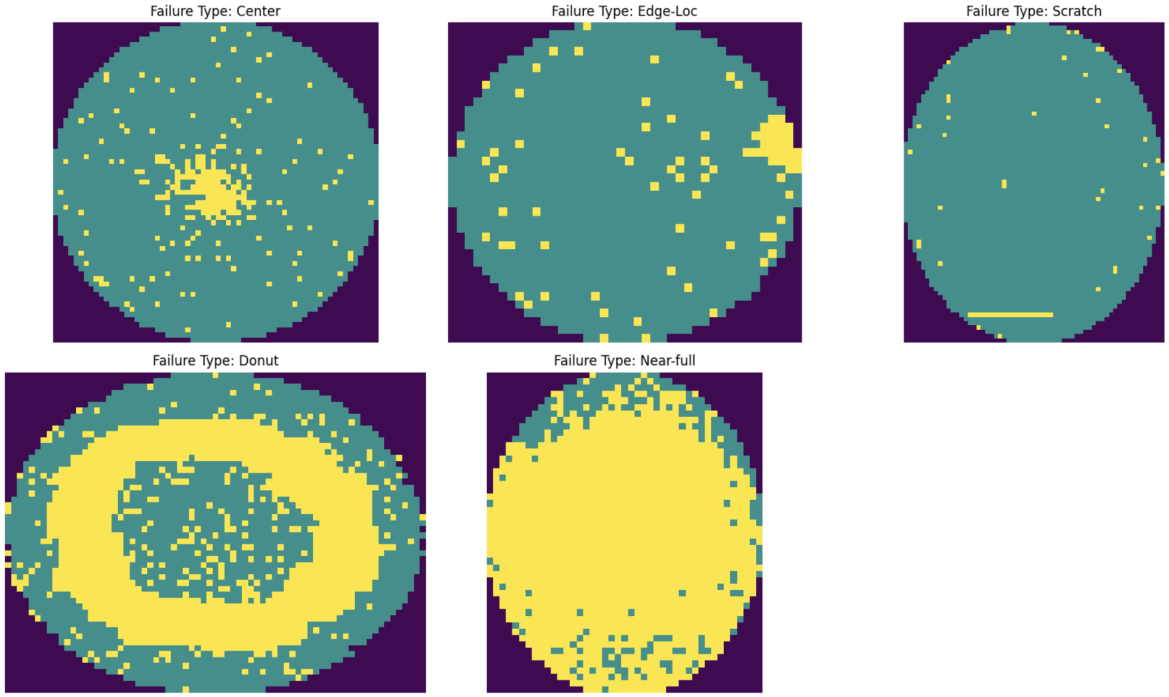
ECE 157A
Lab Report 2

The primary objective of this project is to harness the power of machine learning to predict potential failures in semiconductor manufacturing processes. By analyzing wafer maps and other associated data, the goal is to identify and categorize distinct types of manufacturing defects, ultimately enhancing quality control and reducing production costs. My approach started with a rigorous preprocessing of the wafer maps, standardizing them to a uniform size and converting categorical failure types into numerical labels to make the analysis easier. A baseline model using a Decision Tree classifier was initially used to get an overview of its prediction capabilities. Recognizing the need for better accuracy and precision, we transitioned to using a Support Vector Classifier (SVC). After training, the model's predictions were reverted from numerical labels back to their original string representations, providing a clear, interpretable output. The results were then stored into 2 CSV files.

Dataset Inspection:

Example wafers:

