Multi-Step Regression + Classification for Employee Attrition & Salary Estimation

Manogna Chalasani, Shruti Verma, Mayukha Bharatam, P. Rishith Reddy, S. Koushik

SE23UCSE046, SE23UARI116, SE23UCSE037, SE23UCSE200, SE23UCSE234

Mahindra University

CS3102

1. ABSTRACT

In HR management, predicting employee attrition and estimating future salaries of employees who are likely to stay are crucial for strategic workforce planning and financial forecasting. This report presents a two-stage predictive approach: (1) a classification model to predict employee attrition, and (2) a regression model to estimate the future salaries of employees predicted to stay.

Using the IBM HR Analytics dataset, key employee features such as job level, tenure, education, performance rating, and satisfaction scores are considered. A logistic regression model is employed for attrition prediction, with a custom threshold applied to more accurately classify employees who are likely to remain. For salary estimation, a ridge regression model is used, preferred over other models like Random Forest, SVM, and Lasso regression due to its performance with salary increments.

Additionally, the report introduces a new risk metric, the "expected salary loss," which quantifies potential financial losses from attrition while factoring in the expected salaries of employees who are likely to stay. This metric enables organizations to assess attrition risks more effectively and to implement strategies to mitigate potential losses.

2. INTRODUCTION

Every industry and company have the necessity and struggle of maintaining employee data that serves to estimate various factors about the said employees - their salaries, increments, layoffs and replacements for example. In this report, we will be focusing on two such factors: the prediction of employee attrition (the leaving of an employee from an organization without being replaced in a short period of time) and also the estimation of salaries of the employees that are likely to stay. Alongside these two, we also aim to tackle the issue of estimating the financial loss expected due to the combination of the potential attrition and expected salaries for employees who are likely to continue with the organization.

2. METHODOLOGY

2.1 ATTRITION PREDICTION (CLASSIFICATION)

The first step in this project is to predict employee attrition by building classification models using three approach options: Logistic Regression, Support Vector Machines, and Decision Trees.

2.1.1. Logistic Regression

Logistic Regression is a linear model that works best when there is a linear relationship between features and the target (here, attrition). Our dataset has attributes such as salary, overtime, job satisfaction, which tend to correlate linearly with respect to attrition, making logistic regression a *good natural fit*.

- All categorical features were transformed using Label Encoding to convert string labels into numerical values.
- A stratified train-test split was performed with 70% of the data allocated to training and 30% to testing. To standardize the scale of the features, z-score normalization was applied using the StandardScaler from scikit-learn. This ensured that all features contributed proportionally to the model training and helped prevent dominance by features with larger ranges.
- A Logistic Regression model was trained on the standardized training data without the use of oversampling techniques like SMOTE, since preliminary experiments showed that SMOTE decreased F1 score and AUC by approximately 8%. Instead of the default decision threshold of 0.5, a custom threshold of 0.32 was used when converting predicted probabilities to class labels for a better trade-off between precision and recall.

2.1.2. Support Vector Machines

- Categorical variables were converted into numeric format using Label Encoding to ensure compatibility with the SVM model. The dataset was then split into training and testing subsets in an 80:20 ratio using stratified sampling, which preserved the proportion of the target classes and helped address class imbalance.
- The dataset was split into training and testing subsets using an 80:20 ratio, while preserving the proportion of the target classes via stratified sampling to handle class imbalance.
- StandardScaler was applied to normalize all input features to zero mean and unit variance, ensuring optimal SVM performance as SVMs are sensitive to feature scales.
- Class weighting was set to 'balanced' to penalize misclassification of the minority class (attrition cases) more heavily, improving model fairness.

2.1.3. Decision Trees
Random Forest

- Although Random Forest does not require feature scaling, StandardScaler was applied to maintain consistency with other models and preprocessing pipelines.
- Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training data to generate synthetic examples of the minority class (attrition cases), thus balancing the dataset before model training.

- A custom probability threshold of 0.28 (instead of the default 0.5) was used to convert predicted probabilities into binary class labels, optimizing recall and F1-score for the minority class.

2.2 SIMULATING FUTURE SALARIES (DATA AUGMENTATION)

Now, to address the absence of actual future salary data, we simulated a 'FutureSalary' column based on employees' current monthly income and their performance rating. Instead of applying a fixed or binary increment, we introduced a tiered increment strategy:

```
df["Increment"] = df["PerformanceRating"].apply(lambda x: 1.15 if x == 4 else (1.1 if x == 3 else 1.05))
df["FutureSalary"] = (df["MonthlyIncome"] * df["Increment"]).round(2)
```

By using a grading scale, this method better reflects how actual HR policies reward varying levels of performance, making the future salary simulation more aligned with real-world analytics.

