Contents

1 Label Mapping Fix for SkolClassifierV2

1.1 Issues Resolved

1.1.1 1. `combined_idf does not exist` Error

Problem: When `use_suffixes=False`, the SkolModel was hardcoded to use the `combined_idf` features column, which only exists when `use_suffixes=True`. This caused a runtime error.

Solution: - Modified SkolClassifierV2 to dynamically determine the correct features column name using `FeatureExtractor.get_features_col()` - Pass the correct `features_col` to the SkolModel constructor - The model now uses `word_idf` when suffixes are disabled and `combined_idf` when enabled

1.1.2 2. `'list' object has no attribute 'items'` Error

Problem: The code tried to call `.items()` on the result of `get_label_mapping()`, which returns a list, not a dictionary.

Solution: - Fixed the label mapping creation to properly convert the labels list into the required dictionaries - `_label_mapping`: Dict[str, int] - maps label names to indices (e.g., `{'Label1': 0, 'Label2': 1}`) - `_reverse_label_mapping`: Dict[int, str] - maps indices to label names (e.g., `{0: 'Label1', 1: 'Label2'}`)

1.2 Files Modified

1.2.1 1. skol_classifier/model.py

Changes: - Updated `fit()` method to properly accept and store the `labels` parameter - Labels are now stored for later use in `predict_with_labels()`

Before:

```
def fit(self, train_data: DataFrame, labels: Optional[List[str]] = None) -> Pipe
    labels = labels or self.labels  # This didn't work correctly
    classifier = self.build_classifier()
    pipeline = Pipeline(stages=[classifier])
    self.classifier_model = pipeline.fit(train_data)
    return self.classifier_model
```

After:

```
def fit(self, train_data: DataFrame, labels: Optional[List[str]] = None) -> Pipe
    if labels is not None:
        self.labels = labels
    classifier = self.build_classifier()
    pipeline = Pipeline(stages=[classifier])
    self.classifier_model = pipeline.fit(train_data)
    return self.classifier_model
```

1.2.2 2. skol_classifier/classifier_v2.py

Changes: - Get correct features column name from FeatureExtractor - Pass `features_col` to SkolModel constructor - Pass labels from feature extractor to model's fit() method - Properly create label mappings from the labels list - Added train/test split and statistics calculation

Key additions:

```
# Get the features column name based on configuration
features_col = self._feature_extractor.get_features_col()

# Train model with correct features column
self._model = SkolModel(
    model_type=self.model_type,
    features_col=features_col,  # This is the fix!
    **self.model_params
)

# Get labels from feature extractor
labels = self._feature_extractor.get_labels()
```

```python
# Fit model and pass labels for later use
self._model.fit(featured_df, labels=labels)

# Store label mappings (labels is a list like ['Label1', 'Label2'])
labels_list = self._feature_extractor.get_label_mapping()
if labels_list is not None:
    # Create dict mapping from label to index
    self._label_mapping = {label: i for i, label in enumerate(labels_list)}
    # Create reverse mapping from index to label
    self._reverse_label_mapping = {i: label for i, label in enumerate(labels_lis
```

1.3 Testing

All fixes have been verified:

✅ **Feature Column Selection**: Tested that the model correctly uses: - `word_idf` when `use_suffixes=False` - `combined_idf` when `use_suffixes=True`

✅ **Label Mapping**: Verified that: - Labels list is properly converted to dictionaries - `_label_mapping` has correct type `Dict[str, int]` - `_reverse_label_mapping` has correct type `Dict[int, str]`

✅ **Model Training**: Confirmed that: - SkolModel can train with both feature configurations - Labels are passed correctly to the model - Predictions work with label conversion

1.4 Example Usage

```python
from pyspark.sql import SparkSession
from skol_classifier.classifier_v2 import SkolClassifierV2

spark = SparkSession.builder.getOrCreate()

# Works with use_suffixes=False (word_idf)
classifier1 = SkolClassifierV2(
    spark=spark,
    input_source='files',
    file_paths=['data/*.ann'],
    use_suffixes=False,  # Uses word_idf
    model_type='logistic'
)
results1 = classifier1.fit()

# Works with use_suffixes=True (combined_idf)
classifier2 = SkolClassifierV2(
```

```
    spark=spark,
    input_source='files',
    file_paths=['data/*.ann'],
    use_suffixes=True,  # Uses combined_idf
    model_type='logistic'
)
results2 = classifier2.fit()
```

1.5 Impact

These fixes ensure that: 1. SkolClassifierV2 works correctly with both word-only and word+suffix features 2. Label mappings are properly maintained throughout the training and prediction pipeline 3. The V2 API is now fully functional and ready for use