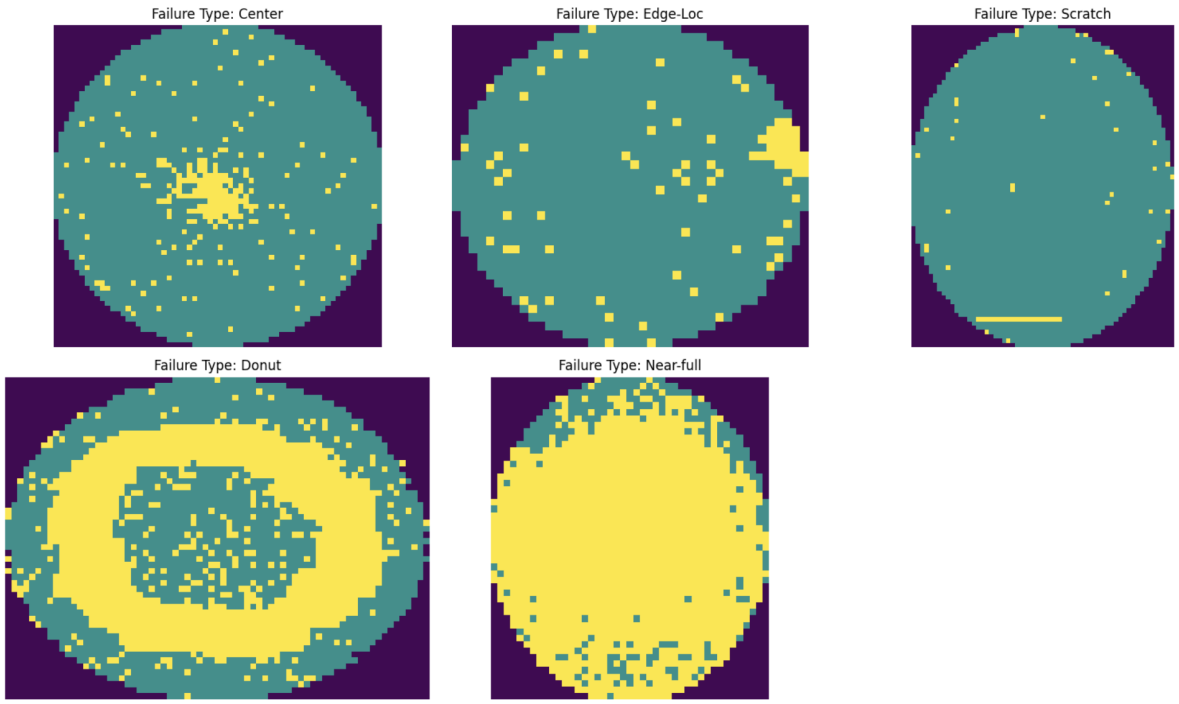


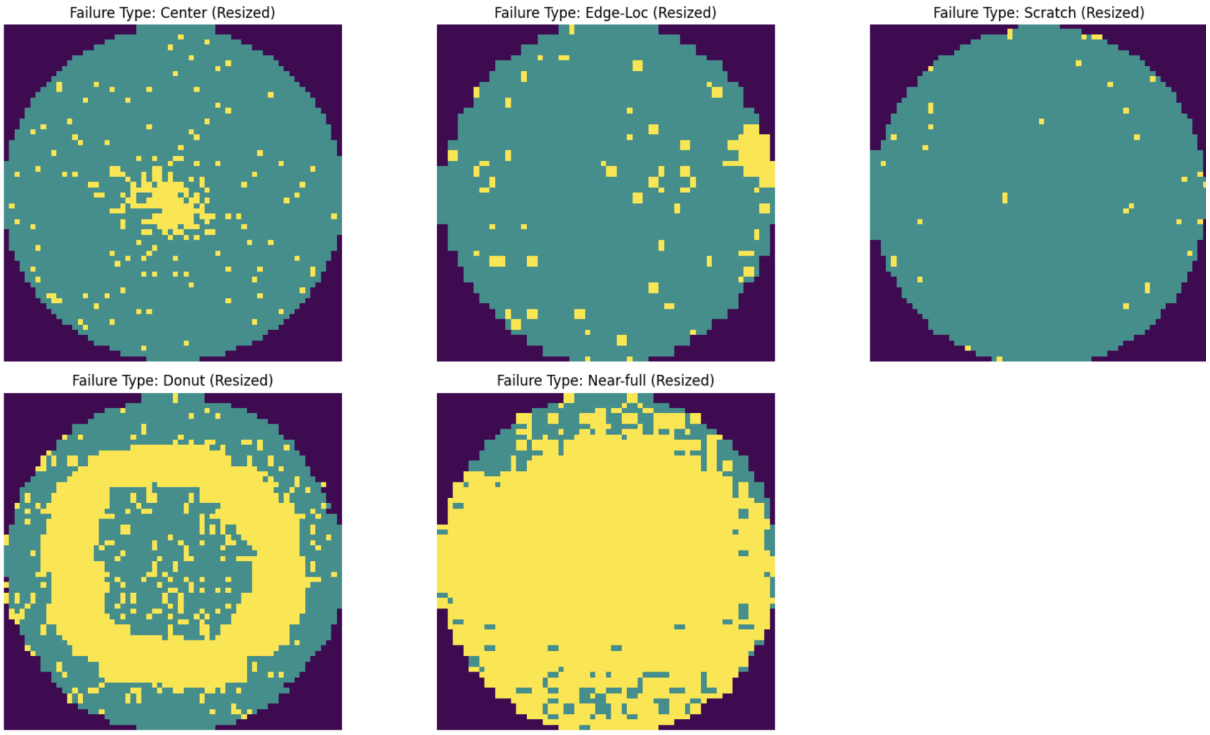
After inspecting the wafer maps for each class, we can make the following observations about the quality of the labels:

Center: This wafer shows a concentrated area of defects right at the center. This aligns well with its label, suggesting a clear defect localized in the middle of the wafer.
Edge-Loc: The defects on this wafer are dispersed near the periphery, aligning with its label. It suggests issues that might be occurring during the edge processing of the wafer.
Scratch: The wafer shows a distinct straight-line defect, resembling a scratch. The label seems appropriate as it captures the linear nature of the defect.
Donut: This wafer displays a unique pattern where there's an outer defect circle and another inner region with scattered defects, giving it a donut-like appearance. The labeling is apt given the circular pattern of defects.
Near-full: This wafer is almost entirely covered with defects, aligning with its "Near-full" label. It indicates a pervasive issue during the manufacturing process that affected almost the entire wafer.

