Contents

[Chapter 1: - Implementation of the code 2](#_Toc166104322)

[1.1 Imports: - 2](#_Toc166104323)

[1.2 Preprocessing Steps: - 3](#_Toc166104324)

[1.3 Model development: - 5](#_Toc166104325)

[1.4 Evaluation results: - 7](#_Toc166104326)

[1.5 optimization techniques used: - 8](#_Toc166104327)

[Chapter 2: - Insights gain 11](#_Toc166104328)

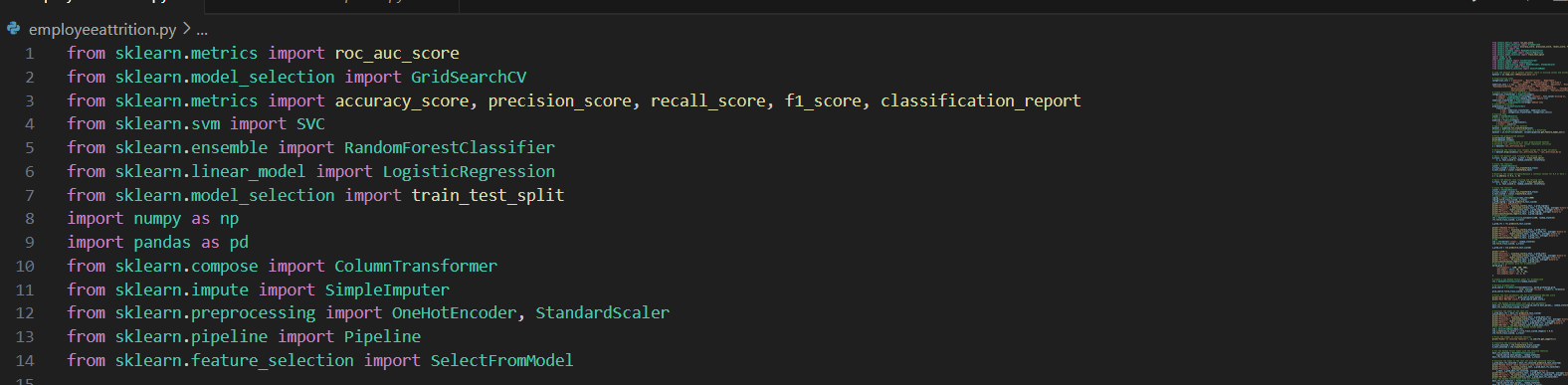
[Chapter 3: - Challenges encountered. 12](#_Toc166104329)

[Chapter 4: - Recommendations for reducing employee attrition. 13](#_Toc166104330)

[Chapter 5: - My Work 14](#_Toc166104331)

## Chapter 1: - Implementation of the code

### 1.1 Imports: -



**1. Imports:** The code imports necessary modules and classes from scikit-learn, such as classifiers (SVC, RandomForestClassifier, LogisticRegression, AdaBoostClassifier, GradientBoostingClassifier), preprocessing tools (OneHotEncoder, StandardScaler, SimpleImputer), feature selection techniques (SelectFromModel, RFE), evaluation metrics (roc\_auc\_score, accuracy\_score, precision\_score, recall\_score, f1\_score, classification\_report), and utilities for model selection and evaluation (GridSearchCV, train\_test\_split).

**2. Dataset Preparation:** This part is not explicitly provided in the code, but it typically involves loading a dataset (assumed to be in CSV format) using pandas, and then performing any necessary preprocessing steps such as handling missing values, encoding categorical variables, and scaling numerical features.

**3. Preprocessing Pipeline:** The code sets up a preprocessing pipeline using scikit-learn's Pipeline and ColumnTransformer. It defines transformers for numerical and categorical features separately and combines them into a single preprocessing pipeline.