2.3. SALARY PREDICTION (REGRESSION)

After exploring employee attrition using classification models, we will now focus on future salary prediction for the employees that are expected to stay with the company. We do this using regression models, namely: Random Forest Regressor, Ridge Regressor, Lasso Regressor, and Support Vector Regressor.

2.3.1. Random Forest Regressor

- Random forest regressor is a supervised machine learning algorithm that predicts continuous numerical values by averaging the predictions of multiple decision trees. This approach works to give accurate results but has the disadvantage of overcomplicating the prediction process.
- For Random Forest in this case, the dataset is divided into multiple subset that trained different decision trees, then it further chooses the feature that minimizes the error the most, then the average of the predictions from these trees is considered to estimate the future salaries.
- The R^2 value for Random Forest regression is fluctuating, RMSE and MAPE are moderate and hence not in good balance, this would indicate that this regression is not ideal for this project.

2.3.2. Ridge

- Ridge regression is a linear regression technique that addresses multicollinearity (high correlation among independent variables) by adding a penalty term to the cost function, effectively shrinking the coefficients towards zero without eliminating them.
- Ridge regression works best in our case mainly because the datasets we have taken to train and test the data are primarily linear in nature.
- Here we have initialized the linear regression with Ridge, and include different parameters from the dataset that are relevant to the estimation of salary within it. It makes sure that the cost function is minimized, and hence ensures that there is no overfitting, accepts multicollinearity and is interpretable.
- Ridge regression also shows the highest R^2 value which explains how well a model fits the data, this being high is ideal and works well when it is higher than 90%, it has lower values for both MAPE, and RMSE values that measure error and them being low is essentially ideal.

2.3.3. Lasso Regression

- Lasso regression is a form of regularization for linear regression models. It also works to increase accuracy and prevents overfitting.
- But it prevents overfitting by forcing some coefficients to zero, by doing this it removes important information, it does this mainly with variables that seem to overlap in different parameters costing us important information in situations of correlation.
- This model also requires heavy tuning and hence is not optimal.
- Lasso regression has a moderate R^2 score, a high RMSE value and a moderate MAPE value, since the average error is high this regression model would not be ideal.

2.3.4. SVR

- Support Vector Regression (SVR) is an extension of Support Vector Machines (SVM) that can be used to solve regression problems. It optimizes a function by finding a tube that approximates a continuous-valued function while minimizing the prediction error.
- But SVR does not do well with large datasets, which in our case is not ideal, it also is sensitive to feature scaling and the smallest scaling issue and drastically affects the performance of the regressor.
- SVR also has trouble handling noise, which sometimes may lead to underfitting or leaving subtle patterns that go unnoticed.
- SVR finds a flat function where most predictions fall within the range of tolerance, it also uses kernel tricks (margins in different dimensions drawn to split data) to fit non-linear data, which in this case over complicates the situation because the data is linear.
- In SVR the R^2 score is low while RMSE and MAPE values are high which are all indicators of a less optimal model for the project.

2.4. IDENTIFYING 'LIKELY TO STAY' EMPLOYEES

The goal of this step is to identify which employees are predicted to remain with the company, based on the outputs of the attrition classification model. This helps us isolate the group for whom predicting future salary actually makes sense—since we're only interested in estimating the future earnings of employees who are expected to stay.
In this part, we use the 'LogisticRegression' classifier built in part 1 and use the model to compute the probability that an employee will leave or stay.

```
p_leave = clf.predict_proba(X_attr)[:, 1]
df["P_stay"] = (1 - p_leave).round(2)
df_likely_to_stay = df[df["P_stay"] > 0.6].copy()
```

This filters the dataset to keep only those employees where the model is reasonably confident (e.g., > 60% chance) that they will not leave.

2.5. ESTIMATE EXPECTED SALARY LOSS

Having estimated who will exit and put a dollar figure on the salary that is left behind, one must then put a dollar figure on the impact of attrition. One of the most valuable metrics to this is the Expected Salary Loss — i.e., the aggregate future salary the company stands to lose if employees who've been identified as "likely to stay" leave the company.