Overall, the labels for the wafers seem to be of high quality, capturing the primary defect patterns in each instance. However, one potential point of contention could be the differentiation between widespread defects like "Near-full" and more concentrated but significant defects. We would have to ensure that the machine learning model can pick up on these nuances and not confuse a densely packed area of defects and a wafer that's nearly fully defective.

Dataset Preparation(Part 1):

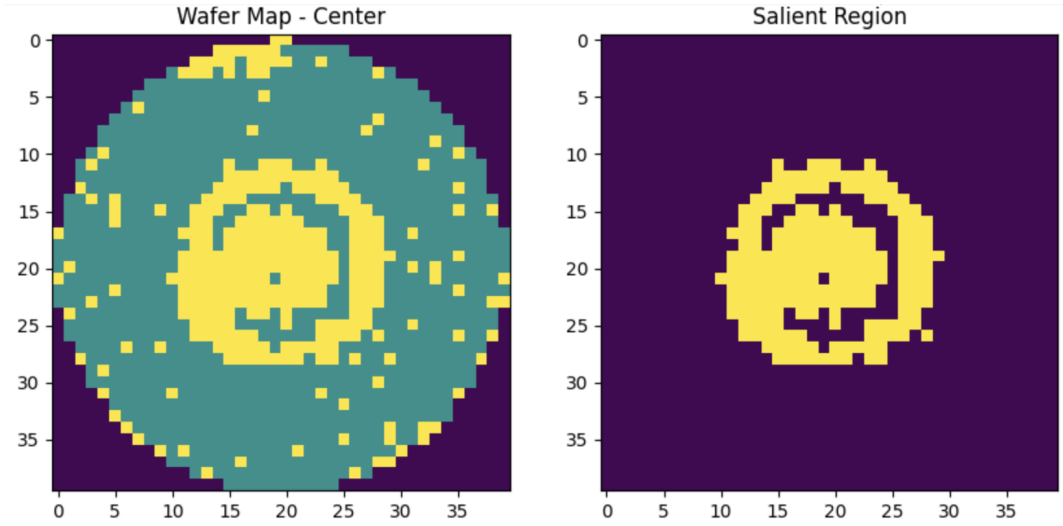




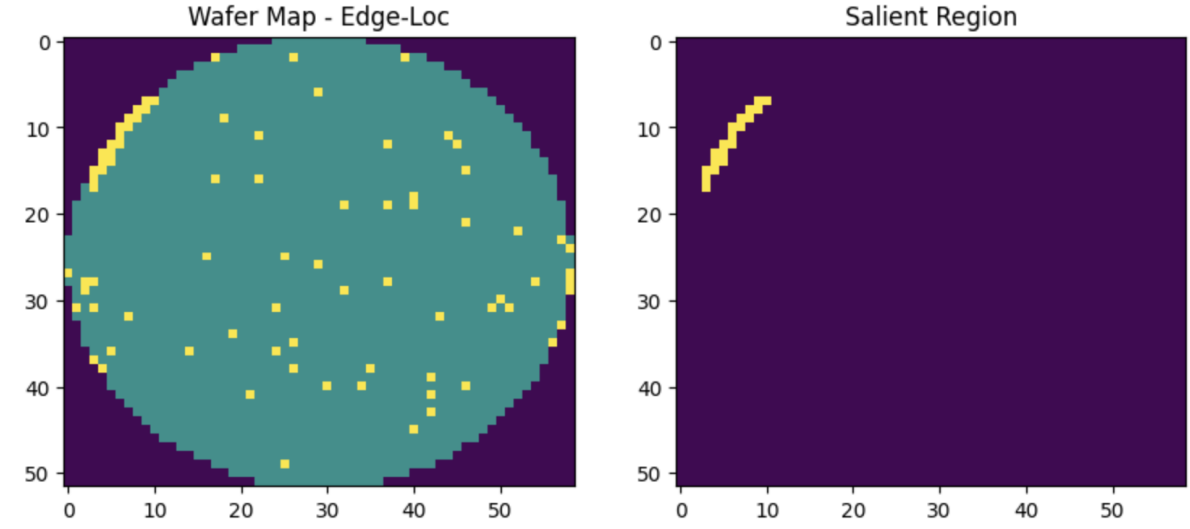

```
string2int = {
'Center': 1,
'Edge-Loc': 2,
'Scratch': 3,
'Donut': 4,
'Near-full': 5
}
```

