Feature Engineering Report

1. Feature Creation

The following new features were created during the feature engineering process:

- Total Units Enrolled: This feature is the sum of curricular units enrolled in the 1st and 2ndsemesters. It provides insights into the overall engagement of a student with their courses.
- Total Units Approved: This is the sum of curricular units approved in both semesters. It serves as an indicator of a student's academic success and progress.
- Approval Rate: This feature is calculated as the ratio of total units approved to total units enrolled. A higher approval rate can indicate better academic performance and understanding of the coursework.
- Improvement in Grades: This feature reflects the difference between the 2nd-semester and 1st-semester grades. It is useful for identifying students who show academic improvement or decline over time.
- Economic Hardship: Calculated as the sum of the unemployment rate and inflation rate, minus GDP. This feature provides a context for understanding the socio-economic environment affecting students.
- Total Units without Evaluations: This is the sum of curricular units without evaluations in both semesters. It can highlight potential gaps in assessment and learning opportunities.

2. Justification for Feature Transformations

Transformations were applied to various features to improve their distribution and relationship with the target variable:

Feature Engineering Report

- Log Transformation: Applied to skewed features to reduce skewness and stabilize variance. This transformation can enhance the performance of many machine learning algorithms that assume normally distributed data.
- Binning: Custom binning was performed on features such as 'Age at enrollment', 'Admission grade', and 'Previous qualification (grade)' to categorize continuous values into discrete bins. Which was later converted to a new feature. This can improve model interpretability and handle non-linear relationships. The respective bins for the feature binning was obtained using the fisher jenks algorithm. For the ‘Age at enrollment’, the intervals used were (17-19], (19- 22], (22-26], (26-30], (30-35], (35-41], (41-49], (49-70]. For the Admission grade, the intervals were (95-107.2], (107.2- 116.9], (116.9-125.2], (125.2-134.1], (134.1-144.6], (144.6-157.7], (157.7-190.0]. The previous qualification grade, the interval used were (95-112.0], (112.0- 122.0], (122.0-129.0], (129.0-136.0], (136.0-144.0], (144.0-154.0], (154.0-166.0], (166.0-190.0]. Three new features ‘age_bins’, ‘admission grade (bins) and ‘previous qualification grade (bins)’ was created using the binning method.

Fisher-Jenks Algorithm: The Jenks optimization method, also called the Jenks natural breaks classification method, is a data clustering method designed to determine the best arrangement of values into different classes. This is done by seeking to minimize each class's average deviation from the class mean, while maximizing each class's deviation from the means of the other classes. In other words, the method tries to reduce the within-class variance and maximize the between-class variance. The Fisher-Jenks optimization is a method used to determine optimal class intervals for continuous data. It is particularly relevant in because it helps in

- Minimize within-class variance: Fisher-Jenks optimization seeks to minimize the variance within each class while maximizing the variance between classes. This ensures

that data points within the same class are as similar as possible, while different classes are significantly different.

- Data-Based Classification: The Fisher Jenks algorithm provides a systematic way of classifying continuous data into meaningful categories, rather than relying on arbitrary grouping methods. You can create adaptive class definitions based on data distribution.

- Resistant to Outliers: The algorithm is robust to outliers and can provide stable class definitions, which is important in many real-world applications where data distributions may be skewed or noisy.

For both the feature engineering and feature transformation, three custom scikit learn transformers will be created to help improve code reusability.

- Feature Binner: The Feature Binner class is a custom scikit learn transformer designed for preprocessing features in a machine learning pipeline. Specifically, it focuses on binning numerical features, which is a common technique used to convert continuous variables into categorical ones. This transformation can enhance the model's performance by capturing non-linear relationships and simplifying the data. The Fisher's Jenks algorithm was used in determining the optimum range of values in each bin with the aid of the Jenkspy library.

```python
class FeatureBinner(BaseEstimator, TransformerMixin):
    def __init__(self):
        # Initialize the bins and labels
        self.age_bins = [16, 19, 22, 26, 30, 35, 41, 49, 71]
        self.ad_grade_bins = [94.0, 107.2, 116.9, 125.2, 134.1, 144.6, 157.7, 191.0]
        self.prev_grade_bins = [94.0, 112.0, 122.0, 129.0, 136.0, 144.0, 154.0, 166.0, 191.0]
        self.labels_ad = [0, 1, 2, 3, 4, 5, 6]
        self.labels_prev = [0, 1, 2, 3, 4, 5, 6, 7]

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        X = X.copy()

        # Apply the binning using pd.cut and label encoding them manually
        X['age_bins'] = pd.cut(X['Age at enrollment'], bins=self.age_bins, labels=self.labels_prev).astype('int')

        X['Admission grade (bins)'] = pd.cut(X['Admission grade'], bins=self.ad_grade_bins, labels=self.labels_ad).astype('int'

        X['Previous grade (bins)'] = pd.cut(X['Previous qualification (grade)'], bins=self.prev_grade_bins, labels=self.labels_
        return X
```