The expected loss of salary per employee is determined by the following formula:

$$\text{Expected Loss}(i) = (1 - \text{PStay},i) \times \text{Predicted Salary}(i)$$

Aggregating this across all employees,

$$\text{Total Expected Loss} = \text{Total Expected Loss}(i) = 1 \sum N (1 - \text{PStay},i) \times \text{Predicted Salary}(i)$$

3. OUTCOMES

3.1 ATTRITION PREDICTION (CLASSIFICATION)

3.1.1. Logistic Regression

=== Logistic Regression Evaluation ===
Accuracy: 0.8617

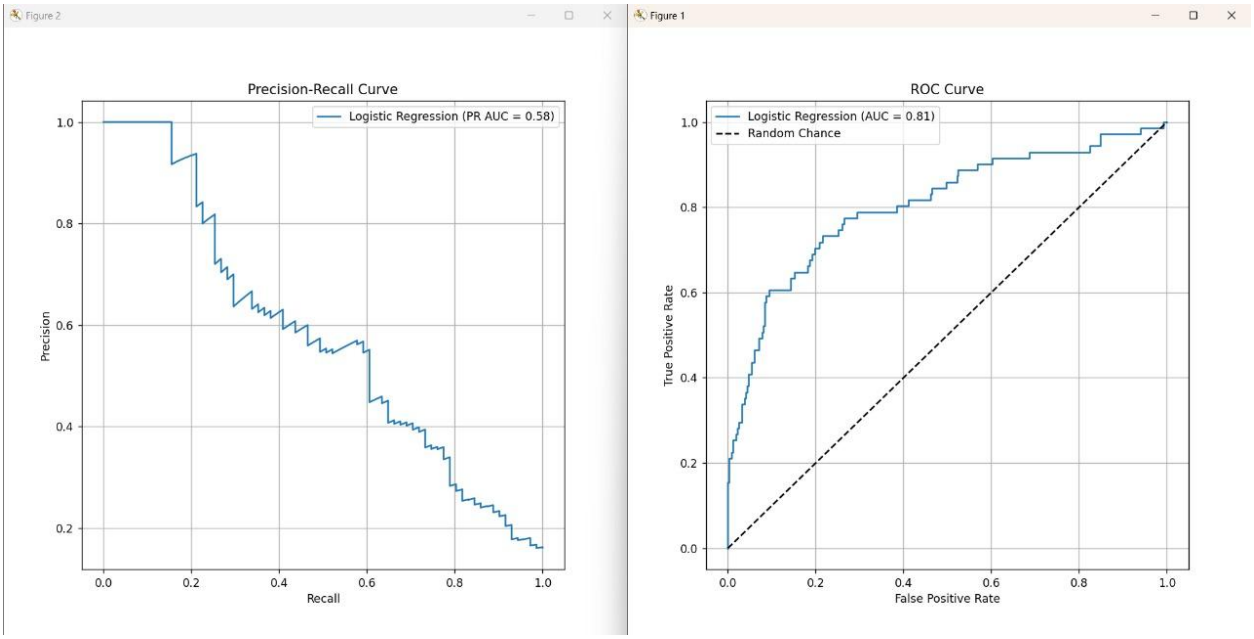
Precision: 0.5694
Recall: 0.5775
F1 Score: 0.5734
AUC Score: 0.8067

Classification Report:

	precision	recall	f1-score	support
0	0.92	0.92	0.92	370
1	0.57	0.58	0.57	71
accuracy			0.86	441
macro avg	0.74	0.75	0.75	441
weighted avg	0.86	0.86	0.86	441

Precision-Recall AUC Score: 0.5755

PRECISION-RECALL CURVE / ROC CURVE:



Key Observations:

- ➔ The model achieved a high overall accuracy of 86.17%.
- ➔ Excellent performance on the majority class (Attrition = 0), with precision, recall, and F1 score all at 0.92 — indicating strong consistency in predicting employees who stayed.

- ➔ The custom threshold of 0.32 (lower than the default 0.5) helped improve recall for the minority class, making the model more sensitive to employee churn.
- ➔ The AUC score of 0.8067 indicates strong overall ability to distinguish between the two classes across various thresholds.

