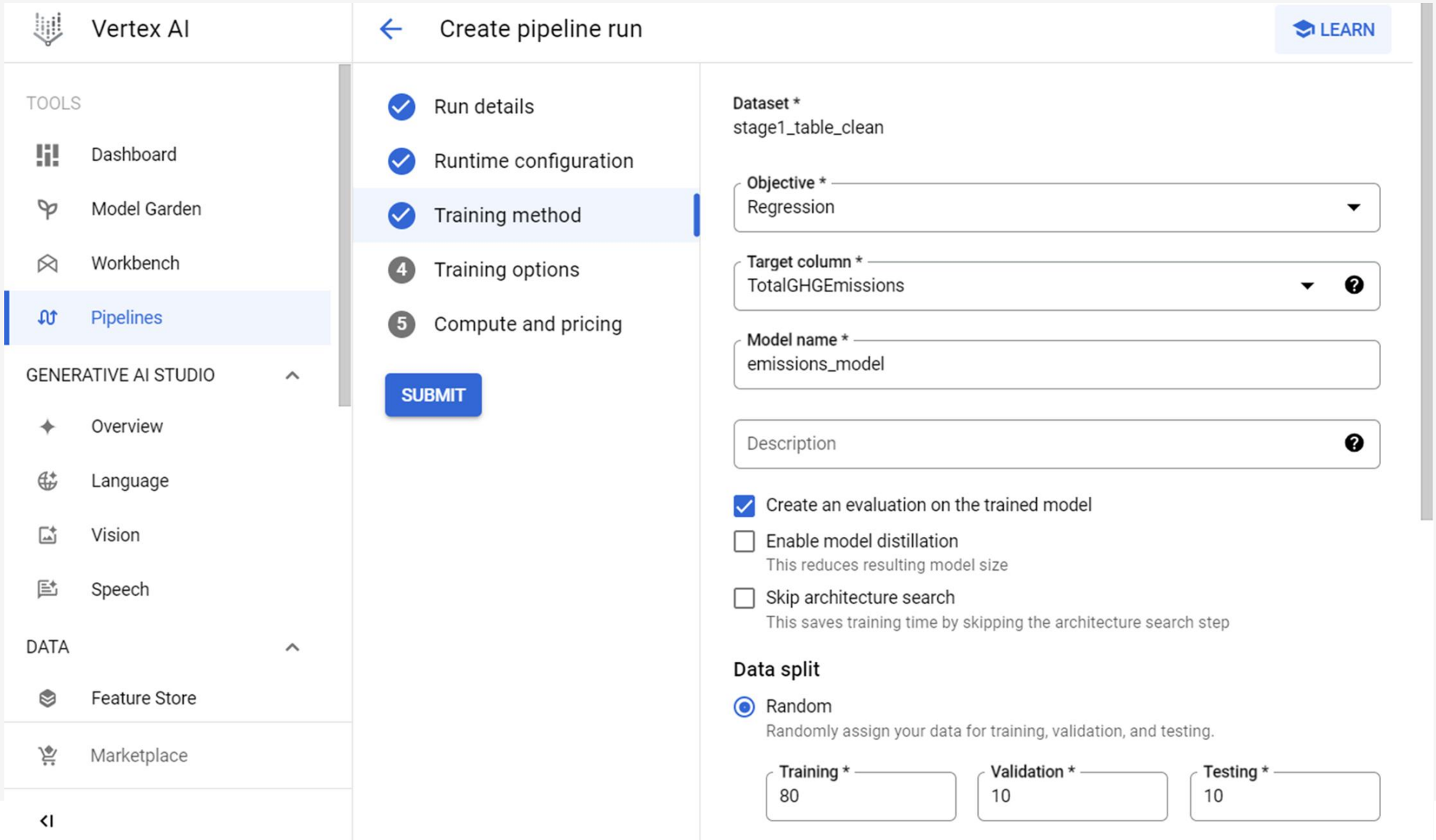
Model / Y	R2	RMSE
Python: Random Forest (TotalGHGEmissions)	0.89	62.3
Python: XGB (Electricity(kWh))	0.88	950 769
BQ ML: DNN (TotalGHGEmissions)	0.94	159
BQ ML: Random Forest (Electricity(kWh))	0.73	1 090 002

