Cloud AutoML Vision Object Detection

REST Resource: projects.locations.models

Resource: Model

API proto representing a trained machine learning model.

JSON representation

JSON representation

```
{
  "name": string,
  "displayName": string,
  "datasetId": string,
  "createTime": string,
  "updateTime": string,
  "deploymentState": enum (DeploymentState (https://cloud.google.com/vision/automl/object-detec

  // Union field model_metadata can be only one of the following:
  "translationModelMetadata": {
    object (TranslationModelMetadata (https://cloud.google.com/vision/automl/object-detection/do
  },
  "imageClassificationModelMetadata": {
    object (ImageClassificationModelMetadata (https://cloud.google.com/vision/automl/object-de
  },
  "textClassificationModelMetadata": {
    object (TextClassificationModelMetadata (https://cloud.google.com/vision/automl/object-dete
  },
  "imageObjectDetectionModelMetadata": {
    object (ImageObjectDetectionModelMetadata (https://cloud.google.com/vision/automl/object-d
  },
  "videoClassificationModelMetadata": {
    object (VideoClassificationModelMetadata (https://cloud.google.com/vision/automl/object-de
  },
  "videoObjectTrackingModelMetadata": {
    object (VideoObjectTrackingModelMetadata (https://cloud.google.com/vision/automl/object-de
  },
  "textExtractionModelMetadata": {
    object (TextExtractionModelMetadata (https://cloud.google.com/vision/automl/object-detectior
  },
  "tablesModelMetadata": {
    object (TablesModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/ref
  },
  "textSentimentModelMetadata": {
    object (TextSentimentModelMetadata (https://cloud.google.com/vision/automl/object-detection/
  }
  // End of list of possible types for union field model_metadata.
}
```

Fields

Fields	
`name`	`string` Output only. Resource name of the model. Format: `projects/{project_id}/locations/{locationId}/models/{modelId}`
`displayName`	`string` Required. The name of the model to show in the interface. The name can be up to 32 characters long and can consist only of ASCII Latin letters A-Z and a-z, underscores (_), and ASCII digits 0-9. It must start with a letter.
`datasetId`	`string` Required. The resource ID of the dataset used to create the model. The dataset must come from the same ancestor project and location.
`createTime`	`string (`Timestamp (https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#google.protobuf.Timestamp) `format)` Output only. Timestamp when the model training finished and can be used for prediction. A timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds. Example: "`2014-10-02T15:01:23.045123456Z`".
`updateTime`	`string (`Timestamp (https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#google.protobuf.Timestamp) `format)` Output only. Timestamp when this model was last updated. A timestamp in RFC3339 UTC "Zulu" format, accurate to nanoseconds. Example: "`2014-10-02T15:01:23.045123456Z`".

Fields	
`deploymentState`	`enum (`DeploymentState (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#DeploymentState)`)` Output only. Deployment state of the model. A model can only serve prediction requests after it gets deployed.
Union field `model_metadata`. Required. The model metadata that is specific to the problem type. Must match the metadata type of the dataset used to train the model. `model_metadata` can be only one of the following:	
`translationModelMetadata`	`object (`TranslationModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#TranslationModelMetadata)`)` Metadata for translation models.
`imageClassificationModelMetadata`	`object (`ImageClassificationModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#ImageClassificationModelMetadata)`)` Metadata for image classification models.
`textClassificationModelMetadata`	`object (`TextClassificationModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#TextClassificationModelMetadata)`)` Metadata for text classification models.
`imageObjectDetectionModelMetadata`	`object (`ImageObjectDetectionModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#ImageObjectDetectionModelMetadata)`)` Metadata for image object detection models.

Fields	
`videoClassificationModelMetadata`	`object (`VideoClassificationModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#VideoClassificationModelMetadata)`)` Metadata for video classification models.
`videoObjectTrackingModelMetadata`	`object (`VideoObjectTrackingModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#VideoObjectTrackingModelMetadata)`)` Metadata for video object tracking models.
`textExtractionModelMetadata`	`object (`TextExtractionModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#TextExtractionModelMetadata)`)` Metadata for text extraction models.
`tablesModelMetadata`	`object (`TablesModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#TablesModelMetadata)`)` Metadata for Tables models.
`textSentimentModelMetadata`	`object (`TextSentimentModelMetadata (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#TextSentimentModelMetadata)`)` Metadata for text sentiment models.

TranslationModelMetadata

Model metadata that is specific to translation.