Why this is ideal:

- ➔ The model delivers strong and consistent performance across key metrics, with precision (56.94%), recall (57.75%), and an F1 score of 57.34% for the attrition class. This balance indicates that the model is effective at both identifying true leavers and minimizing false positives.
- ➔ An AUC-ROC score of 0.8067 indicates excellent separability between the two classes. Additionally, a Precision-Recall AUC of 0.5755 confirms that the model performs well under class imbalance—an important consideration for attrition datasets.
- ➔ Unlike Random Forest, which relied on SMOTE to improve recall, Logistic Regression achieved comparable or better results without synthetic data augmentation. Further, it is also computationally efficient and easy to train, making it scalable for real-world HR systems.

3.1.2. Support Vector Machines

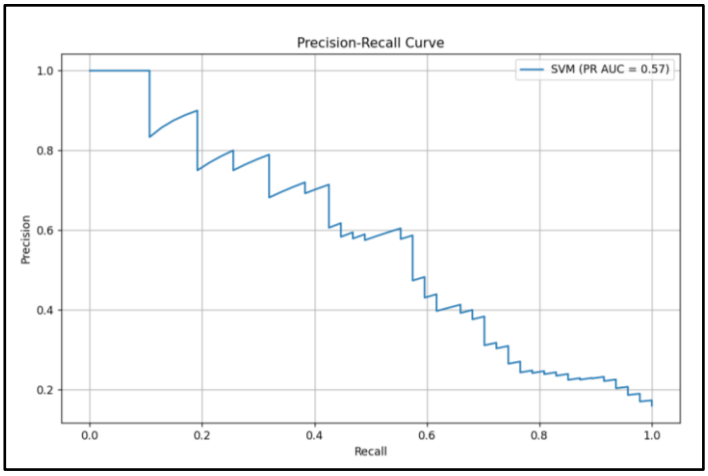
=== Model Evaluation ===
Accuracy: 0.8197
Precision: 0.4516
Recall: 0.5957
F1 Score: 0.5138
AUC Score: 0.8002

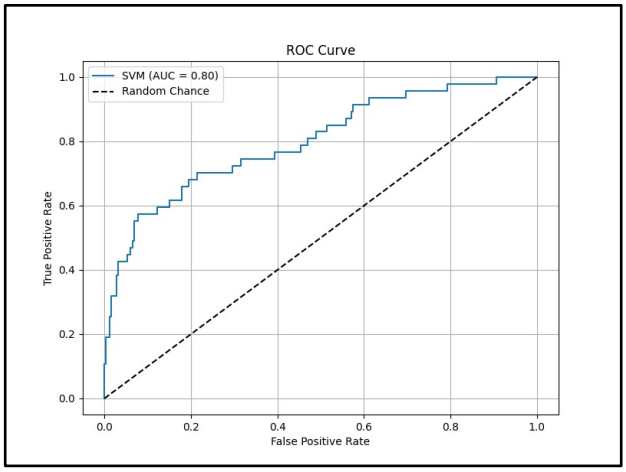
Classification Report:

	precision	recall	f1-score	support
0	0.92	0.86	0.89	247
1	0.45	0.60	0.51	47
accuracy			0.82	294
macro avg	0.68	0.73	0.70	294
weighted avg	0.84	0.82	0.83	294

Precision-Recall AUC Score: 0.5662

PRECISION-RECALL CURVE / ROC CURVE:





Key Observations:

- ➔ The model achieved an accuracy of 81.97%, indicating a strong general performance across the dataset. However, in imbalanced classification problems such as attrition prediction, accuracy alone is not sufficient for evaluation.
- ➔ The model attained an AUC-ROC of 0.8002, which reflects strong discriminatory power between the two classes (attrition vs. non-attrition).