Feature Engineering

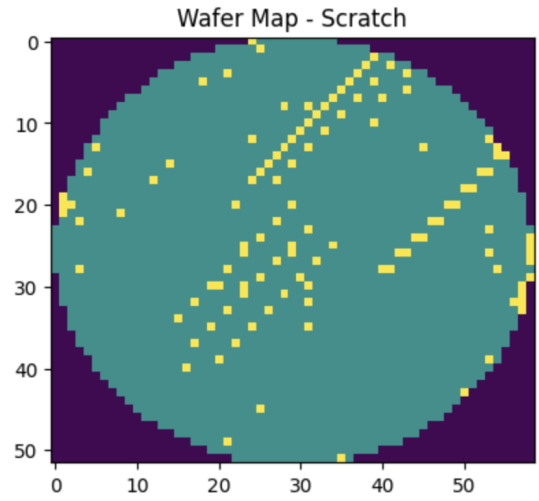
```
Failure Class: Center
Area Ratio: 0.28
Perimeter Ratio: 4.43
Max Distance from Center: 10.30
Min Distance from Center: 0.00
Major Axis Ratio: 0.99
Minor Axis Ratio: 0.91
Solidity: 0.74
Eccentricity: 0.39
Yield Loss: 0.19
Edge Yield Loss: 0.62
```

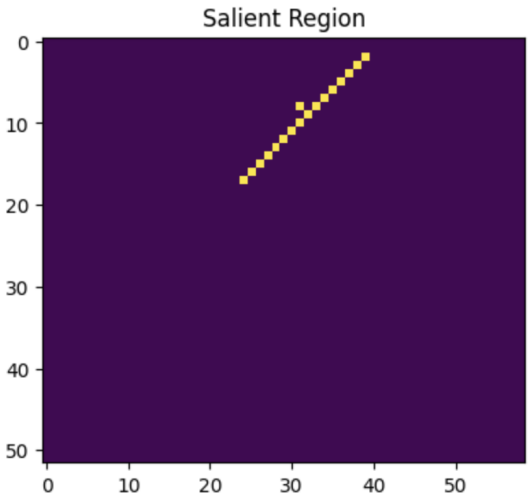


Failure Class: Edge-Loc
Area Ratio: 0.02
Perimeter Ratio: 0.46
Max Distance from Center: 28.69
Min Distance from Center: 27.23
Major Axis Ratio: 0.56
Minor Axis Ratio: 0.07
Solidity: 0.70
Eccentricity: 0.99
Yield Loss: 0.03
Edge Yield Loss: 0.18

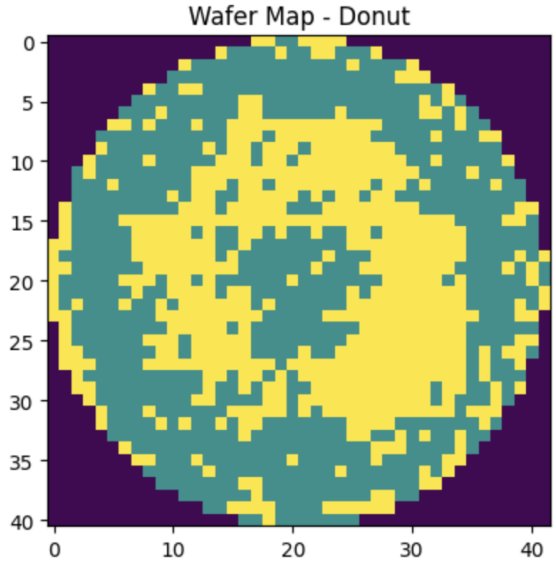


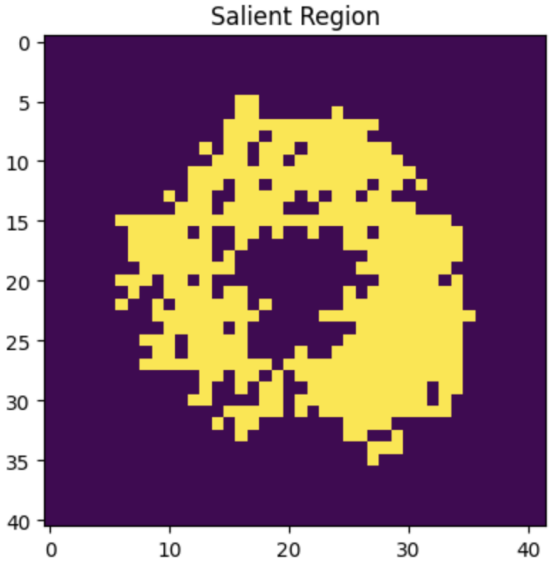
Failure Class: Scratch
Area Ratio: 0.01
Perimeter Ratio: 0.47
Max Distance from Center: 25.81
Min Distance from Center: 10.55
Major Axis Ratio: 0.97
Minor Axis Ratio: 0.05
Solidity: 0.52
Eccentricity: 1.00
Yield Loss: 0.04
Edge Yield Loss: 0.29