Fig 1: Code snapshot of the custom feature binner transform

- FeatureTransformer: The FeatureTransformer class is a custom transformer designed to enhance a dataset by creating new features derived from existing ones. It is built to integrate seamlessly with the Scikit-learn's pipeline.

```python
class FeatureTransformer(BaseEstimator, TransformerMixin):
    def __init__(self):
        pass

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        X = X.copy()

        # Total Units Enrolled
        X['Total Units Enrolled'] = X['Curricular units 1st sem (enrolled)'] + X['Curricular units 2nd sem (enrolled)']

        # Total Units Approved
        X['Total Units Approved'] = X['Curricular units 1st sem (approved)'] + X['Curricular units 2nd sem (approved)']

        # Average curricular units
        X['Average curricular units'] = (X['Curricular units 1st sem (grade)'] + X['Curricular units 2nd sem (grade)']) / 2

        # Approval Rate (Handle division by zero to remove NaN errors)
        X['Approval Rate'] = np.where(X['Total Units Enrolled'] != 0,
                                      X['Total Units Approved'] / X['Total Units Enrolled'],
                                      0)

        # Improvement in Grades
        X['Improvement in Grades'] = X['Curricular units 2nd sem (grade)'] - X['Curricular units 1st sem (grade)']

        # Economic Hardship
        X['Economic Hardship'] = X['Unemployment rate'] + X['Inflation rate'] - X['GDP']

        # Total Units without Evaluations
        X['Total Units without Evaluations'] = X['Curricular units 1st sem (without evaluations)'] + X['Curricular units 2nd se

        return X
```

Fig 2: Code snapshot of the Feature Transformer transformer

- Polynomial-Interaction-Features: The PolynomialFeaturesInteraction class is a custom transformer designed to enhance a dataset by generating polynomial and interaction features. By applying polynomial transformations specifically to a selected set of numerical features, this transformer allows machine learning models to capture more complex relationships within the data, potentially improving their predictive performance. The transformer is built to integrate seamlessly with Scikit-learn's pipeline.

```
class PolynomialFeaturesInteraction(BaseEstimator, TransformerMixin):
    def __init__(self):
        # Initialize the PolynomialFeatures transformer from sklearn
        self.poly_transformer = PolynomialFeatures(degree=2, interaction_only=True, include_bias=False)

    def fit(self, X, y=None):
        # Select numerical features to apply polynomial transformation
        self.numerical_features = ['Admission grade', 'Age at enrollment', 'Curricular units 1st sem (credited)',
                                   'Curricular units 1st sem (enrolled)',
                                   'Curricular units 1st sem (evaluations)',
                                   'Curricular units 1st sem (approved)',
                                   'Curricular units 1st sem (without evaluations)',
                                   'Curricular units 2nd sem (credited)',
                                   'Curricular units 2nd sem (enrolled)',
                                   'Curricular units 2nd sem (evaluations)',
                                   'Curricular units 2nd sem (approved)',
                                   'Curricular units 2nd sem (without evaluations)', 'Total Units Enrolled',
                                   'Total Units Approved', 'Average curricular units', 'Approval Rate', 'Improvement in Grades',
                                   'Economic Hardship', 'Total Units without Evaluations', 'Unemployment rate', 'Inflation rate',
                                   'GDP']
        # Fit the polynomial transformer on the numerical features
        self.poly_transformer.fit(X[self.numerical_features])
        return self

    def transform(self, X):
        # Transform the numerical features
        numerical_df = X[self.numerical_features]
        poly_features = self.poly_transformer.transform(numerical_df)

        # Create a DataFrame for the polynomial features
        poly_df = pd.DataFrame(poly_features, columns=self.poly_transformer.get_feature_names_out(self.numerical_features))

        # Drop the numerical features from the original DataFrame
        cat_df = X.drop(columns=self.numerical_features)
```

