**1.Exploratory Data Analysis:**

* **Initial overview :** It has been found that there are no missing values in our dataset.
* **Target variable analysis:** It shows a very severe class imbalance through our observations and illustrations which means only a tiny fraction of our data represents the "epitope patterns" (Class 1). This is very important when handling with data preprocessing and modelling steps and also needs strategies like class rebalancing.
* **Feature analysis:** I have provided the descriptive statistics by transposing the output and focusing on specific insights rather than trying to view 1650 columns at once. Using filtering, there are some columns that don’t vary much can be removed later. Also, some features indicating potential outliers or features on a vastly different scale than others.
* **Group based analysis:** There are 94 different groups (batches) , each having its respective invader parts (rows). This is very important for splitting our data into training and testing sets and also helps prevent data leakage .
* **Visualization :** Principal Component Analysis (PCA), clearly shows 91% captured variance in dimensions but severe non-linearity and also sets expectations for the difficulty of the classification task and highlights the need for robust preprocessing and modelling strategies.

**2.Data Pre-processing:**

* **Sorting:** Separating the feature columns with only numerical variable from the mixed datatypes is needed for the next steps in the pipeline such as imputing, scaling and also keeping the info\_group column into a separate variable for group-based splitting for training and testing.
* **Imputation and Variable scaling:** The imputation step fills the missing values with the chosen median and since there are no missing values found already , keeping it makes our pipeline more robust if df\_holdout.csv introduce missing values. The scaling step transformed the features to have a mean of 0 and a std deviation of 1 with similar numerical ranges (-3 to 3) provides consistent scaling.
* **Feature selection:** Initially, tried PCA with n\_components=100 gives only 59.5% variance and biased classification. Then, tried PCA with 95% variance reduced 1635 to 1000 (features) but still misclassification. Finally, selectKbest (with mutual information) has been chosen which focuses not only variance but also relevance. However, some powerful and valid techniques like BORUTA or MRMR should be chosen but it has not , due to its higher computational cost and time.
* **Class rebalancing:** It clearly shows that SMOTE (Synthetic Minority Over-sampling Technique) directly addresses the severe class imbalance issue identified in your EDA. Also, I applied SMOTE here to the full data set only for demonstration purposes, instead it should be applied before splitting to prevent data leakage within cross-validation folds.

**3.Modelling:**

* **Pipeline creation:** Creating a pipeline for imputing (treating missing values), scaling (universal scale) and for feature selection using selectKbest with 200 features using mutual information.
* **Model selection:** Random Forest Classifier and XGBoost Classifier models because of its good performance on non-linear datasets, high dimensional datasets, its robustness and also good at handling datasets with severe class imbalance. This is only my prior belief but the actual performance metrics will be how well they *actually* predict the correct badge on unseen data. Also, XGBoost classifier has a scale weight that specially handles the minority classes.
* **Model training:** GroupKFold with 5 folds that ensures that all data points from a single Info\_group will always stay together, either entirely in the training set or entirely in the test set in order to prevent data leakage. The outermost and the inner loops has been executed carefully that the pipeline's steps *learn* their parameters and applying only in the training sets. It does *not* learn any new parameters from the test data. SMOTE is applied *only* to the processed training data. The chosen model is trained using the **fully processed and SMOTE-resampled training data** for the current fold. After training, the model makes predictions on the X\_test\_processed data. The test set learned *only from the training data* and was *not* resampled by SMOTE. This ensures a fair and realistic evaluation.

**4.Final Summary:**

* The biggest challenge here is predicting a rare (Class 1) from a massive dataset (45,000 samples, 1635 features) with severe class imbalance and preventing data leakage.
* To overcome this, SMOTE and some feature reduction techniques were used.
* The initial attempt of PCA with 100 components and aiming for PCA with 95% explained variance has been tried but failed to predict the classes.
* Therefore, I have chosen SelectKBest with mutual information directly selects features based on their **relevance to the target variable (including non-linear relationships)**, which is a more targeted approach for classification.
* Initially tried with 200 features achieving a **Balanced Accuracy of 0.70 and a Recall of 0.40** for both models (XGBoost slightly better). This showed a significant improvement, indicating that more relevant features has been identified.
* Increasing k from 200 to 500 likely introduced **more noise or irrelevant features** that diluted the signal, making it harder for the models to learn. (RandomForestClassifier went back to 0.00 recall, and XGBoost's Balanced Accuracy dropped to 0.52 and Recall to 0.05)
* RandomForestClassifierconsistently failed to predict Class 1 when using PCA or 500 features even with balanced class weight, it seems overly conservative. On the other hand, XGBoostClassifier performed slightly bettter, especially in terms of Recall, and was significantly faster. It showed its best performance with 200 features.
* Therefore, SelectKBest with 200 features was the most effective feature reduction strategy and XGBoost is the "better" model among all my trials leading to the best Balanced Accuracy and Recall for Class 1.

**5.Limitations:**

* I am very much limited by my personal computer's capacity , prevented comprehensive hyperparameter tuning using RandomizedSearchCV (which I attempted) , as models often perform much better with optimized parameters. This also prevented me trying more computationally intensive feature reduction methods like BORUTA or MRMR.
* The models are using a default 0.5 probability threshold and future work should involve finding an optimal threshold that balances Precision and Recall based on the specific costs of false positives vs. false negatives in this problem.