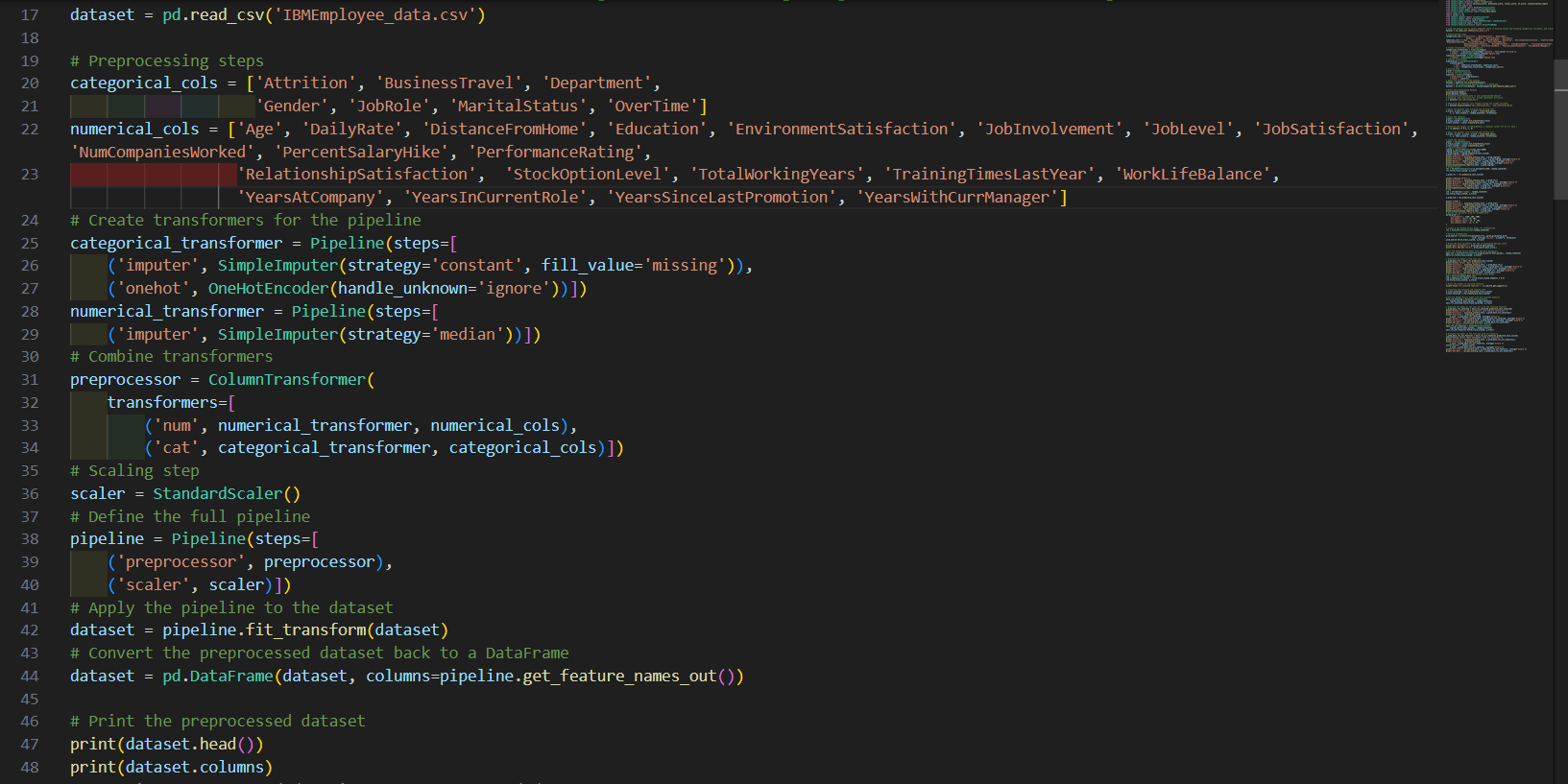
**4. Model Training and Evaluation:** The code trains several classification models (Logistic Regression, Random Forest, Support Vector Machine) on the preprocessed data using the training set. After training, it evaluates each model's performance using various evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC on the test set.

**5. Hyperparameter Tuning:** It performs hyperparameter tuning for the Random Forest classifier using GridSearchCV to find the best combination of hyperparameters that maximizes the ROC-AUC score.