Fig 3: Code snapshot of the Polynomial Feature Interaction transformer

- LogTransformer: The LogTransformer class is a custom transformer designed to enhance a dataset by applying logarithmic transformation to certain skewed columns in the dataset.

```
class LogTransformer(BaseEstimator, TransformerMixin):
    def __init__(self, columns=None):
        # Initialize instance variables
        self.columns = None  # Will store the names of columns to transform
        self.new_cols = []  # List to keep track of columns selected for log transformation
        pass

    def fit(self, X, y=None):
        # Make a copy of the input DataFrame to avoid modifying the original data
        X = X.copy()

        # Identify columns to consider for log transformation by excluding specified categorical columns
        self.columns = X.drop(['Marital status', 'Daytime/evening attendance', 'Application order', 'Nationality', 'Course'
                               'Application mode', 'Previous qualification', "Mother's qualification",
                               "Father's qualification", "Mother's occupation", "Father's occupation",
                               'Displaced', 'Educational special needs', 'Debtor', 'Tuition fees up to date',
                               'Gender', 'Scholarship holder', 'International', 'age_bins',
                               'Admission grade (bins)', 'Previous grade (bins)', 'Target'], axis=1).columns

        # Loop through specified columns to assess skewness
        for col in self.columns:
            # Calculate skewness, ignoring NaN values
            col_skewness = stats.skew(X[col].dropna())

            # Check if skewness is above the threshold and ensure values are non-negative
            if (col_skewness > 1 or col_skewness < -1) and (X[col] >= 0).all():
                self.new_cols.append(col)  # Store column names for transformation
        return self

    def transform(self, X):
        # Create a copy of the input DataFrame
        X = X.copy()

        # Loop through the columns identified for log transformation
```

Fig 4: Code snapshot of the Log transformer

After this the three transformers will be combined in a scikit learn pipeline to transform and create new features in the dataset. The transformers was created to improve code reusability, modularity, scalability and to prevent data leakage into a new dataset. The pipeline was later saved using the pickle library as ‘feature_engineering.pkl’ for future use.

```
data_transformed
```

	Admission grade	Age at enrollment	Curricular units 1st sem (credited)	Curricular units 1st sem (enrolled)	Curricular units 1st sem (evaluations)	Curricular units 1st sem (approved)	Curricular units 1st sem (without evaluations)	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	...	Tuitio fee up t dat
0	127.3	3.044522	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	
1	142.5	2.995732	0.0	1.945910	6.0	6.0	0.0	0.0	6.0	6.0	...	
2	124.8	2.995732	0.0	1.945910	0.0	0.0	0.0	0.0	6.0	0.0	...	
3	119.6	3.044522	0.0	1.945910	8.0	6.0	0.0	0.0	6.0	10.0	...	
4	141.5	3.828641	0.0	1.945910	9.0	5.0	0.0	0.0	6.0	6.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4419	122.2	2.995732	0.0	1.945910	7.0	5.0	0.0	0.0	6.0	8.0	...	
4420	119.0	2.944439	0.0	1.945910	6.0	6.0	0.0	0.0	6.0	6.0	...	
4421	149.5	3.433987	0.0	2.079442	8.0	7.0	0.0	0.0	8.0	9.0	...	
4422	153.8	3.044522	0.0	1.791759	5.0	5.0	0.0	0.0	5.0	6.0	...	
4423	152.0	3.135494	0.0	1.945910	8.0	6.0	0.0	0.0	6.0	6.0	...	