Vertex AI Auto ML

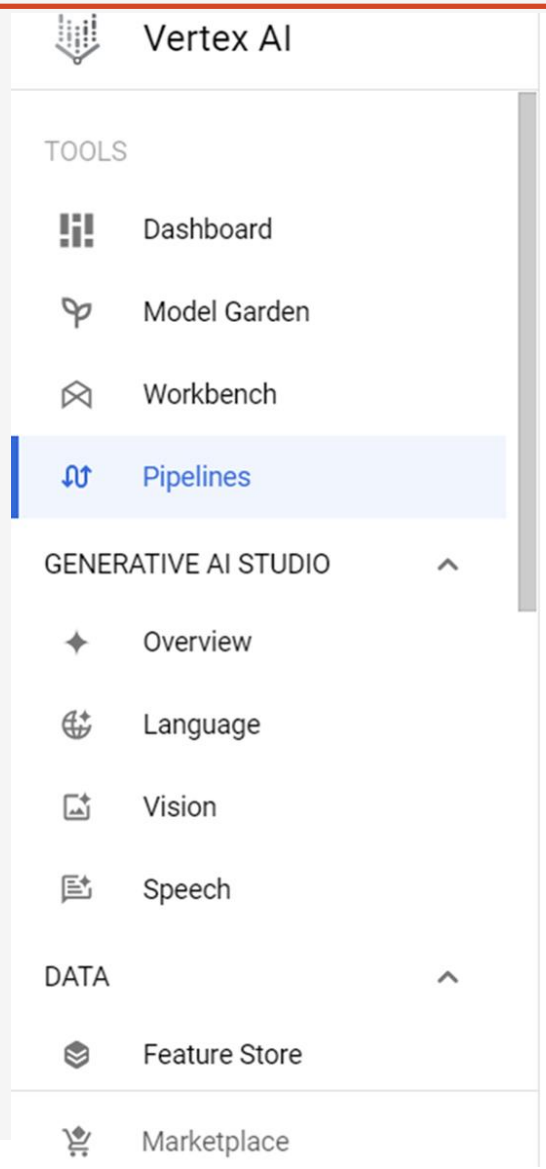
Loading dataset

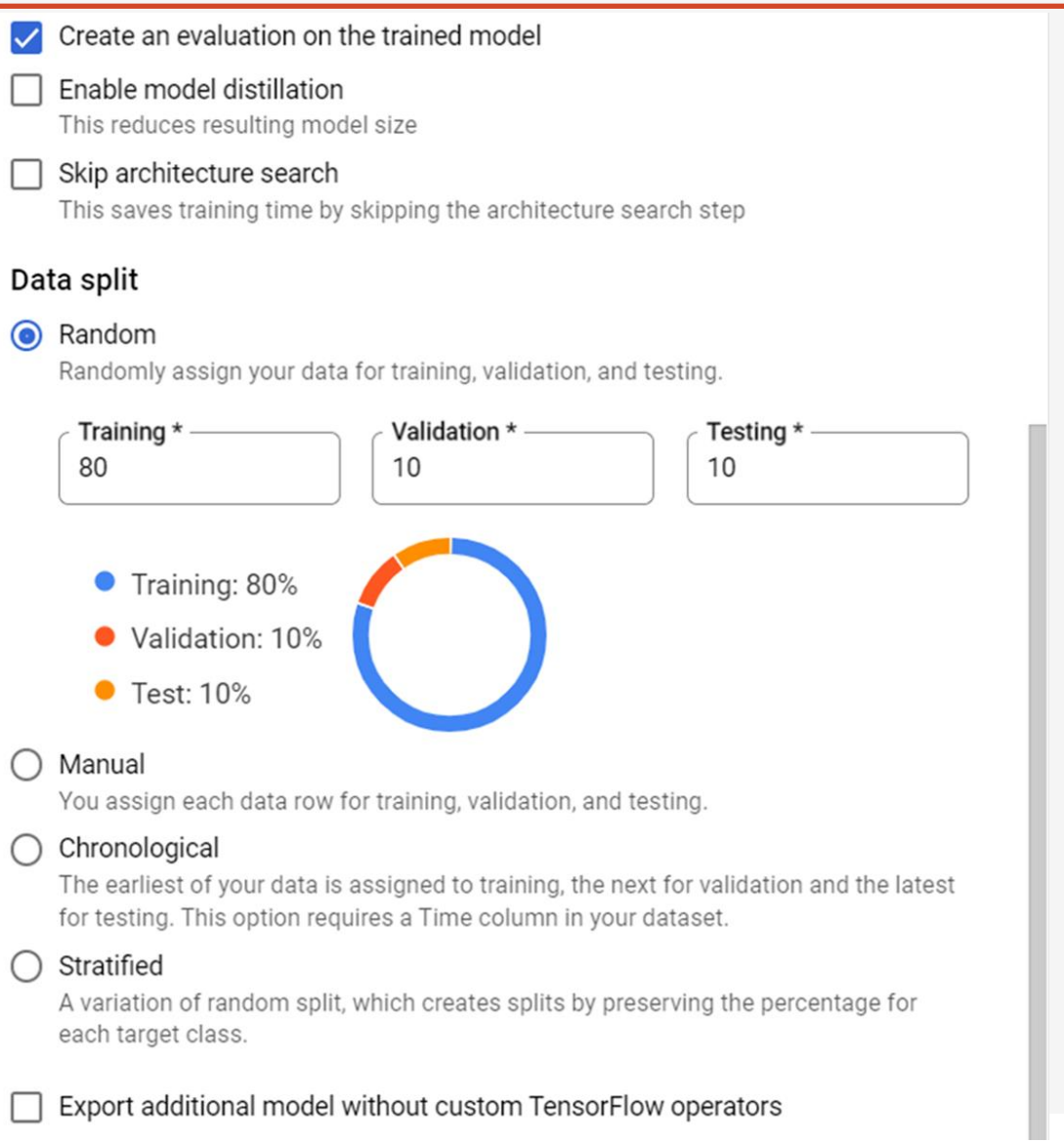


Step by step pipeline: Regression model

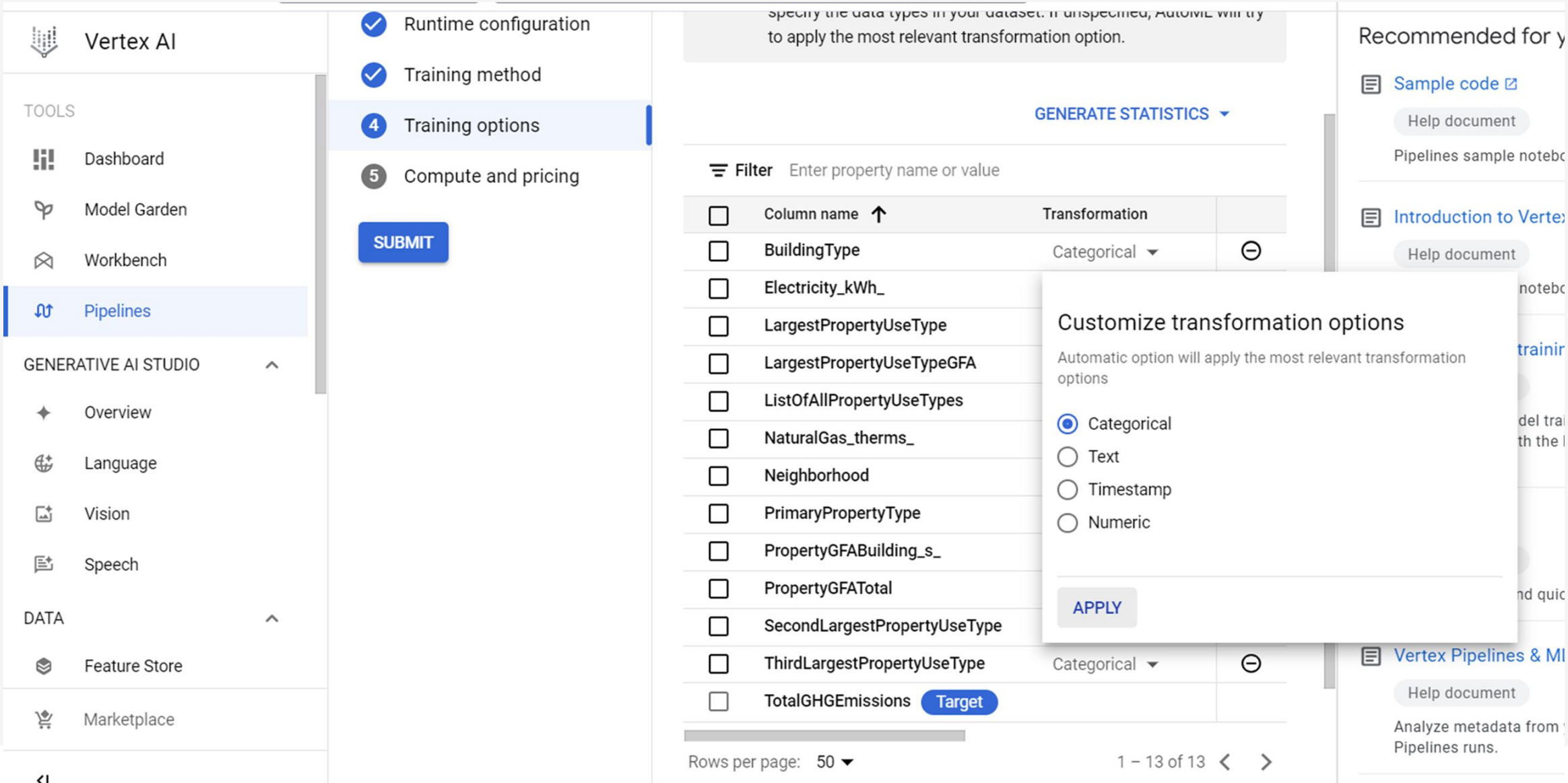


Precise train_test_split

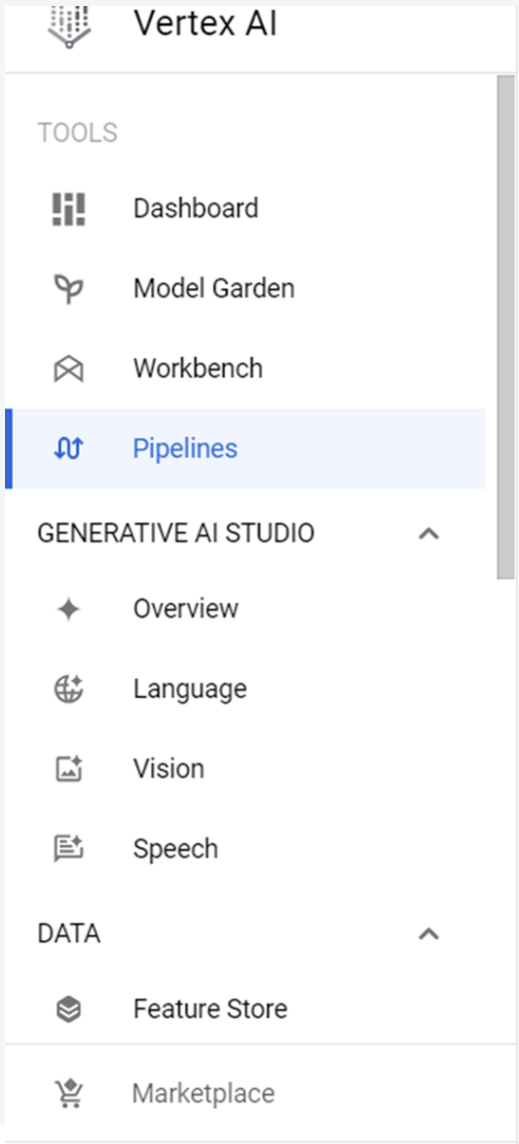




Transform data (change string to categorical)



Define optimization objective



Optimization objective *

- (●) RMSE (Default)
  Capture more extreme values accurately
- ( ) MAE
  View extreme values as outliers with less impact on the model
- ( ) RMSLE
  Penalize error on relative size rather than absolute value. Especially helpful when both predicted and actual values can be quite large. It is undefined when the predicted or ground truth is less than 0.