JSON representation

```
{
  "baseModel": string,
  "sourceLanguageCode": string,
  "targetLanguageCode": string
}
```

Fields

`baseModel`	`string` The resource name of the model to use as a baseline to train the custom model. If unset, we use the default base model provided by Google Translate. Format: `projects/{project_id}/locations/{locationId}/models/{modelId}`
`sourceLanguageCode`	`string` Output only. Inferred from the dataset. The source languge (The BCP-47 language code) that is used for training.
`targetLanguageCode`	`string` Output only. The target languge (The BCP-47 language code) that is used for training.

ImageClassificationModelMetadata

Model metadata for image classification.

JSON representation

JSON representation

```
{
  "baseModelId": string,
  "trainBudget": string,
  "trainCost": string,
  "stopReason": string,
  "modelType": string,
  "nodeQps": number,
  "nodeCount": string,
  "disableEarlyStopping": boolean
}
```

Fields

baseModelId	**string** Optional. The ID of the **base** model. If it is specified, the new model will be created based on the **base** model. Otherwise, the new model will be created from scratch. The **base** model must be in the same **project** and **location** as the new model to create, and have the same **modelType**.
trainBudget	**string (int64** (https://developers.google.com/discovery/v1/type-format) **format)** Required. The train budget of creating this model, expressed in hours. The actual **trainCost** will be equal or less than this value.
trainCost	**string (int64** (https://developers.google.com/discovery/v1/type-format) **format)** Output only. The actual train cost of creating this model, expressed in hours. If this model is created from a **base** model, the train cost used to create the **base** model are not included.
stopReason	**string** Output only. The reason that this create model operation stopped, e.g. **BUDGET_REACHED**, **MODEL_CONVERGED**.
modelType	**string** Optional. Type of the model. The available values are: * **cloud** - Model to be used via prediction calls to AutoML API. This is the default value. * **mobile-low-latency-1** - A model that, in addition to providing prediction via AutoML API, can also be exported (see

Fields

`AutoMl.ExportModel` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/export#google.cloud.automl.v1beta1.AutoMl.ExportModel)) and used on a mobile or edge device with TensorFlow afterwards. Expected to have low latency, but may have lower prediction quality than other models. * **`mobile-versatile-1`** - A model that, in addition to providing prediction via AutoML API, can also be exported (see `AutoMl.ExportModel` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/export#google.cloud.automl.v1beta1.AutoMl.ExportModel)) and used on a mobile or edge device with TensorFlow afterwards. * **`mobile-high-accuracy-1`** - A model that, in addition to providing prediction via AutoML API, can also be exported (see `AutoMl.ExportModel` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/export#google.cloud.automl.v1beta1.AutoMl.ExportModel)) and used on a mobile or edge device with TensorFlow afterwards. Expected to have a higher latency, but should also have a higher prediction quality than other models. * **`mobile-core-ml-low-latency-1`** - A model that, in addition to providing prediction via AutoML API, can also be exported (see `AutoMl.ExportModel` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/export#google.cloud.automl.v1beta1.AutoMl.ExportModel)) and used on a mobile device with Core ML afterwards. Expected to have low latency, but may have lower prediction quality than other models. * **`mobile-core-ml-versatile-1`** - A model that, in addition to providing prediction via AutoML API, can also be exported (see `AutoMl.ExportModel` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/export#google.cloud.automl.v1beta1.AutoMl.ExportModel)) and used on a mobile device with Core ML afterwards. * **`mobile-core-ml-high-accuracy-1`** - A model that, in addition to providing prediction via AutoML API, can also be exported (see `AutoMl.ExportModel` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/export#google.cloud.automl.v1beta1.AutoMl.ExportModel)) and used on a mobile device with Core ML afterwards. Expected to have

Fields	
	a higher latency, but should also have a higher prediction quality than other models.
`nodeQps`	`number` Output only. An approximate number of online prediction QPS that can be supported by this model per each node on which it is deployed.
`nodeCount`	`string (int64` (https://developers.google.com/discovery/v1/type-format) `format)` Output only. The number of nodes this model is deployed on. A node is an abstraction of a machine resource, which can handle online prediction QPS as given in the nodeQps field.
`disableEarlyStopping`	`boolean` Use the entire training budget. This disables the early stopping feature. By default, the early stopping feature is enabled, which means that AutoML Image Classification might stop training before the entire training budget has been used.

TextClassificationModelMetadata

Model metadata that is specific to text classification.

ImageObjectDetectionModelMetadata

Model metadata specific to image object detection.

