use serde::{Deserialize, Serialize};

use serde\_json::Value;

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct Root {

#[serde(rename = "common\_prefix")]

pub common\_prefix: String,

#[serde(rename = "densification\_transform\_spec")]

pub densification\_transform\_spec: DensificationTransformSpec,

#[serde(rename = "identity\_transform\_spec")]

pub identity\_transform\_spec: Vec<IdentityTransformSpec>,

#[serde(rename = "complex\_feature\_type\_transform\_spec")]

pub complex\_feature\_type\_transform\_spec: Vec<ComplexFeatureTypeTransformSpec>,

#[serde(rename = "input\_features\_map")]

pub input\_features\_map: Value,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct DensificationTransformSpec {

pub discrete: Discrete,

pub cont: Cont,

pub binary: Binary,

pub string: Value, // Use StringType

pub blob: Blob,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct Discrete {

pub tag: String,

#[serde(rename = "generic\_feature\_type")]

pub generic\_feature\_type: i64,

#[serde(rename = "feature\_identifier")]

pub feature\_identifier: String,

#[serde(rename = "fixed\_length")]

pub fixed\_length: i64,

#[serde(rename = "default\_value")]

pub default\_value: DefaultValue,

#[serde(rename = "input\_features")]

pub input\_features: Vec<InputFeature>,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct DefaultValue {

#[serde(rename = "type")]

pub type\_field: String,

pub value: String,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct InputFeature {

#[serde(rename = "feature\_id")]

pub feature\_id: i64,

#[serde(rename = "full\_feature\_name")]

pub full\_feature\_name: String,

#[serde(rename = "feature\_type")]

pub feature\_type: i64,

pub index: i64,

#[serde(rename = "maybe\_exclude")]

pub maybe\_exclude: bool,

pub tag: String,

#[serde(rename = "added\_at")]

pub added\_at: i64,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct Cont {

pub tag: String,

#[serde(rename = "generic\_feature\_type")]

pub generic\_feature\_type: i64,

#[serde(rename = "feature\_identifier")]

pub feature\_identifier: String,

#[serde(rename = "fixed\_length")]

pub fixed\_length: i64,

#[serde(rename = "default\_value")]

pub default\_value: DefaultValue,

#[serde(rename = "input\_features")]

pub input\_features: Vec<InputFeature>,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct Binary {

pub tag: String,

#[serde(rename = "generic\_feature\_type")]

pub generic\_feature\_type: i64,

#[serde(rename = "feature\_identifier")]

pub feature\_identifier: String,

#[serde(rename = "fixed\_length")]

pub fixed\_length: i64,

#[serde(rename = "default\_value")]

pub default\_value: DefaultValue,

#[serde(rename = "input\_features")]

pub input\_features: Vec<InputFeature>,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct StringType {

pub tag: String,

#[serde(rename = "generic\_feature\_type")]

pub generic\_feature\_type: i64,

#[serde(rename = "feature\_identifier")]

pub feature\_identifier: String,

#[serde(rename = "fixed\_length")]

pub fixed\_length: i64,

#[serde(rename = "default\_value")]

pub default\_value: DefaultValue,

#[serde(rename = "input\_features")]

pub input\_features: Vec<InputFeature>,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct Blob {

pub tag: String,

#[serde(rename = "generic\_feature\_type")]

pub generic\_feature\_type: i64,

#[serde(rename = "feature\_identifier")]

pub feature\_identifier: String,

#[serde(rename = "fixed\_length")]

pub fixed\_length: i64,

#[serde(rename = "default\_value")]

pub default\_value: DefaultValue,

#[serde(rename = "input\_features")]

pub input\_features: Vec<Value>,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct IdentityTransformSpec {

#[serde(rename = "feature\_id")]

pub feature\_id: i64,

#[serde(rename = "full\_feature\_name")]

pub full\_feature\_name: String,

#[serde(rename = "feature\_type")]

pub feature\_type: i64,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct ComplexFeatureTypeTransformSpec {

#[serde(rename = "feature\_id")]

pub feature\_id: i64,

#[serde(rename = "full\_feature\_name")]

pub full\_feature\_name: String,

#[serde(rename = "feature\_type")]

pub feature\_type: i64,

pub index: i64,

#[serde(rename = "maybe\_exclude")]

pub maybe\_exclude: bool,

pub tag: String,

#[serde(rename = "tensor\_data\_type")]

pub tensor\_data\_type: Option<i64>,

#[serde(rename = "added\_at")]

pub added\_at: i64,

#[serde(rename = "tensor\_shape")]

#[serde(default)]

pub tensor\_shape: Vec<i64>,

}

#[derive(Default, Debug, Clone, PartialEq, Serialize, Deserialize)]

#[serde(rename\_all = "camelCase")]

pub struct InputFeatureMapRecord {

#[serde(rename = "feature\_id")]

pub feature\_id: i64,

#[serde(rename = "full\_feature\_name")]

pub full\_feature\_name: String,

#[serde(rename = "feature\_type")]

pub feature\_type: i64,

pub index: i64,

#[serde(rename = "maybe\_exclude")]

pub maybe\_exclude: bool,

pub tag: String,

#[serde(rename = "added\_at")]

pub added\_at: i64,

}