4424 rows × 278 columns

Fig 5: The transformed dataset using the created feature engineering pipeline

3. Analysis of Feature Importance and Selection Results

Various methods were utilized to assess feature importance and select relevant features:

- Correlation Feature Selector: Correlation feature selection is a feature selection method used to identify and select features that are most relevant to the target variable while removing redundant or highly correlated features. The 50 selected features with a high correlation to the dataset were Curricular units 1st sem (approved), Average curricular units, Tuition fees up to date, Scholarship holder, Age at enrollment, Debtor, Gender, Application mode, Improvement in Grades, Admission grade Curricular units 1st sem (evaluations), Age at enrollment Economic Hardship, Improvement in Grades Inflation rate, Curricular units 1st sem (enrolled), Admission grade, Displaced, Previous qualification (grade), Curricular units 2nd sem (without evaluations), Curricular units 2nd sem (evaluations), Marital status, Application order, Curricular units 1st sem (without evaluations), Daytime/evening attendance, Curricular units 1st sem (without evaluations) Curricular units 2nd sem

(without evaluations), Previous qualification, Curricular units 2nd sem (without evaluations) Improvement in Grades, Curricular units 2nd sem (without evaluations) Inflation rate, Curricular units 1st sem (evaluations), GDP, Mother's qualification, Curricular units 2nd sem (without evaluations) GDP, Curricular units 1st sem (credited) Improvement in Grades, Curricular units 1st sem (without evaluations) Improvement in Grades, Curricular units 1st sem (credited), Curricular units 1st sem (credited) Inflation rate, Inflation rate, Economic Hardship, Curricular units 1st sem (without evaluations) GDP, Curricular units 2nd sem (approved) Improvement in Grades, Curricular units 1st sem (without evaluations) Inflation rate, Improvement in Grades GDP, Inflation rate GDP, Nationality, Curricular units 1st sem (credited) GDP, Unemployment rate, Curricular units 1st sem (credited) Curricular units 1st sem (without evaluations), Educational special needs, Mother's occupation, Curricular units 2nd sem (credited) Improvement in Grades, International, Curricular units 1st sem (credited) Curricular units 2nd sem (without evaluations).

ANOVA F-Statistic: ANOVA (Analysis of Variance) Feature Selection is a technique used to evaluate the statistical significance of features in relation to a target variable, particularly in classification tasks. It is often implemented using the ANOVA F-test to determine how much the different levels of a categorical target variable influence the variance in the numerical feature values. The SelectKBest method was employed using ANOVA F-statistic, selecting the top 50 features based on their statistical significance with respect to the target. They are

Curricular units 1st sem (approved), Curricular units 2nd sem (approved), Total Units Approved, Average curricular units, Approval Rate, Admission grade Curricular units 1st sem (approved), Admission grade Curricular units 2nd sem (approved), Admission grade Total Units Approved, Admission grade Approval Rate, Age at enrollment Curricular units 2nd sem (approved), Age at enrollment Total Units Approved, Age at enrollment Approval Rate, Curricular units 1st sem (enrolled) Curricular units 1st sem (approved), Curricular units 1st sem (enrolled) Curricular units 2nd sem

(approved), Curricular units 1st sem (enrolled) Total Units Approved, Curricular units 1st sem (enrolled) Approval Rate, Curricular units 1st sem (evaluations) Curricular units 2nd sem (approved), Curricular units 1st sem (evaluations) Total Units Approved, Curricular units 1st sem (evaluations) Approval Rate, Curricular units 1st sem (approved) Curricular units 2nd sem (enrolled), Curricular units 1st sem (approved) Curricular units 2nd sem (evaluations), Curricular units 1st sem (approved) Curricular units 2nd sem (approved), Curricular units 1st sem (approved) Total Units Enrolled, Curricular units 1st sem (approved) Total Units Approved, Curricular units 1st sem (approved) Average curricular units, Curricular units 1st sem (approved) Approval Rate, Curricular units 1st sem (approved) Economic Hardship, Curricular units 1st sem (approved) Unemployment rate, Curricular units 2nd sem (enrolled) Curricular units 2nd sem (approved), Curricular units 2nd sem (enrolled) Total Units Approved, Curricular units 2nd sem (enrolled) Approval Rate, Curricular units 2nd sem (evaluations) Curricular units 2nd sem (approved), Curricular units 2nd sem (evaluations) Total Units Approved, Curricular units 2nd sem (evaluations) Approval Rate, Curricular units 2nd sem (approved) Total Units Enrolled, Curricular units 2nd sem (approved) Total Units Approved, Curricular units 2nd sem (approved) Average curricular units, Curricular units 2nd sem (approved) Approval Rate, Curricular units 2nd sem (approved) Economic Hardship, Curricular units 2nd sem (approved) Unemployment rate, Total Units Enrolled Total Units Approved, Total Units Enrolled Approval Rate, Total Units Approved Average curricular units, Total Units Approved Approval Rate, Total Units Approved Economic Hardship, Total Units Approved Unemployment rate, Average curricular units Approval Rate, Approval Rate Economic Hardship, Approval Rate Unemployment rate, Curricular units 2nd sem (grade)

