MLOps Group No: 92

Group Members Names:

1. PEYALA SAMARASIMHA REDDY - 2023AA05072
2. PEGALLAPATI SAI MAHARSHI - 2023AA05924
3. ANIRUDDHA DILIP MADURWAR - 2023AA05982
4. K VAMSIKRISHNA - 2023AA05209

MLOps Assignment-2

M2: Feature Engineering & Explainability - 10M

Justification of selected features based on explainability results

We have used the Fashion MNIST dataset, as it is a complex image data, it will not give the direct explainable things in auto eda, LIME or SHAP reports or visualizations because it's not the tabular data or structure data. Fashion MNIST data is converted into flatten dimensional shape and give for the model building or visualization, so here we visualised the 28x28 images into 784 pixel feature space, so these are visualised in the plots here, so based only on these we analysed.

1. t-SNE (Dimensionality Reduction and Visualization)

Insights:

- The **t-SNE (t-distributed Stochastic Neighbor Embedding)** visualization reduces the **784-pixel feature space** into 2D space while preserving local similarity.
- It shows how well the dataset clusters into distinct classes.
- The quality of clustering determines whether the raw feature space captures meaningful information.
- **Good Separation:** If classes are well-separated, raw pixel values contain meaningful information, and deep feature extraction may not be necessary.
- **Overlapping Clusters:** If significant overlap is observed, feature extraction techniques such as PCA or autoencoders may be required to improve class separation.
- **Dimensionality Reduction Potential:** Instead of using all 784 pixels, we can use lower-dimensional representations, keeping only the most informative features.

Feature Engineering Strategy:

- Reduce feature space using **Principal Component Analysis (PCA)** or **Autoencoders** before applying classifiers.
- Focus on regions of the image where class separation is strongest, avoiding redundant pixels.
- Use **edge detection filters** (like Sobel or Canny) to enhance feature extraction.

2. InterpretML (Feature Importance via Explainable Boosting Machines - EBM)

Insights:

- **EBM (Explainable Boosting Machine)** identifies the most influential features (pixels) in classification.
- The bar graph highlights the **top-ranked pixels** based on their contribution to the model's decision-making.
- **Highly Ranked Pixels:** Some pixels have a significantly higher contribution than others, meaning not all 784 pixels are equally important.
- **Spatial Importance:** The most important pixels tend to be concentrated in **edges, contours, and key structural areas** of the image (e.g., collars, sleeves, shoe edges).
  Avoiding Redundant Pixels: Many pixels in the background or non-informative areas contribute little and can be safely removed.

Feature Engineering Strategy:

- Select only **highly ranked pixels** (top 100-200) instead of all 784 pixels to reduce computational cost.
- Use **image processing techniques (like thresholding and edge detection)** to emphasize the most informative regions.
- Consider **supervised feature selection** where the least important pixels are pruned iteratively.

3. SHAP (Shapley Additive Explanations - Feature Contribution Per Sample)

Insights:

- SHAP values indicate **how much each pixel contributes** to a particular prediction.
- **Red pixels** increase confidence in a particular class, while **blue pixels** reduce confidence.
- **SHAP explains individual predictions**, unlike EBM, which gives an overall feature importance ranking.
- **Localized Influence:** Some pixels are highly relevant for certain samples but not globally important.
- **Context-Specific Importance:** The most critical pixels depend on the class—edges may be more important for shoes, while texture pixels matter for shirts.

Feature Engineering Strategy:

- Use **class-specific feature selection** instead of a universal approach.

- Implement **adaptive feature pruning**—keep relevant pixels based on SHAP values rather than applying a fixed mask.
- Consider **region-based analysis**, extracting features from **local patches** instead of using all pixels.

4. LIME (Local Interpretability with Perturbation)

Insights:

- LIME (Local Interpretable Model-Agnostic Explanations) generates **perturbations** and measures how predictions change. It highlights regions that strongly influence classification.
- **Region-Based Importance:** Unlike pixel-wise SHAP, LIME shows which **larger regions** matter most.
- **Human-Readable Interpretability:** LIME reveals the **part of the image that contributes most** to the model's decision, making it useful for debugging.
- **Robustness Check:** If key regions **vary significantly across perturbations**, the model may be relying on spurious correlations.

Feature Engineering Strategy:

- Instead of **pixel-level selection**, consider **region-based selection**—dividing images into **3x3 or 4x4 patches** and using only the most relevant ones.
- Validate feature importance across different **perturbation-based techniques** to ensure robustness.Improve interpretability by ensuring the model is **not relying on misleading patterns** (e.g., background artifacts).

Final Feature Selection Strategy and Justification (used it for refining) :

Based on the above analysis, the optimal feature selection approach involves:

1. **Dimensionality Reduction:**
    - Use **PCA** to reduce from **784 to ~50-100 principal components** while preserving variance.
    - Train models on lower-dimensional features instead of raw pixels.

2. **Feature Pruning via EBM & SHAP:**
    - Keep the **top 100-200 most important pixels** identified by EBM.
    - Remove pixels with **near-zero importance** across multiple explainability techniques.

3. **Region-Based Feature Extraction (LIME-Inspired):**
    - Divide the image into **small patches (e.g., 3x3 or 4x4 regions)** and select only the most informative ones.
    - Consider **convolutional features** instead of raw pixel values.

4. **Adaptive Feature Selection (Class-Specific):**
    - Use **different feature selection masks for different classes** based on SHAP values. Implement **context-aware pruning** (e.g., edges for footwear, texture for fabrics).