Architecture search

Number of parallel trials: 35

Cross validation training

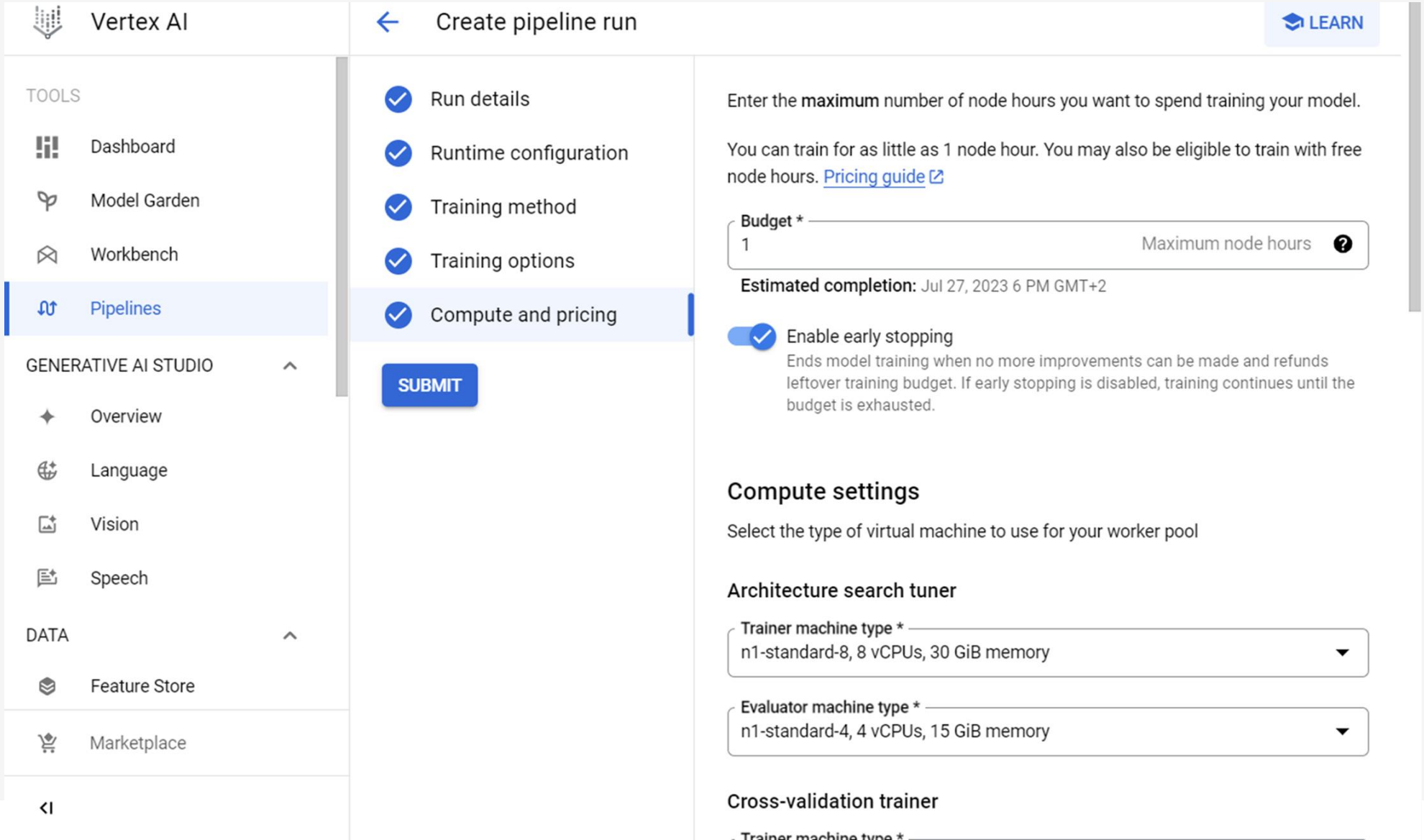
Number of selected trials: 5

Number of parallel trainers: 35

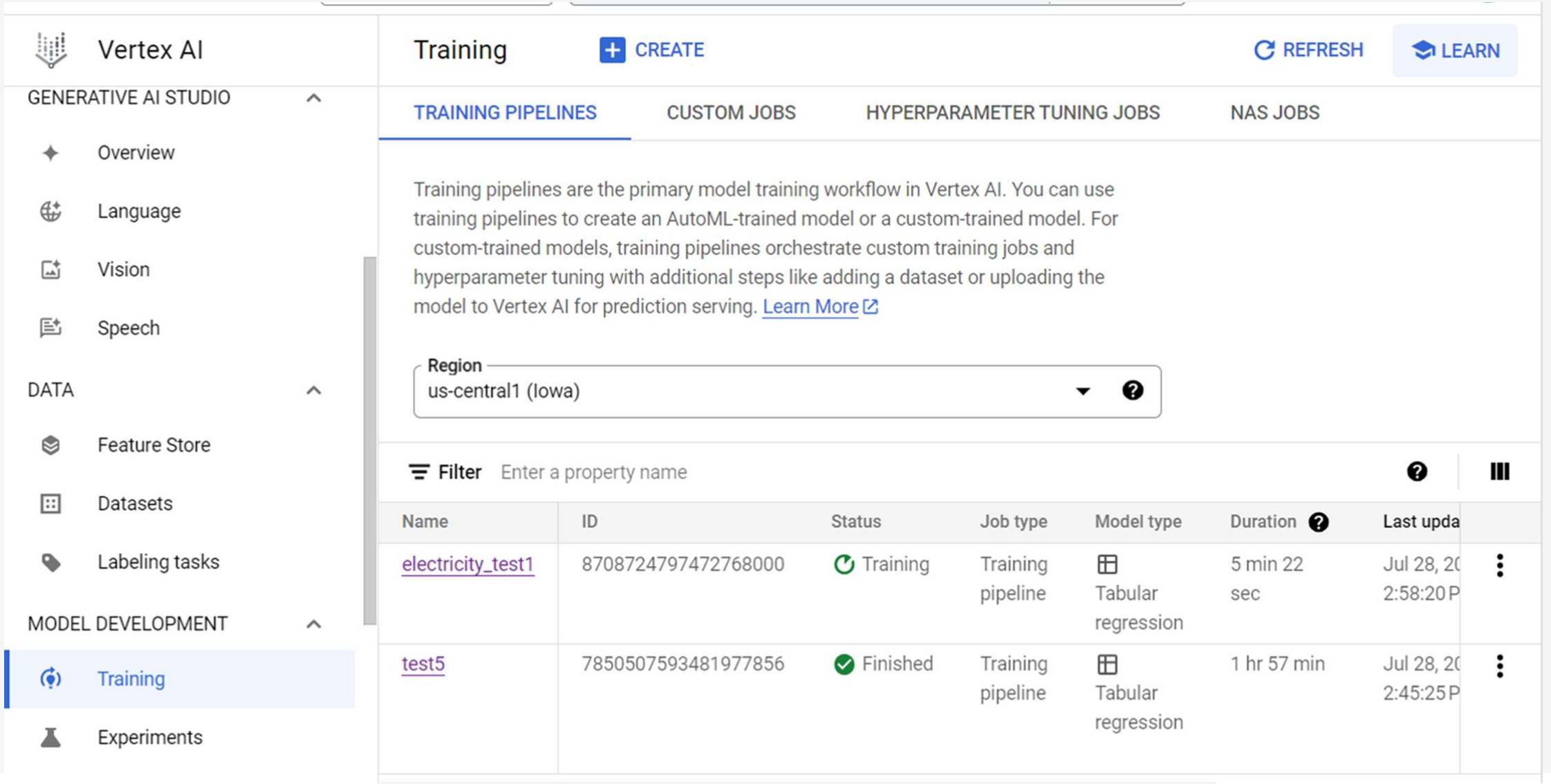
Additional settings

Study spec override (JSON)