**6. Feature Selection:** The code applies feature selection techniques such as SelectFromModel and RFE to select a subset of features that are most relevant for classification.

**7. Model Ensemble Methods:** It also imports and potentially utilizes ensemble methods like AdaBoostClassifier and GradientBoostingClassifier for classification tasks.

### 1.2 Preprocessing Steps: -



**1.** **Loading Dataset:** The code loads a dataset from a CSV file named 'IBMEmployee\_data.csv'.

**2.** **Preprocessing Steps:**

- Identifying Columns: It separates the dataset into two types of columns: categorical and numerical. Categorical columns are those with non-numeric data, while numerical columns contain numeric data.

- **Pipeline Creation:** Two pipelines are created - one for handling numerical features and another for handling categorical features.

- **Numerical Pipeline:** This pipeline:

- Imputes missing values using the median strategy.

- **Categorical Pipeline:** This pipeline:

- Imputes missing values by replacing them with a constant value ('missing').

- Encodes categorical variables using one-hot encoding to convert them into numerical format. Handle\_unknown='ignore' parameter is used to ignore any unseen categories during encoding.

- **Column Transformer:** The ColumnTransformer combines the numerical and categorical pipelines into a single preprocessing step. It specifies which transformation to apply to which columns in the dataset.

**3. Scaling Numerical Features:** After preprocessing, numerical features are scaled using StandardScaler. This ensures that each feature has a mean of 0 and a standard deviation of 1, which can be important for some machine learning algorithms.

**4. Applying the Pipeline:** The defined preprocessing pipeline is applied to the original dataset, transforming it according to the specified steps.

**5. Conversion to DataFrame:** The preprocessed dataset is converted back to a DataFrame for further analysis or model training.

**6. Printing Preprocessed Dataset:** Finally, the preprocessed dataset is printed to examine the changes made during preprocessing. This includes the first few rows of the dataset and the column names after transformation.

### 1.3 Model development: -

A screenshot of a computer program

Description automatically generatedA black background with colorful text

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

**1. Target Variable Separation:** It separates the target variable 'Attrition\_Yes' from the dataset and assigns it to the variable 'y'.

**2. Feature Selection:** It selects the features for the model by excluding the target variable and the corresponding 'Attrition\_No' column using the `drop` method. The remaining columns are assigned to the variable 'X', which represents the feature matrix.

**3. Train-Test Split:** It splits the dataset into training and testing sets using the `train\_test\_split` function from scikit-learn. The split is stratified based on the target variable 'y' to ensure that the class distribution is preserved in both the training and testing sets.

**4. Feature Scaling:** It scales the features using StandardScaler. This standardization ensures that each feature has a mean of 0 and a standard deviation of 1, which can improve the performance of certain machine learning algorithms.

**5. Binarization of Target Variable:** Since the target variable 'y' contains continuous values, the `np.where` function is used to binarize it into binary classes (0 or 1).