- Mutual Information: Mutual Information (MI) is a feature selection technique that measures the dependency between two variables, often used to select features based on how much information

they share with the target variable. It is particularly useful when detecting nonlinear relationships between features and the target. Mutual information was calculated to identify non-linear relationships,resulting in the selection of another set of 50 features. They are

'Curricular units 2nd sem (approved)', 'Total Units Approved', 'Approval Rate', 'Admission grade Curricular units 2nd sem (approved)', 'Admission grade Total Units Approved', 'Admission grade Approval Rate', 'Age at enrollment Curricular units 2nd sem (approved)', 'Age at enrollment Total Units Approved', 'Age at enrollment Approval Rate', 'Curricular units 1st sem (enrolled) Curricular units 1st sem (approved)', 'Curricular units 1st sem (enrolled) Curricular units 2nd sem (approved)', 'Curricular units 1st sem (enrolled) Total Units Approved', 'Curricular units 1st sem (enrolled) Approval Rate', 'Curricular units 1st sem (evaluations) Curricular units 2nd sem (approved)', 'Curricular units 1st sem (evaluations) Approval Rate', 'Curricular units 1st sem (approved) Curricular units 2nd sem (enrolled)', 'Curricular units 1st sem (approved) Curricular units 2nd sem (approved)', 'Curricular units 1st sem (approved) Total Units Enrolled', 'Curricular units 1st sem (approved) Total Units Approved', 'Curricular units 1st sem (approved) Average curricular units', 'Curricular units 1st sem (approved) Approval Rate', 'Curricular units 1st sem (approved) Economic Hardship', 'Curricular units 1st sem (approved) Unemployment rate', 'Curricular units 1st sem (approved) GDP', 'Curricular units 2nd sem (enrolled) Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (enrolled) Total Units Approved', 'Curricular units 2nd sem (enrolled) Approval Rate', 'Curricular units 2nd sem (evaluations) Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (evaluations) Approval Rate', 'Curricular units 2nd sem (approved) Total Units Enrolled', 'Curricular units 2nd sem (approved) Total Units Approved', 'Curricular units 2nd sem (approved) Average curricular units', 'Curricular units 2nd sem (approved) Approval Rate', 'Curricular units 2nd sem (approved) Economic Hardship', 'Curricular units 2nd sem (approved) Unemployment rate', 'Curricular units 2nd sem (approved) Inflation rate', 'Curricular units 2nd sem (approved) GDP', 'Total Units Enrolled Total Units Approved', 'Total Units Enrolled Approval

Rate', 'Total Units Approved Average curricular units', 'Total Units Approved Approval Rate', 'Total Units Approved Economic Hardship', 'Total Units Approved Unemployment rate', 'Total Units Approved Inflation rate', 'Total Units Approved GDP', 'Average curricular units Approval Rate', 'Approval Rate Economic Hardship', 'Approval Rate Unemployment rate', 'Approval Rate Inflation rate', 'Approval Rate GDP'