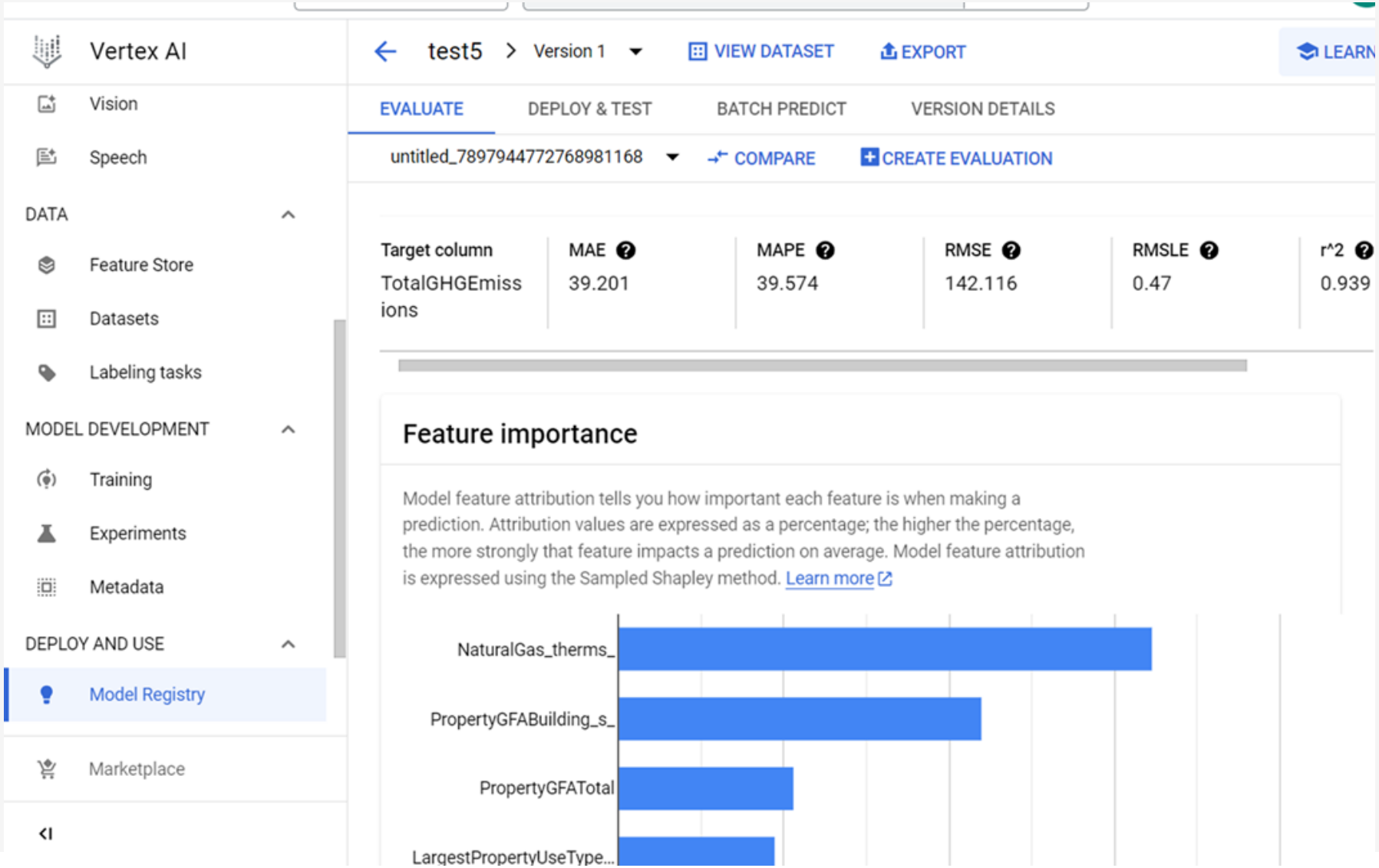
Setting budget and compute type



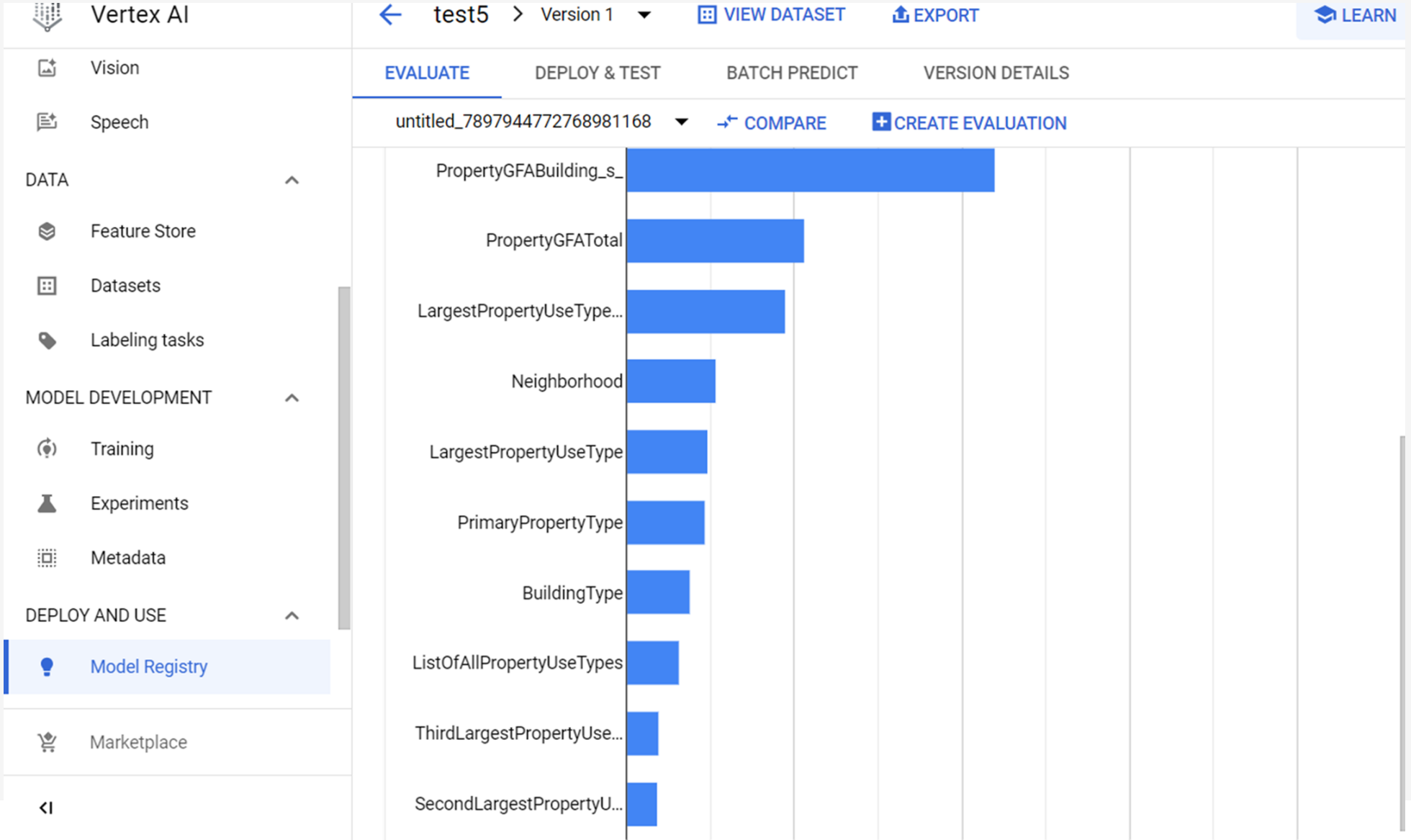