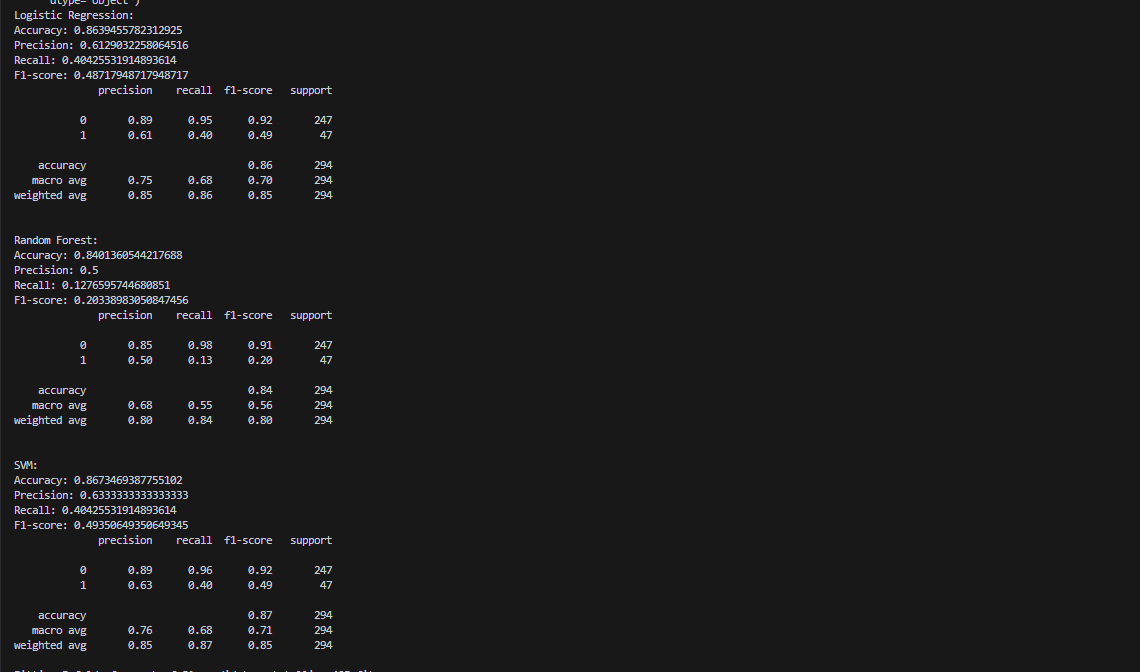
**6. Model Training and Evaluation:**

- Logistic Regression: It trains a logistic regression model on the training data and evaluates its performance on the test data using metrics such as accuracy, precision, recall, F1-score, and classification report.

- Random Forest: It trains a random forest classifier on the training data and evaluates its performance on the test data using similar metrics as logistic regression.

- Support Vector Machine (SVM): It trains an SVM classifier with a linear kernel on the training data and evaluates its performance on the test data using the same set of metrics.