JSON representation

JSON representation

```
{
  "modelType": string,
  "nodeCount": string,
  "nodeQps": number,
  "stopReason": string,
  "trainBudgetMilliNodeHours": string,
  "trainCostMilliNodeHours": string,
  "disableEarlyStopping": boolean
}
```

Fields

modelType	**string** Optional. Type of the model. The available values are: * **cloud-high-accuracy-1** - (default) A model to be used via prediction calls to AutoML API. Expected to have a higher latency, but should also have a higher prediction quality than other models. * **cloud-low-latency-1** - A model to be used via prediction calls to AutoML API. Expected to have low latency, but may have lower prediction quality than other models.
nodeCount	**string (int64** (https://developers.google.com/discovery/v1/type-format) **format)** Output only. The number of nodes this model is deployed on. A node is an abstraction of a machine resource, which can handle online prediction QPS as given in the qps_per_node field.
nodeQps	**number** Output only. An approximate number of online prediction QPS that can be supported by this model per each node on which it is deployed.
stopReason	**string** Output only. The reason that this create model operation stopped, e.g. **BUDGET_REACHED**, **MODEL_CONVERGED**.

Fields	
`trainBudgetMilliNodeHours`	`string (int64` (https://developers.google.com/discovery/v1/type-format) `format)` The train budget of creating this model, expressed in milli node hours i.e. 1,000 value in this field means 1 node hour. The actual `trainCost` will be equal or less than this value. If further model training ceases to provide any improvements, it will stop without using full budget and the stopReason will be `MODEL_CONVERGED`. Note, node_hour = actual_hour * number_of_nodes_invovled. For model type `cloud-high-accuracy-1`(default) and `cloud-low-latency-1`, the train budget must be between 20,000 and 2,000,000 milli node hours, inclusive. The default value is 216, 000 which represents one day in wall time. For model type `mobile-low-latency-1`, `mobile-versatile-1`, `mobile-high-accuracy-1`, `mobile-core-ml-low-latency-1`, `mobile-core-ml-versatile-1`, `mobile-core-ml-high-accuracy-1`, the train budget must be between 1,000 and 100,000 milli node hours, inclusive. The default value is 24, 000 which represents one day in wall time.
`trainCostMilliNodeHours`	`string (int64` (https://developers.google.com/discovery/v1/type-format) `format)` Output only. The actual train cost of creating this model, expressed in milli node hours, i.e. 1,000 value in this field means 1 node hour. Guaranteed to not exceed the train budget.
`disableEarlyStopping`	`boolean` Use the entire training budget. This disables the early stopping feature. By default, the early stopping feature is enabled, which means that AutoML Image Object Detection might stop training before the entire training budget has been used.

VideoClassificationModelMetadata

Model metadata specific to video classification.

VideoObjectTrackingModelMetadata

Model metadata specific to video object tracking.

TextExtractionModelMetadata

Model metadata that is specific to text extraction.

TablesModelMetadata

Model metadata specific to AutoML Tables.

JSON representation

```
{
  "targetColumnSpec": {
    object (ColumnSpec (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/
  },
  "inputFeatureColumnSpecs": [
    {
      object (ColumnSpec (https://cloud.google.com/vision/automl/object-detection/docs/reference/re
    }
  ],
  "optimizationObjective": string,
  "tablesModelColumnInfo": [
    {
      object (TablesModelColumnInfo (https://cloud.google.com/vision/automl/object-detection/doc
    }
  ],
  "trainBudgetMilliNodeHours": string,
  "trainCostMilliNodeHours": string,
  "disableEarlyStopping": boolean
}
```

Fields

Fields	
`targetColumnSpec`	`object (`ColumnSpec (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.datasets.tableSpecs.columnSpecs#ColumnSpec)`)` Column spec of the dataset's primary table's column the model is predicting. Snapshotted when model creation started. Only 3 fields are used: name - May be set on models.create, if it's not then the ColumnSpec corresponding to the current targetColumnSpecId of the dataset the model is trained from is used. If neither is set, models.create will error. displayName - Output only. dataType - Output only.

Fields	
`inputFeatureColumnSpecs[]`	`object (`ColumnSpec (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.datasets.tableSpecs.columnSpecs#ColumnSpec)`)` Column specs of the dataset's primary table's columns, on which the model is trained and which are used as the input for predictions. The `target_column` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#TablesModelMetadata.FIELDS.target_column_spec) as well as, according to dataset's state upon model creation, `weight_column` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.datasets#TablesDatasetMetadata.FIELDS.weight_column_spec_id) , and `ml_use_column` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.datasets#TablesDatasetMetadata.FIELDS.ml_use_column_spec_id) must never be included here. Only 3 fields are used: • name - May be set on models.create, if set only the columns specified are used, otherwise all primary table's columns (except the ones listed above) are used for the training and prediction input. • displayName - Output only. • dataType - Output only.