Training process is longlasting



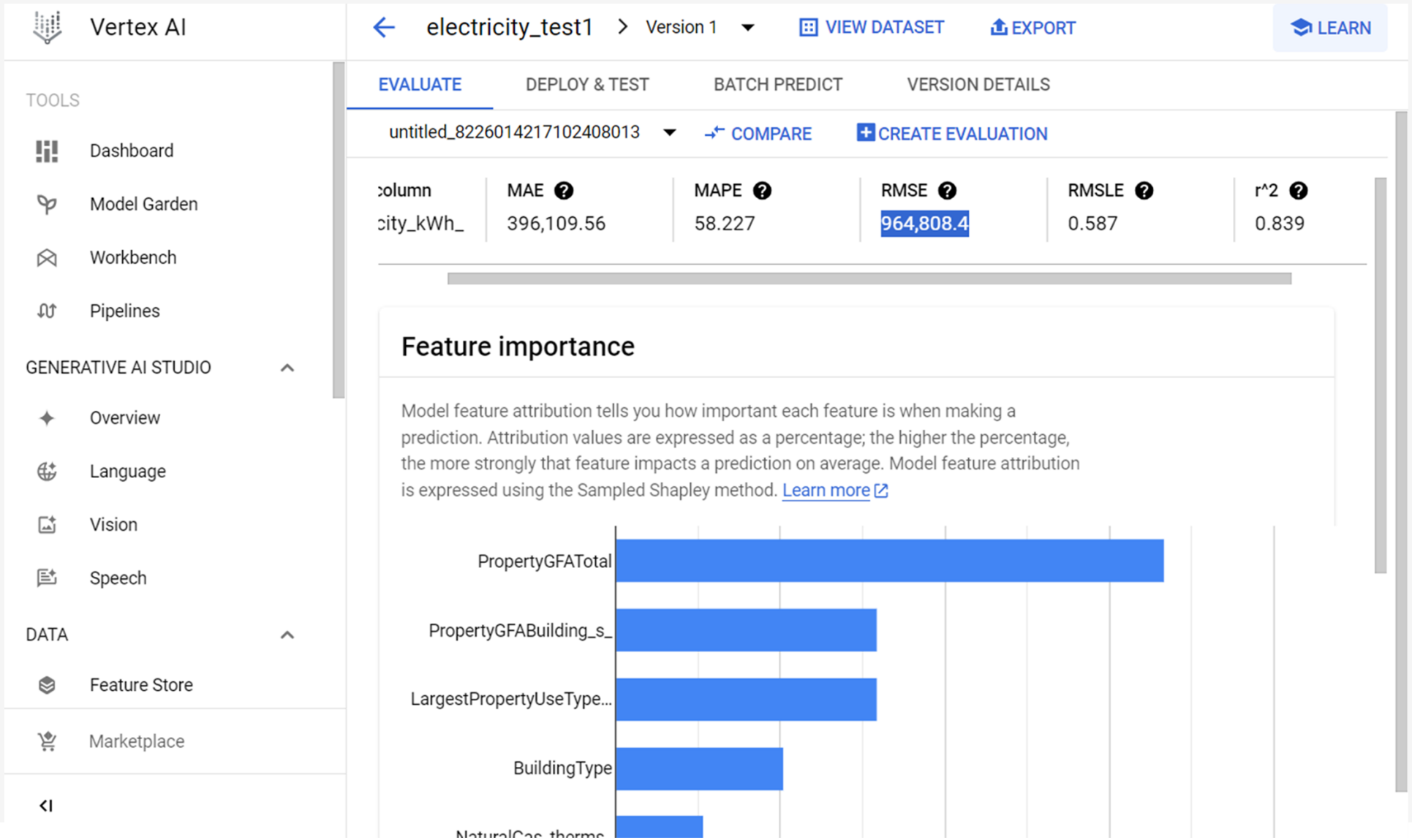
Evaluation (TotalGHGEmissions)