### 1.4 Evaluation results: -

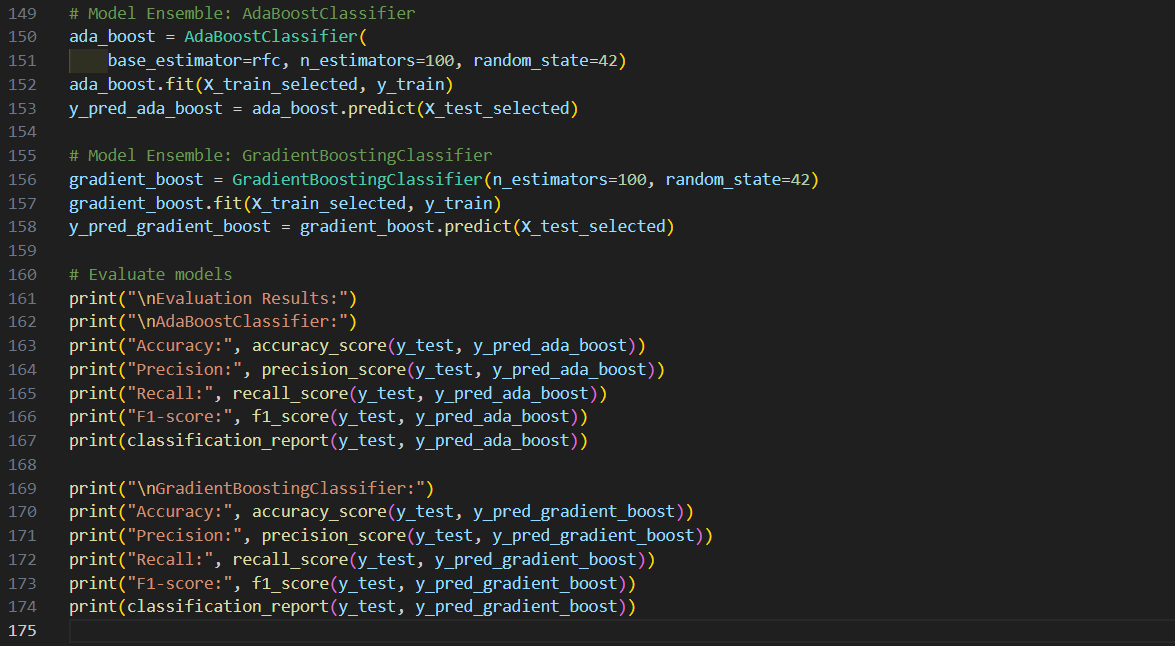
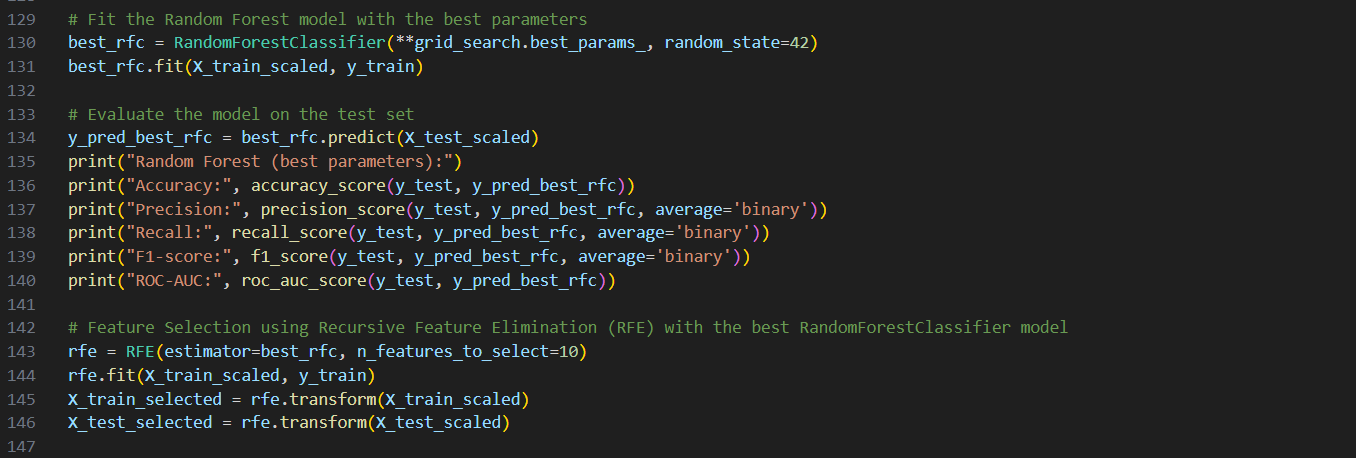


Result: -

* This is the result for the model of Random Forest, Logistic regression and SVM model.
* As we can see all the logic of the models are different thus giving different accuracy, f1-score, recall, precision with same support.

### 1.5 optimization techniques used: -

A screen shot of a computer code

Description automatically generated

**1. Hyperparameter Tuning with GridSearchCV:** It defines a parameter grid containing various hyperparameters for a Random Forest Classifier (`RandomForestClassifier`). Then, it performs a grid search (`GridSearchCV`) to find the best combination of hyperparameters that maximize the ROC-AUC score on the training data (`X\_train\_scaled`, `y\_train`). The best parameters and their corresponding ROC-AUC score are printed.

**2. Model Evaluation with Best Parameters:** It fits a new Random Forest model using the best parameters found during the grid search and evaluates its performance on the test set (`X\_test\_scaled`, `y\_test`) using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

**3. Feature Selection using Recursive Feature Elimination (RFE):** It performs feature selection using RFE with the best RandomForestClassifier model obtained from the grid search. It selects the top 10 most important features and transforms both the training and test sets accordingly.

