Models

A Core ML model consists of a specification version and a model description, and can be any one of the following types:

Neural Networks

- NeuralNetwork

Regressors

- NeuralNetworkRegressor
- TreeEnsembleRegressor
- GLMRegressor
- SupportVectorRegressor

Classifiers

- NeuralNetworkClassifier
- TreeEnsembleClassifier
- GLMClassifier
- SupportVectorClassifier

Feature Engineering

- Imputer
- Scaler
- Normalizer
- OneHotEncoder
- CategoricalMapping
- FeatureVectorizer
- DictVectorizer

Pipeline

A pipeline consisting of one or more models.

```
message Pipeline {
    repeated Model models = 1;
}
```

PipelineClassifier

A classifier pipeline.

```
message PipelineClassifier {
    Pipeline pipeline = 1;
}
```

PipelineRegressor

A regressor pipeline.

```
message PipelineRegressor {
    Pipeline pipeline = 1;
}
```

FeatureDescription

A feature description, consisting of a name, short description, and type.

```
message FeatureDescription {
    string name = 1;
    string shortDescription = 2;
    FeatureType type = 3;
}
```

Metadata

Model metadata, consisting of a short description, a version string, an author, a license, and any other user defined key/value meta data.

```
message Metadata {
    string shortDescription = 1;
    string versionString = 2;
    string author = 3;
    string license = 4;
    map<string, string> userDefined = 100;
}
```

ModelDescription

A description of a model, consisting of descriptions of its input and output features. Both regressor and classifier models require the name of the primary predicted output feature (`predictedFeatureName`). Classifier models can specify the output feature containing probabilities for the predicted classes (`predictedProbabilitiesName`).

```
message ModelDescription {
    repeated FeatureDescription input = 1;
    repeated FeatureDescription output = 10;

    string predictedFeatureName = 11;
    string predictedProbabilitiesName = 12;

    Metadata metadata = 100;
}
```

Model

A Core ML model, consisting of a specification version, a model description, and a model type.

Core ML model compatibility is indicated by a monotonically increasing specification version number, which is incremented anytime a backward-incompatible change is made (this is functionally equivalent to the MAJOR version number described by Semantic Versioning 2.0.0). The Core ML framework in macOS currently supports specification version `1` .

```
message Model {
    int32 specificationVersion = 1;
    ModelDescription description = 2;

    // start at 200 here
    // model specific parameters:
    oneof Type {
        // pipeline starts at 200
        PipelineClassifier pipelineClassifier = 200;
        PipelineRegressor pipelineRegressor = 201;
        Pipeline pipeline = 202;

        // regressors start at 300
        GLMRegressor glmRegressor = 300;
        SupportVectorRegressor supportVectorRegressor = 301;
        TreeEnsembleRegressor treeEnsembleRegressor = 302;
        NeuralNetworkRegressor neuralNetworkRegressor = 303;

        // classifiers start at 400
        GLMClassifier glmClassifier = 400;
        SupportVectorClassifier supportVectorClassifier = 401;
        TreeEnsembleClassifier treeEnsembleClassifier = 402;
        NeuralNetworkClassifier neuralNetworkClassifier = 403;

        // generic models start at 500
        NeuralNetwork neuralNetwork = 500;

        // feature engineering starts at 600
        OneHotEncoder oneHotEncoder = 600;
        Imputer imputer = 601;
        FeatureVectorizer featureVectorizer = 602;
        DictVectorizer dictVectorizer = 603;
        Scaler scaler = 604;
        CategoricalMapping categoricalMapping = 606;
        Normalizer normalizer = 607;
        ArrayFeatureExtractor arrayFeatureExtractor = 609;

        // simple mathematical functions used for testing start at 900
        Identity identity = 900;

        // reserved until 1000
    }
}
```

Identity

An identity model.

This model returns given inputs as outputs, unchanged. Intended to be used for testing purposes.

```
message Identity {
}
```