- Recursive Feature Elimination (RFE): Recursive Feature Elimination (RFE) is a feature selection technique that recursively removes the least important features based on a model's performance. It is a wrapper method that selects features by considering a machine learning model's predictive power and eliminating features in a backward elimination fashion. Logistic Regression was utilized as the estimator to iteratively select 50 features, yielding a refined set of features. These features includes 'Age at enrollment', 'Curricular units 2nd sem (credited)', 'Curricular units 2nd sem (enrolled)', 'Curricular units 2nd sem (evaluations)', 'Approval Rate', 'Improvement in Grades', 'Total Units without Evaluations', 'Admission grade Curricular units 1st sem (enrolled)', 'Admission grade Curricular units 1st sem (evaluations)', 'Admission grade Curricular units 1st sem (without evaluations)', 'Admission grade Curricular units 2nd sem (evaluations)', 'Admission grade Curricular units 2nd sem (without evaluation)'.
- Random Forest Feature Importance: The Random Forest Classifier provided insights into feature importance, identifying the top features that contribute to the model's predictions. These features include 'Father's occupation', 'Age at enrollment', 'Total Units Enrolled Approval Rate', 'Mother's occupation', 'Age at enrollment Curricular units 2nd sem (enrolled)', 'Admission grade Economic Hardship', 'Curricular units 1st sem (evaluations) Average curricular units', 'Curricular units 1st sem (enrolled) Curricular units 2nd sem (approved)', 'Admission grade Curricular units 2nd sem (enrolled)', 'Age at enrollment Unemployment rate', 'Admission grade GDP', 'Approval Rate Economic Hardship', 'Admission grade Unemployment rate', 'Admission grade Inflation rate', 'Previous

qualification (grade)', 'Curricular units 2nd sem (evaluations) Average curricular units', 'Admission grade', 'Age at enrollment Curricular units 1st sem (approved)', 'Age at enrollment Average curricular units', 'Curricular units 2nd sem (enrolled) Total Units Approved', 'Average curricular units', 'Curricular units 2nd sem (approved) Total Units Enrolled', 'Curricular units 2nd sem (approved) Unemployment rate', 'Curricular units 1st sem (approved) Average curricular units', 'Curricular units 1st sem (approved) Total Units Approved', 'Age at enrollment Curricular units 2nd sem (approved)', 'Admission grade Age at enrollment', 'Curricular units 2nd sem (enrolled) Approval Rate', 'Total Units Approved', 'Curricular units 1st sem (evaluations) Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (evaluations) Curricular units 2nd sem (approved)', 'Curricular units 1st sem (approved) Approval Rate', 'Approval Rate Unemployment rate', 'Curricular units 2nd sem (approved) Economic Hardship', 'Tuition fees up to date', 'Total Units Approved Average curricular units', 'Curricular units 2nd sem (approved)', 'Age at enrollment Approval Rate', 'Curricular units 1st sem (enrolled) Approval Rate', 'Curricular units 1st sem (approved) Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (approved) Average curricular units', 'Curricular units 2nd sem (enrolled) Curricular units 2nd sem (approved)', 'Admission grade Total Units Approved', 'Admission grade Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (approved) Approval Rate', 'Curricular units 2nd sem (approved) Total Units Approved', 'Total Units Approved Approval Rate', 'Admission grade Approval Rate', 'Average curricular units Approval Rate', 'Approval Rate'

- Lasso Regression: Lasso regression, short for Least Absolute Shrinkage and Selection Operator, is a regularization technique used in linear regression that is particularly effective for feature selection. It achieves this by introducing a penalty term to the loss function that encourages the coefficients of less important features to be shrunk towards zero. The features selected using this method include 'Curricular units 1st sem (enrolled) Curricular units 2nd sem (without evaluations)',