Fields	
`optimizationObjective`	`string` Objective function the model is optimizing towards. The training process creates a model that maximizes/minimizes the value of the objective function over the validation set. The supported optimization objectives depend on the prediction type. If the field is not set, a default objective function is used. CLASSIFICATION_BINARY: "MAXIMIZE_AU_ROC" (default) - Maximize the area under the receiver operating characteristic (ROC) curve. "MINIMIZE_LOG_LOSS" - Minimize log loss. "MAXIMIZE_AU_PRC" - Maximize the area under the precision-recall curve. "MAXIMIZE_PRECISION_AT_RECALL" - Maximize precision for a specified recall value. "MAXIMIZE_RECALL_AT_PRECISION" - Maximize recall for a specified precision value. CLASSIFICATION_MULTI_CLASS : "MINIMIZE_LOG_LOSS" (default) - Minimize log loss. REGRESSION: "MINIMIZE_RMSE" (default) - Minimize root-mean-squared error (RMSE). "MINIMIZE_MAE" - Minimize mean-absolute error (MAE). "MINIMIZE_RMSLE" - Minimize root-mean-squared log error (RMSLE).
`tablesModelColumnInfo[]`	`object (`TablesModelColumnInfo (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models#TablesModelColumnInfo) `)` Output only. Auxiliary information for each of the inputFeatureColumnSpecs with respect to this particular model.

Fields	
`trainBudgetMilliNodeHours`	`string (int64` (https://developers.google.com/discovery/v1/type-format) `format)` Required. The train budget of creating this model, expressed in milli node hours i.e. 1,000 value in this field means 1 node hour. The training cost of the model will not exceed this budget. The final cost will be attempted to be close to the budget, though may end up being (even) noticeably smaller - at the backend's discretion. This especially may happen when further model training ceases to provide any improvements. If the budget is set to a value known to be insufficient to train a model for the given dataset, the training won't be attempted and will error. The train budget must be between 1,000 and 72,000 milli node hours, inclusive.
`trainCostMilliNodeHours`	`string (int64` (https://developers.google.com/discovery/v1/type-format) `format)` Output only. The actual training cost of the model, expressed in milli node hours, i.e. 1,000 value in this field means 1 node hour. Guaranteed to not exceed the train budget.
`disableEarlyStopping`	`boolean` Use the entire training budget. This disables the early stopping feature. By default, the early stopping feature is enabled, which means that AutoML Tables might stop training before the entire training budget has been used.

TablesModelColumnInfo

An information specific to given column and Tables Model, in context of the Model and the predictions created by it.

JSON representation

JSON representation

```
{
  "columnSpecName": string,
  "columnDisplayName": string,
  "featureImportance": number
}
```

Fields

`columnSpecName`	`string` Output only. The name of the ColumnSpec describing the column. Not populated when this proto is outputted to BigQuery.
`columnDisplayName`	`string` Output only. The display name of the column (same as the displayName of its ColumnSpec).
`featureImportance`	`number` Output only. When given as part of a Model (always populated): Measurement of how much model predictions correctness on the TEST data depend on values in this column. A value between 0 and 1, higher means higher influence. These values are normalized - for all input feature columns of a given model they add to 1. When given back by models.predict (populated iff `featureImportance` param (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/predict#body.request_body.FIELDS.params) is set) or Batch models.predict (populated iff `featureImportance` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/predict#body.request_body.FIELDS.params) param is set): Measurement of how impactful for the prediction returned for the given row the value in this column was. A value between 0 and 1, higher means larger impact. These values are normalized - for all input feature columns of a single predicted row they add to 1.

TextSentimentModelMetadata

Model metadata that is specific to text sentiment.

DeploymentState

Deployment state of the model.

Enums	
`DEPLOYMENT_STATE_UNSPECIFIED`	Should not be used, an un-set enum has this value by default.
`DEPLOYED`	Model is deployed.
`UNDEPLOYED`	Model is not deployed.

Methods	
`batchPredict` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/batchPredict)	Perform a batch prediction.
`create` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/create)	Creates a model.
`delete` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/delete)	Deletes a model.
`deploy` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/deploy)	Deploys a model.

Methods	
`export` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/export)	Exports a trained, "export-able", model to a user specified Google Cloud Storage location.
`exportEvaluatedExamples` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/exportEvaluatedExamples)	Exports examples on which the model was evaluated (i.e.
`get` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/get)	Gets a model.
`getIamPolicy` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/getIamPolicy)	Gets the access control policy for a resource.
`list` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/list)	Lists models.
`predict` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/predict)	Perform an online prediction.
`setIamPolicy` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/setIamPolicy)	Sets the access control policy on the specified resource.
`undeploy` (https://cloud.google.com/vision/automl/object-detection/docs/reference/rest/v1beta1/projects.locations.models/undeploy)	Undeploys a model.

Last updated October 14, 2019.