Failure Class: Donut
Area Ratio: 0.47
Perimeter Ratio: 7.48
Max Distance from Center: 16.29
Min Distance from Center: 3.20
Major Axis Ratio: 1.58
Minor Axis Ratio: 1.36
Solidity: 0.64
Eccentricity: 0.52
Yield Loss: 0.34
Edge Yield Loss: 0.79

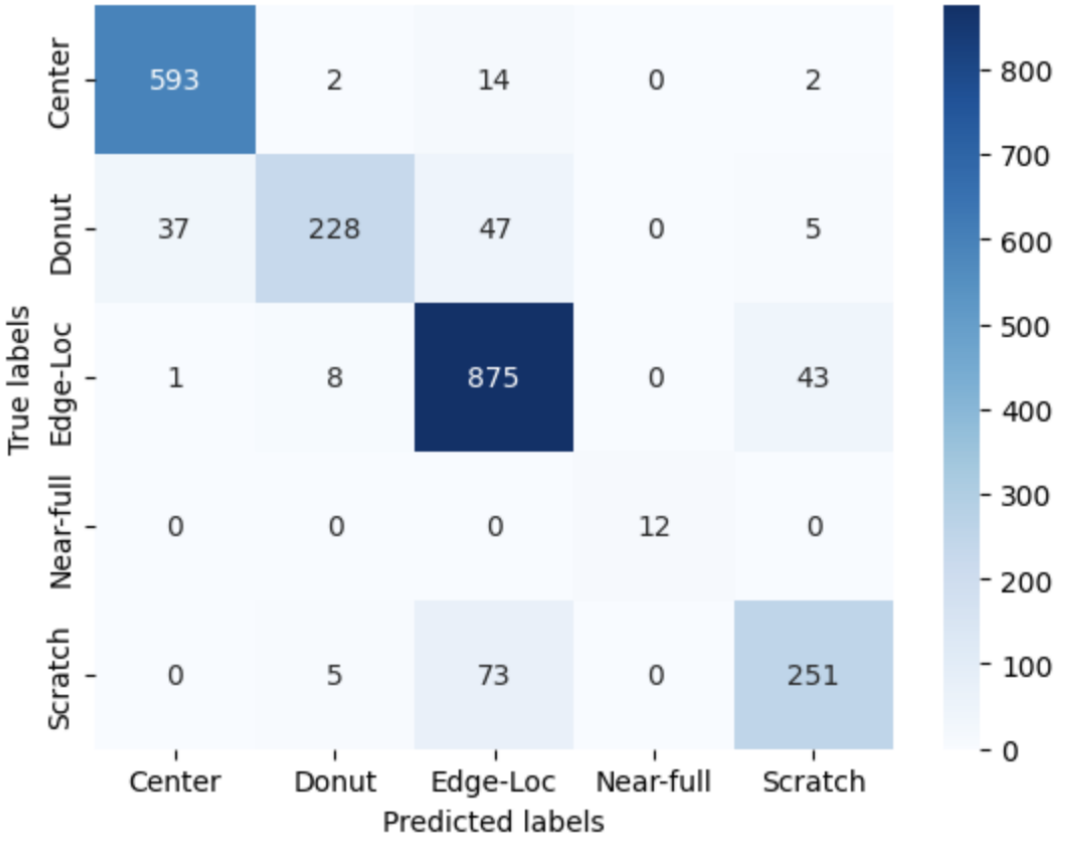




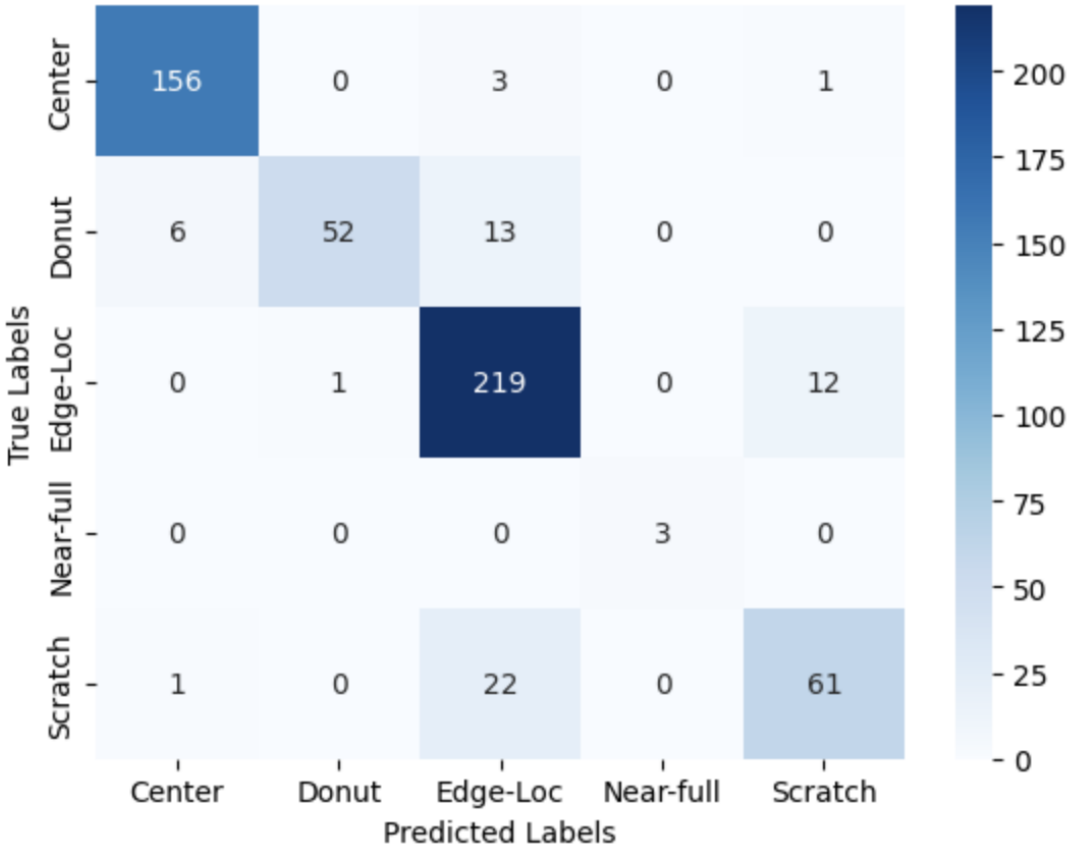
Failure Class: Near-full
Area Ratio: 0.99
Perimeter Ratio: 8.07
Max Distance from Center: 22.07
Min Distance from Center: 0.50
Major Axis Ratio: 2.01
Minor Axis Ratio: 1.93
Solidity: 0.96
Eccentricity: 0.28
Yield Loss: 0.77
Edge Yield Loss: 1.00

Model Training and Validation (Decision Tree)

```
Model accuracy on training data set is 89.21%
Model accuracy on training data set for each class is {'Center':
0.9705400981996727, 'Donut': 0.7192429022082019, 'Edge-Loc':
0.9439050701186623, 'Near-full': 1.0, 'Scratch': 0.7629179331306991}%
```




```
Model accuracy on validation data set is 89.27%
Model accuracy on validation data set for each class is {'Center': 0.975,
'Donut': 0.7323943661971831, 'Edge-Loc': 0.9439655172413793, 'Near-full':
1.0, 'Scratch': 0.7261904761904762}%
```



- **Center:**
  Rule: Starting at the root, if maxDistFromCenter is less than or equal to 32.078, and eccentricity is less than or equal to 0.936, then the majority class is "Center" with 59 samples.
  Intuitive Description: The rule essentially filters out defects based on their distance from the center and their shape (eccentricity). If a defect is close to the center and has a certain shape profile (defined by the eccentricity), it's labeled as a "Center" defect. This is intuitive as we'd expect center defects to be close to the center of the object.
- **Edge-Loc:**
  Rule: Starting at the root, if minDistFromCenter is greater than 12.107, and areaRatio is less than or equal to 0.822, then the majority class is "Edge-Loc" with 78 samples.