'Curricular units 1st sem (credited) Unemployment rate', 'Curricular units 1st sem (credited) Total Units Enrolled', 'Curricular units 1st sem (approved) Curricular units 2nd sem (enrolled)', 'Curricular units 1st sem (credited) Curricular units 2nd sem (approved)', 'Curricular units 1st sem (enrolled) Curricular units 1st sem (evaluations)', 'Curricular units 1st sem (enrolled) Curricular units 1st sem (approved)', 'Curricular units 1st sem (enrolled) Curricular units 1st sem (without evaluations)', 'Curricular units 1st sem (approved) Curricular units 2nd sem (credited)', 'Curricular units 1st sem (enrolled) Curricular units 2nd sem (credited)', 'Curricular units 1st sem (enrolled) Curricular units 2nd sem (enrolled)', 'Curricular units 1st sem (enrolled) Curricular units 2nd sem (evaluations)', 'Curricular units 1st sem (enrolled) Curricular units 2nd sem (approved)', 'Curricular units 1st sem (credited) GDP', 'Curricular units 1st sem (credited) Curricular units 2nd sem (without evaluations)', 'Curricular units 2nd sem (enrolled) Unemployment rate', 'Improvement in Grades Inflation rate', 'Curricular units 1st sem (evaluations) Average curricular units', 'Father's qualification', 'Curricular units 1st sem (approved) Curricular units 1st sem (without evaluations)', 'Admission grade', 'Curricular units 1st sem (without evaluations) Improvement in Grades', 'International', 'Curricular units 1st sem (grade)', 'Total Units Approved Approval Rate', 'Curricular units 2nd sem (credited) Approval Rate', 'Improvement in Grades GDP', 'Curricular units 1st sem (credited) Inflation rate', 'Curricular units 1st sem (credited) Approval Rate', 'Curricular units 2nd sem (credited) Inflation rate', 'Curricular units 2nd sem (credited) Improvement in Grades', 'Previous qualification', 'Curricular units 1st sem (without evaluations) Curricular units 2nd sem (without evaluations)', 'Mother's occupation', 'Approval Rate Improvement in Grades', 'Age at enrollment Unemployment rate', 'Gender', 'Curricular units 2nd sem (approved) Inflation rate', 'Course', 'Age at enrollment Curricular units 2nd sem (enrolled)', 'Curricular units 2nd sem (credited)', 'Debtor', 'Curricular units 2nd sem (evaluations) Average curricular units', 'Age at enrollment Total Units Enrolled', 'Scholarship holder', 'Approval Rate', 'Curricular units 2nd sem

(enrolled)', 'Tuition fees up to date', 'Average curricular units Approval Rate', 'Curricular units 2nd sem (approved) Approval Rate'

4. Visualization of Dimensionality Reduction Results

Dimensionality reduction techniques such as PCA and t-SNE were applied to visualize the data:

- PCA Visualization: PCA was performed to reduce the dimensionality of the dataset while preserving variance. To determine the optimum number of components, a graph was plotted i.e graph of explained variance vs No of components. This graph helps give an insight to how much explained variance can be sacrificed for a given number of components. After analyzing this graph, 50 PCA components was taken as the optimum.