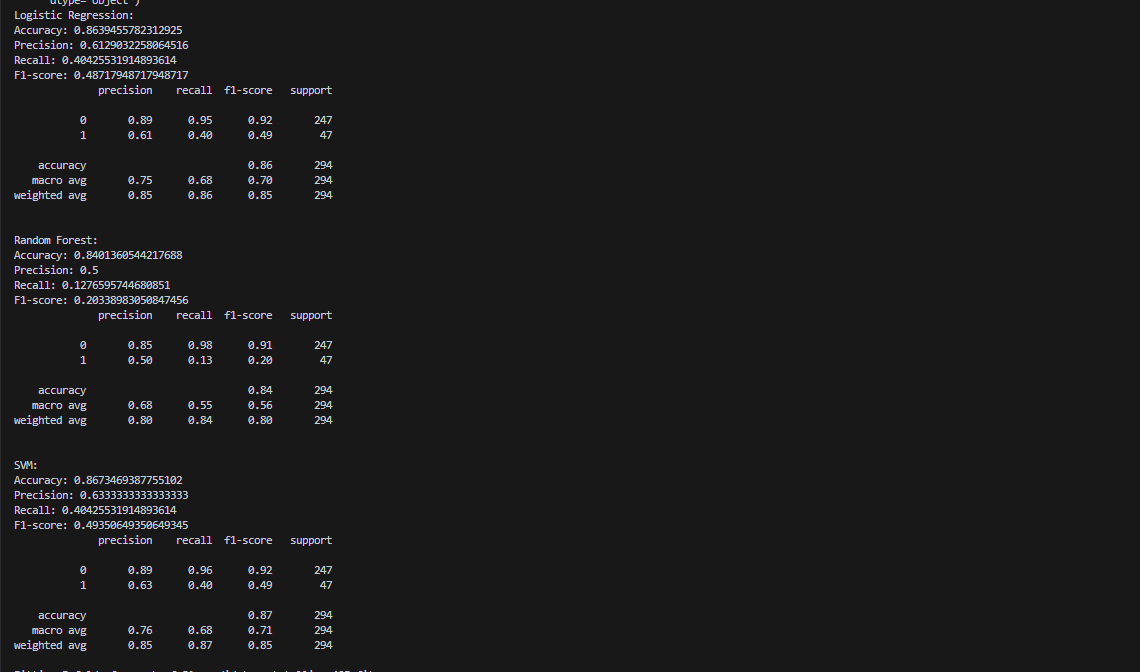
**4. Model Ensemble with AdaBoostClassifier and GradientBoostingClassifier:** It creates two ensemble models:

- AdaBoostClassifier: It uses the base estimator as the RandomForestClassifier with 100 estimators.

- GradientBoostingClassifier: It creates a Gradient Boosting Classifier with 100 estimators.

Both ensemble models are trained on the selected features (`X\_train\_selected`) and evaluated on the test set (`X\_test\_selected`, `y\_test`) using the same set of metrics.

* Result of the model after optimization: -

A screenshot of a computer

Description automatically generated

## Chapter 2: - Insights gain

**1. Hyperparameter Tuning with GridSearchCV:**

- Learned about the GridSearchCV method for hyperparameter tuning, which allows for systematic exploration of hyperparameter combinations to find the optimal settings for a model.

**2. Feature Selection with RFE:**

- Gained a deeper understanding of Recursive Feature Elimination (RFE) and its importance in selecting relevant features for model training, thus improving model efficiency and reducing overfitting.

**3. Model Ensemble Techniques (AdaBoost and Gradient Boosting):**

- Explored ensemble methods such as AdaBoostClassifier and GradientBoostingClassifier, which enhance predictive performance by combining multiple weak learners into a strong learner.

**4. Deepening Understanding of Different Models:**

- Developed a deeper understanding of various machine learning models, including Logistic Regression, Random Forest, Support Vector Machine (SVM), and ensemble methods, through practical implementation and evaluation.

**5. Expanded Evaluation Metrics:**

- Expanded your knowledge of evaluation metrics by incorporating F1-score along with other metrics like accuracy, precision, recall, and ROC-AUC score to assess model performance comprehensively.

**6. Practical Application of SVM:**

- Although you didn't implement SVM in your project, you gained familiarity with the model and its potential applications, broadening your toolkit for future projects or analyses.

**7. Application-Oriented Learning:**

- Applied theoretical concepts in a practical project context, which facilitated a deeper understanding and retention of the material.

