Claims Note Visualization for By-Peril Model Evaluation

Colin Gibbons-Fly

Table of contents

1 Project Overview

This notebook supports the AI Model Validation capstone project at **CSAA Insurance Group**, where we evaluate the *By-Peril* classification model developed by the Labs team. The model aims to improve peril specificity for enhanced pricing segmentation.

The visualization tool built here leverages transformer-based sentence embeddings and UMAP projection to: - Analyze the semantic space of claim notes, - Evaluate agreement between human-assigned and model-predicted peril types, - Identify ambiguous cases for retraining or manual review.

2 Objectives

- Transform unstructured claim notes into vector embeddings using a small Hugging Face transformer.
- Project high-dimensional embeddings into 2D space for visualization.
- Highlight patterns in peril agreement/disagreement.
- Empower business stakeholders and model developers with interpretable insights.

3 Input Data

The tool expects a CSV with the following key fields:

Column	Description
`DESCRIPTION`	Brief summary of the claim
`NOTES_SUMMARY`	Adjuster or agent notes
`CURRENT_PERIL_TYPE`	Existing peril label
`MODEL_PERIL_TYPE`	Predicted label from the By Peril model
`JUDGE_LABEL`	(Optional) Verdict from LLM-as-a-Judge

4 Code Overview

4.1 1. Data Loader & Initializer

```
class PerilVisualizationAnalysis:
    def __init__(self, csv_path: str):
        self.df = pd.read_csv(csv_path)
        self.embeddings = None
        self.umap_coords = None
```

This class handles CSV loading and prepares state for embeddings and visualization.

4.2 2. Preprocessing

```python
def preprocess_data(self):
    self.df['model_agreement'] = (
        self.df['CURRENT_PERIL_TYPE'] == self.df['MODEL_PERIL_TYPE']
    )
    self.df['combined_text'] = (
        self.df['DESCRIPTION'].fillna('') + ' ' +
        self.df['NOTES_SUMMARY'].fillna('')
    )
```

- Computes a model agreement flag.
- Combines text fields for transformer input.

4.3 3. Embedding Generation

```python
sbert = SentenceTransformer("all-MiniLM-L6-v2")
self.embeddings = sbert.encode(self.df['combined_text'].tolist(),
show_progress_bar=True)
```

Uses a compact Sentence-BERT model to encode each claim note into a 384-dimensional vector.

4.4 4. Dimensionality Reduction

```python
from umap import UMAP

umap = UMAP(n_components=2, metric='cosine', random_state=42)
self.umap_coords = umap.fit_transform(self.embeddings)
```

UMAP compresses embeddings to 2D while preserving semantic structure.

4.5 5. Visualization

```python
def plot_by_peril_type(...):
    ...
```

- Produces an interactive scatter plot.

- Colors and filters data by peril type or agreement flag.
- Enables:
  - Peril clustering analysis,
  - Mislabel detection,
  - Embedding-based comparison across model versions.

5 Insights Delivered

```
This tool creates an interpretable visual map of semantic space,
helping stakeholders detect:
- Confusion zones between similar perils (e.g., wind vs hail),
- Outlier or borderline claims,
- Systematic bias in either the model or reference labels.
```

6 Use Cases by Stakeholder

Role	Insight Gained
Pricing Actuaries	Confirm peril accuracy for better segmentation
Data Scientists	Identify patterns in misclassification, flag retraining zones
Business Owners	Visualize model behavior and impact