  Intuitive Description: Edge-Loc defects appear to be primarily determined by their distance from the center (being farther away) and their relative size (area ratio). This makes sense intuitively, as defects located on the edge would be farther from the center, and their size might differentiate them from other edge defects.

- **Donut**:
  Rule: If minDistFromCenter is greater than 12.107, and areaRatio is greater than 0.822 but less than or equal to 0.054, then the majority class is "Donut" with 36 samples.
  Intuitive Description: Donut defects seem to be characterized by being farther from the center and having a specific size profile. The rule suggests that these defects have a distinctive size (area ratio) which might resemble a donut. This is intuitive if we think of donuts as circular defects that might not be directly at the center or the edge, but somewhere in between.
- **Scratch:**
  Rule: If maxDistFromCenter is less than or equal to 32.078, eccentricity is less than or equal to 0.936, and gin is greater than 0.165, then the majority class is "Scratch" with 11 samples.
  Intuitive Description: Scratches seem to be identified based on their proximity to the center, their shape (eccentricity), and a specific "gin" value. Intuitively, scratches might be close to the center, but they may have a distinct shape or pattern that sets them apart from other defects. The "gin" criterion might capture this distinct pattern.
- **Near-full:**
  Rule: If minDistFromCenter is greater than 12.107, areaRatio is less than or equal to 0.822, and gin is equal to 0.0, then the class is "Near-full" with 12 samples.
  Intuitive Description: Near-full defects are farther from the center, have a specific size profile (defined by the area ratio), and have a gin value of 0. The gin value might suggest a particular uniformity or pattern. Intuitively, a near-full defect might be a defect that covers almost the full object, hence the specific size and pattern criteria.