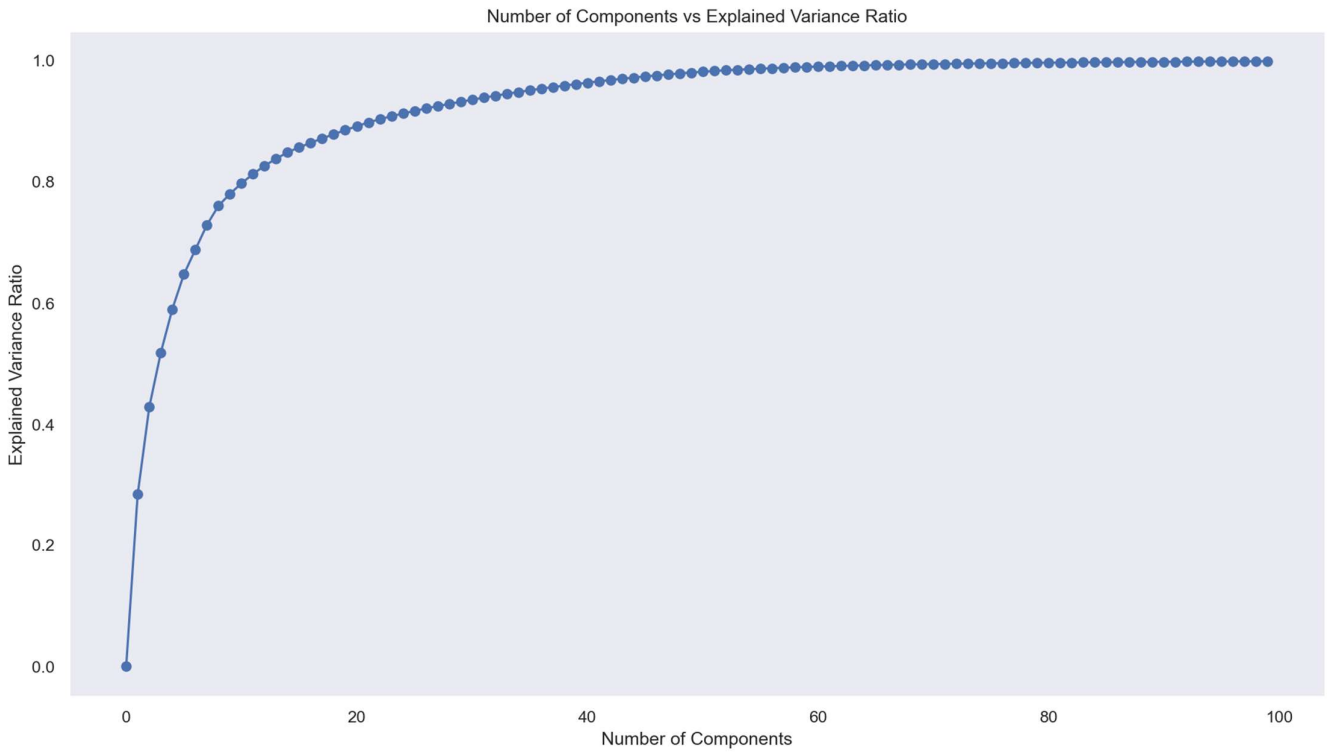

Fig 6: Graph of Explained variance ratio vs Number of components.

- t-SNE Visualization: t-SNE is particularly effective for visualizing high-dimensional data in two dimensions. The resulting scatter plot provides insights into the clusters formed by different classes, highlighting the separability of the data.

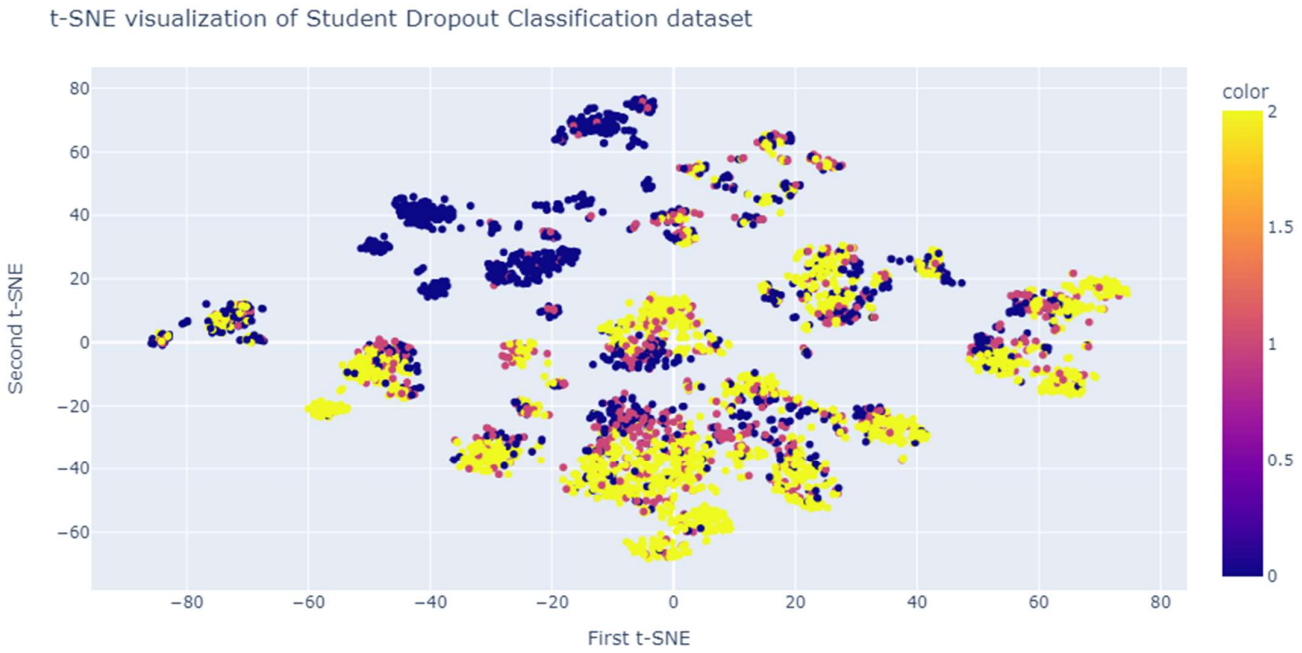

Fig 7: T-SNE Visualization