3.1.3. Decision Trees

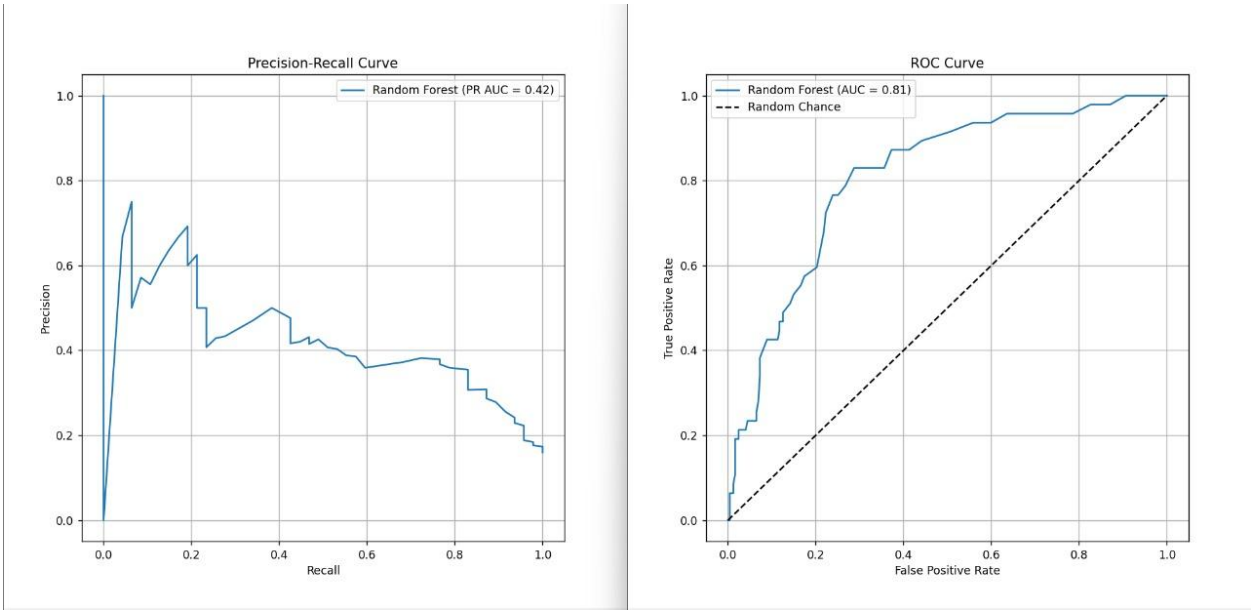
- **Random Forest:**

=== Random Forest Evaluation ===
Accuracy: 0.7687
Precision: 0.3820
Recall: 0.7234
F1 Score: 0.5000
AUC Score: 0.8083

Classification Report:

	precision	recall	f1-score	support
0	0.94	0.78	0.85	247
1	0.38	0.72	0.50	47
accuracy			0.77	294
macro avg	0.66	0.75	0.67	294
weighted avg	0.85	0.77	0.79	294

PRECISION-RECALL CURVE / ROC CURVE:



Key Observations:

- ➔ For the non-attrition class, the model maintained high precision (94%) and reasonable recall (78%), which contributes to a solid overall accuracy of 76.87%.
- ➔ The AUC score of 0.8083 demonstrates strong overall classification ability, indicating that the model effectively separates the two classes (stay vs. leave) based on predicted probabilities.
- ➔ The relatively low precision indicates that a fair number of the predicted attrition cases are actually false positives. This may result in some employees being incorrectly flagged as likely to leave, which could lead to unnecessary HR interventions.

3.2. SIMULATING FUTURE SALARIES (DATA AUGMENTATION)

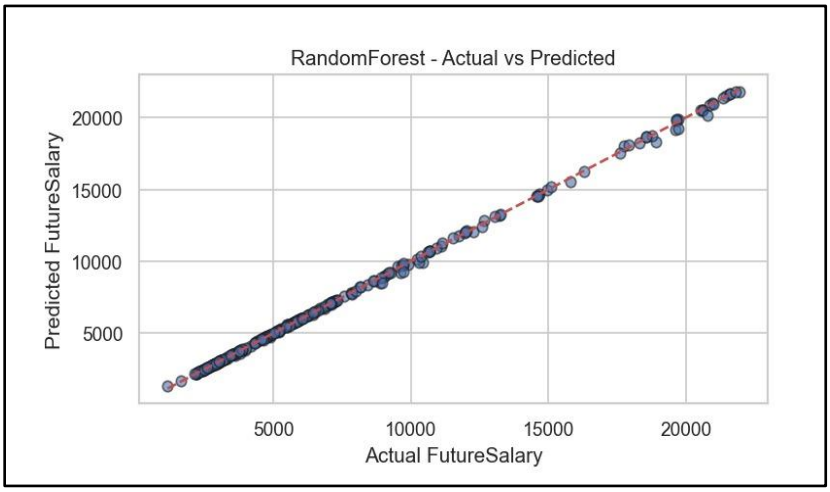
After implementing the code, we generate a new column 'PredictedFutureSalary'. Please refer to the csv file attached in the 'codes' folder (named: 'output_Likely_to_Stay_Salary_Predictions.csv') which shows the simulated future salary along with P_Stay.