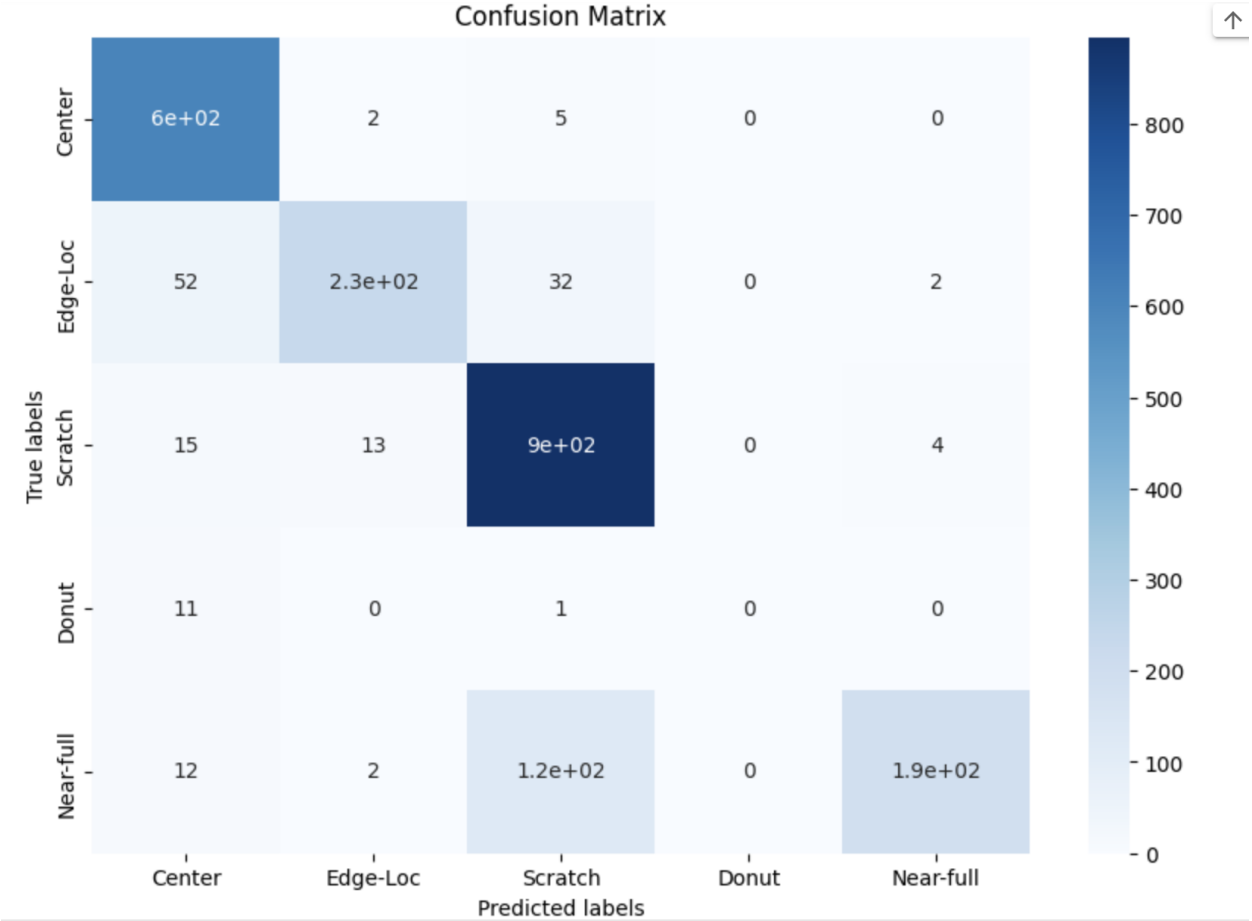
In summary, the rules derived from the decision tree generally align with the intuitive descriptions of the failure types. Each rule captures unique characteristics that differentiate one defect type from another.