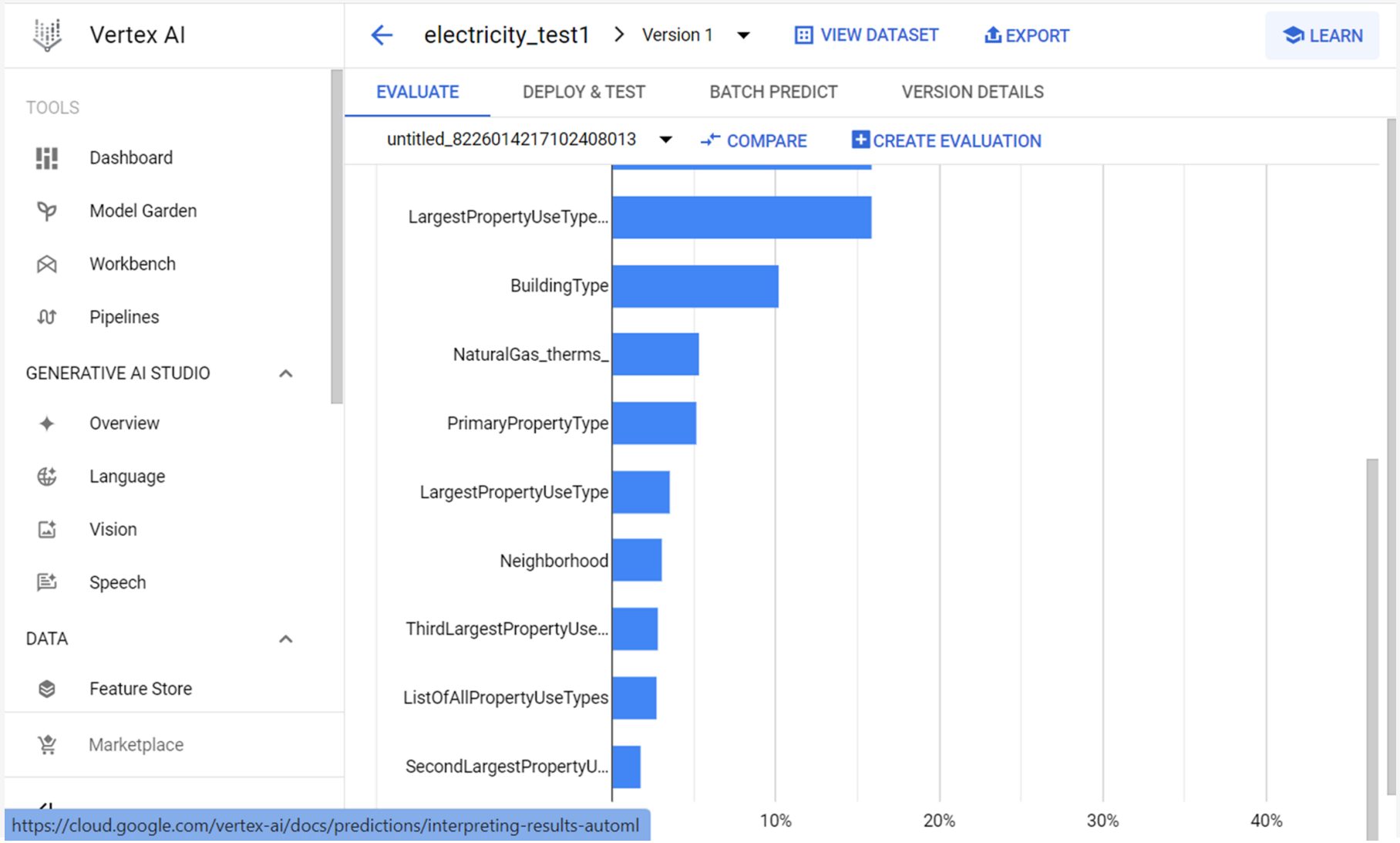
Feature Importance (TotalGHGEmissions)



Evaluation (Electricity(kWh))



Feature Importance (Electricity(kWh))