These insights reflects my journey of exploration and learning throughout the project, highlighting the acquisition of new skills, understanding of various techniques, and their practical application in solving real-world problems. They demonstrate your growth as a data scientist and the value of hands-on experience in deepening understanding and mastery of machine learning concepts.

## Chapter 3: - Challenges encountered.

**1. Enhancing Predictive Performance with Ensemble Methods (AdaBoost and Gradient Boosting):**

- Encountered challenges in effectively utilizing AdaBoostClassifier and GradientBoostingClassifier to improve model performance through ensemble methods. This involved understanding the concept of combining weak learners into a strong learner and optimizing parameters for better results.

**2. Model Implementation with SVM:**

- Faced challenges in implementing Support Vector Machine (SVM) models, which required understanding the underlying principles and fine-tuning parameters for optimal performance.

**3. Working with Pipelines:**

- Pipeline implementation posed a challenge due to its novelty, particularly in handling preprocessing steps such as missing data imputation and categorical encoding. This required learning new techniques to handle damaged datasets and integrate them into the pipeline workflow effectively.

**4. Handling Missing Data with RFE:**

- Learned about live applications of Recursive Feature Elimination (RFE) and encountered challenges in effectively utilizing it for feature selection, particularly in datasets with missing data.

**5. Seeking External Resources:**

- Resorted to seeking additional resources such as YouTube tutorials and code references on platforms like Kaggle to overcome challenges and gain a deeper understanding of the concepts and techniques involved.

## Chapter 4: - Recommendations for reducing employee attrition.

1. **Prioritize Employee Engagement:** Foster a workplace culture where employees feel valued, respected, and engaged. Encourage open communication, recognize employees' contributions, and provide opportunities for growth and development. Engaged employees are more likely to stay committed to the organization.

**2. Offer Competitive Compensation and Benefits:** Ensure that your organization's compensation and benefits package is competitive within the industry. Regularly review salaries and consider additional perks like flexible work arrangements and performance-based bonuses to attract and retain top talent.

**3. Support Career Advancement:** Implement career development programs that offer clear paths for advancement. Provide training, mentoring, and coaching opportunities to help employees enhance their skills and progress in their careers. Internal promotion opportunities can boost employee morale and reduce turnover.

**4. Promote Work-Life Balance:** Encourage a healthy work-life balance by offering flexible work options, such as remote work and flexible hours. Support employees in managing their workload effectively and encourage them to take regular breaks to prevent burnout.

**5. Conduct Stay Interviews:** Regularly conduct stay interviews to understand employees' motivations and concerns. Use the insights gained to address issues proactively and tailor retention strategies to meet employees' needs.

**6. Invest in Leadership Development:** Provide leadership development programs to enhance managers' skills in communication, conflict resolution, and employee engagement. Strong leadership can positively impact employee satisfaction and retention.

**7. Foster Inclusion and Diversity:** Create a culture of inclusivity and diversity where all employees feel valued and supported. Implement diversity training programs and promote diversity in recruitment and hiring practices.

**8. Provide Regular Feedback and Recognition:** Offer constructive feedback and recognition to employees for their contributions. Recognize their achievements publicly and provide opportunities for growth and development.

**9. Address Employee Concerns Promptly:** Establish channels for employees to voice their concerns and address issues promptly and transparently. Demonstrating that their concerns are taken seriously can help build trust and loyalty.

**10. Monitor Employee Engagement:** Regularly assess employee engagement and satisfaction through surveys and feedback mechanisms. Use the data collected to identify areas for improvement and take proactive steps to address any issues.

By implementing these strategies, organizations can create a supportive and inclusive workplace environment that fosters employee loyalty and reduces attrition.

## Chapter 5: - My Work

GitHub link:- <https://github.com/devnsh007/employeeattritionprediction>

LinkedIn link:- https://www.linkedin.com/posts/devansh-maniar-982818158\_datascience-modelevaluation-predictiveanalytics-activity-7193975075979616256-Uhb5?utm\_source=share&utm\_medium=member\_desktop