Model Training and Validation (SVC)

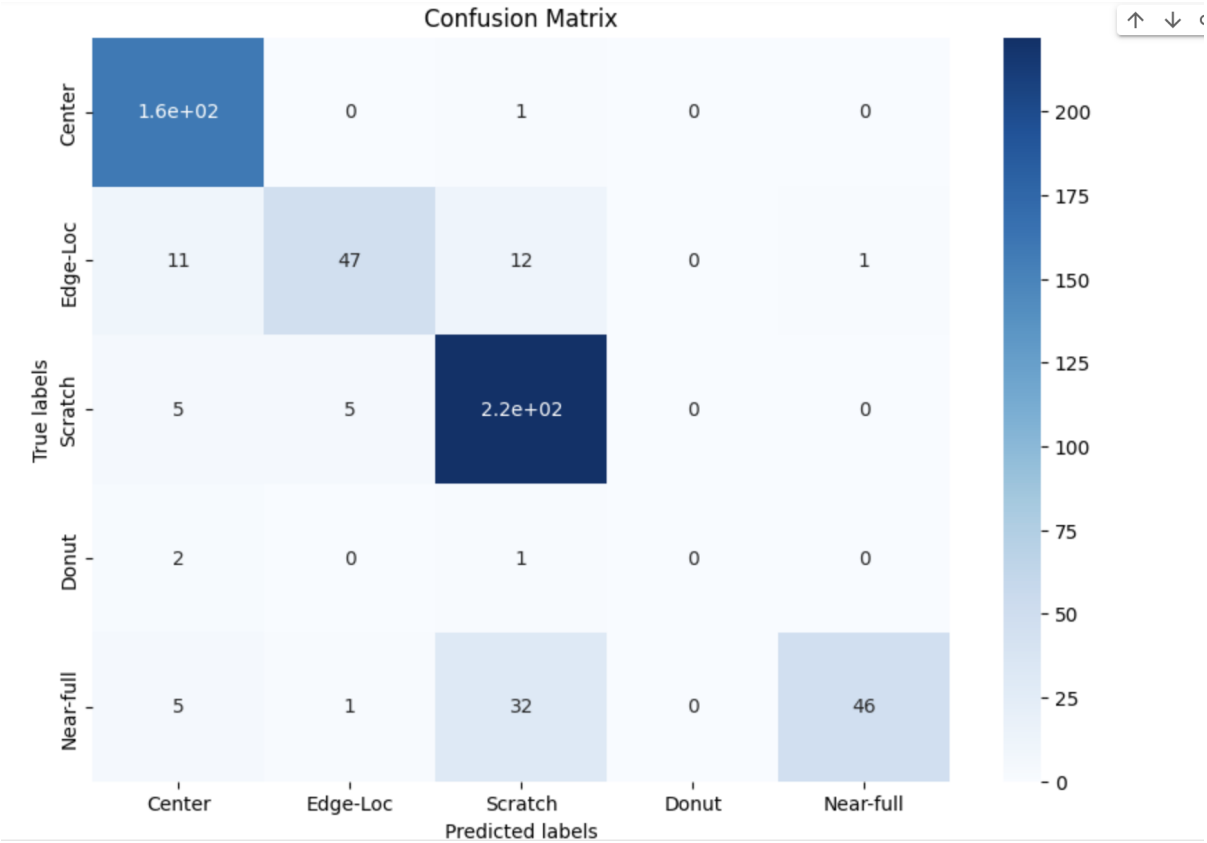
```
Per Failure Type Accuracy for Training Samples:
Accuracy for Center: 98.85%
Accuracy for Edge-Loc: 72.87%
Accuracy for Scratch: 96.55%
Accuracy for Donut: 0.00%
Accuracy for Near-full: 58.36%

Per Failure Type Accuracy for Validation Samples:
Accuracy for Center: 99.38%
Accuracy for Edge-Loc: 66.20%
Accuracy for Scratch: 95.69%
Accuracy for Donut: 0.00%
Accuracy for Near-full: 54.76%
```

Training Set:



Validation Set:



Questions (Decision Tree and SVC)

- Decision Trees offer a transparent way of understanding decisions made by the model. It's like following a flowchart to a conclusion, making it easier to explain and justify results. On the downside, they can be sensitive to data changes and might overfit. Support Vector Classifiers (SVC), on the other hand, are robust and can handle complex decision boundaries well. However, they are computationally intensive and are not as straightforward to explain as decision trees. In a semiconductor company setting, if interpretability is a priority—for instance, to give reasons for wafer rejections—a decision tree would be the choice. However, if prediction accuracy is more critical and there are enough computational resources, SVC might be preferred.
- Achieving 99% accuracy is really hard, especially if the features don't capture all underlying patterns like in this case. To approach this, one might consider gathering more data or different types of data related to wafer maps. Regular reviews of model performance on new data can also help, allowing for model updates as needed. In terms of feature engineering, domain-specific knowledge can be a goldmine. Insights from experts in semiconductor manufacturing might reveal additional features or lead to the transformation of existing features to better capture the nuances of wafer defects.