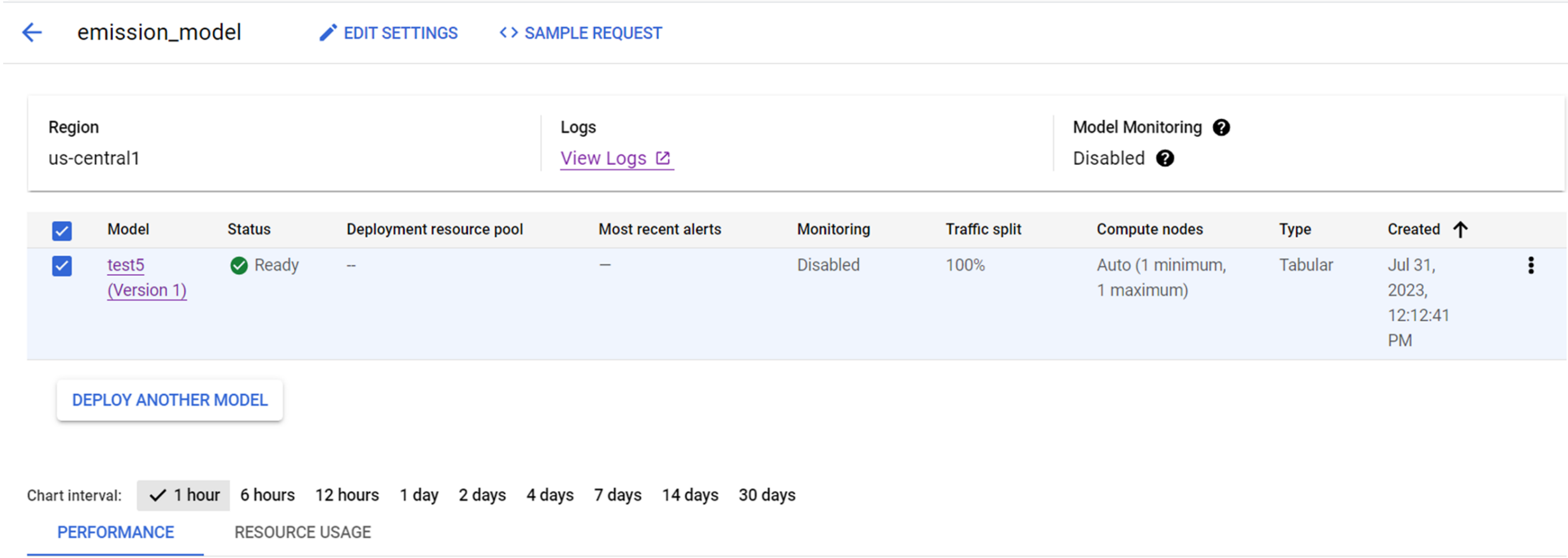


Summary of the results for test data

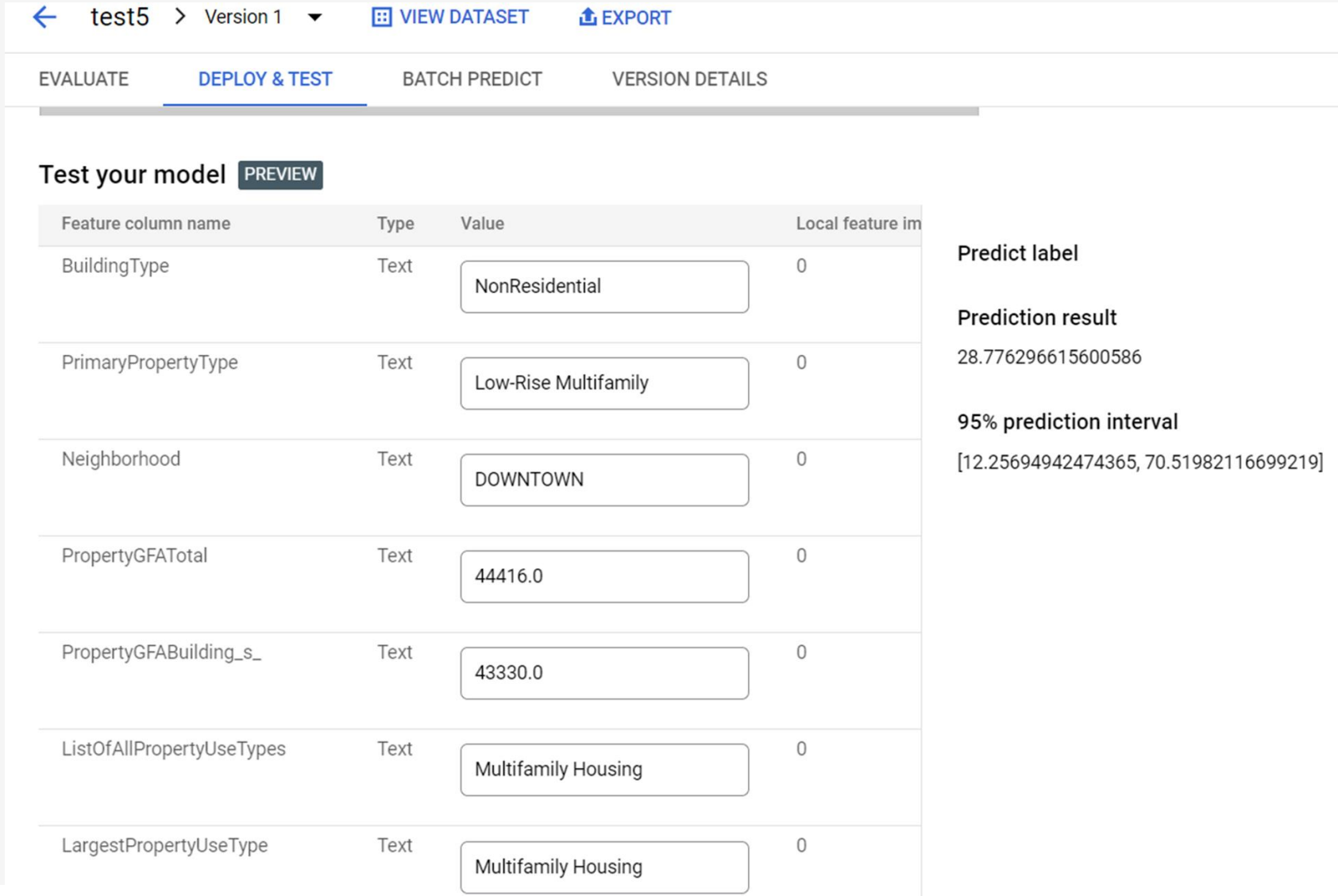
Model / Y	R2	RMSE
Python: Random Forest (TotalGHGEmissions)	0.89	62.3
Python: XGB (Electricity(kWh))	0.88	950 769
BQ ML: DNN (TotalGHGEmissions)	0.94	159
BQ ML: Random Forest (Electricity(kWh))	0.73	1 090 002
Vertex AI Auto ML: Tabular Regression (TotalGHGEmissions)	0.94	142
Vertex AI Auto ML: Tabular Regression (Electricity(kWh))	0.84	964 808

Model Deployment (example)

Deploy the model (placed in Bucket)



Automatic model test



Workbench request to the model

test_endpoint.ipynb

Code | git | Execute | Python (Local)

[1]:
```
from google.cloud import aiplatform
```

[2]:
```
project="733376066474"
endpoint_id="4941851768062476288"
location="us-central1"
endpoint_name=f"projects/{project}/locations/{location}/endpoints/{endpoint_id}"
endpoint=aiplatform.Endpoint(endpoint_name=endpoint_name)
```

[3]:
```
instances=[{'BuildingType':'NonResidential','PrimaryPropertyType':'Low-Rise Multifamily','Neighborhood':'DOWN
           'PropertyGFATotal':'44416.0','PropertyGFABuilding_s_':'43330.0','ListOfAllPropertyUseTypes':'Multi
           'LargestPropertyUseType':'Multifamily Housing','LargestPropertyUseTypeGFA':'39900.0',
            'SecondLargestPropertyUseType':'Parking','ThirdLargestPropertyUseType':'Retail Store',
           'NaturalGas_therms_':'3261.589844'}]
```

[4]:
```
endpoint.predict(instances=instances)
```

[4]:
```
Prediction(predictions=[{'value': 28.77629661560059, 'upper_bound': 70.51982116699219, 'lower_bound': 12.256
94942474365}], deployed_model_id='153388469144518656', model_version_id='1', model_resource_name='projects/7
33376066474/locations/us-central1/models/2837401905662001152', explanations=None)
```