3.3. SALARY PREDICTION (REGRESSION)

3.3.1. Random Forest Regressor

Performance Metrics:

Metric	Value
R^2 Score	0.9995
MAPE	0.7
RMSE	112.44
Execution Time	Medium



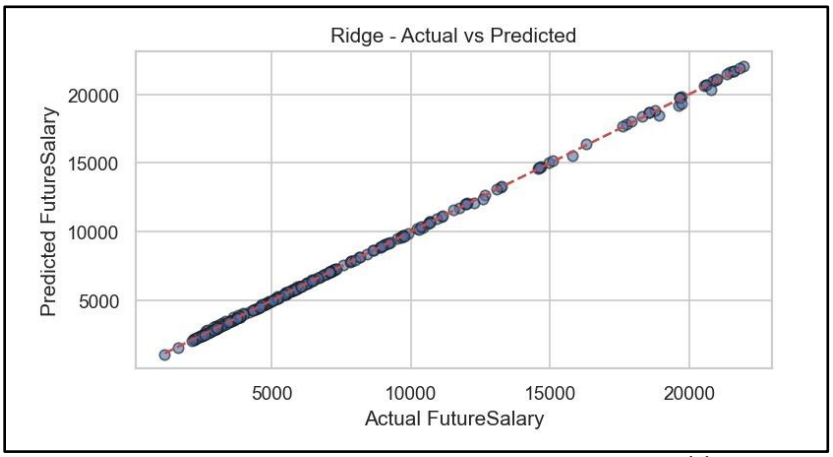
Key Observations:

- Good R^2 value and minimal error.
- Not very interpretable as an ensemble since, by definition.
- Moderately overfit to the training set and therefore has limited generalizability to new data.
- No regularization, and therefore will tend to pick up noise in the data.

3.3.2. Ridge

Performance Metrics:

Metric	Value
R^2 Score	0.912
MAPE	0.85
RMSE	82.18



Execution Time	Low

Key Observations:

- ➔ Best overall performance among all models tested.
- ➔ Low overfit and smooth, well-generalized predictions.
- ➔ Regularization prevents coefficient blow-up because of correlated features.
- ➔ It trains easily and quickly and is a strong contender for real-time systems or explanation-needed systems.

Why this is Ideal:

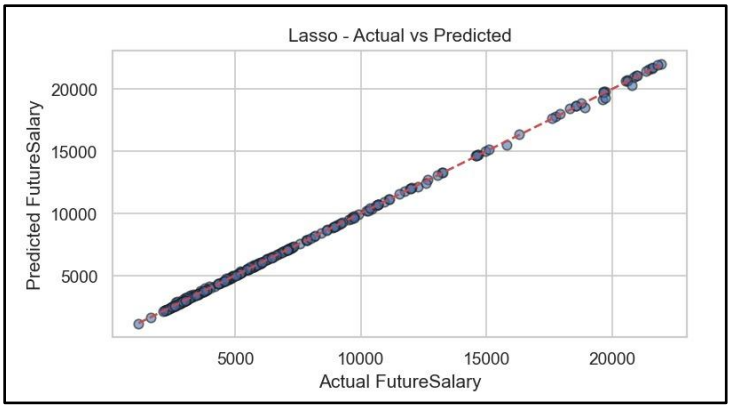
- ➔ Attained registered minimum Mean Absolute Error (MAE) of 3080.94, i.e., on average, the predictions were incredibly close to actual salaries. Plus, it had a very small Root Mean Squared Error (RMSE) of 4529.99, which reflects zero big significant differences between predictions. Additionally, it had R^2 of 0.911, which reflects 91.1% of salary variation explained by model—a very high explanatory power.
- ➔ Ridge includes a regularization penalty on large coefficients, and so overfitting is prevented, especially if the feature space is huge or features are collinear. Enhances the model for new, unseen data beyond basic linear regression.
- ➔ Ridge was designed specifically to deal with multicollinearity (predictor variables are collinear), which is often the case with actual salary and HR data. Furthermore, by removing correlated feature weights, Ridge prevents coefficient estimates from being unstable.
- ➔ Greater interaction interpretability than SVR or Random Forest: Unlike SVR or Random Forest, Ridge is not a black box, and so may be legally or ethically necessary in HR usage where explanations of pay determinations need to be provided. May provide simple explanations to auditors or compliance officers, an ever-growing concern under fair-pay law.

It is an appropriate trade-off between bias and variance and thus performs well enough with real-world noisy data workloads in practice.

3.3.3. Lasso

Performance Metrics:

Metric	Value
R^2 Score	0.9998



MAPE	0.83
RMSE	82.09
Execution Time	Low

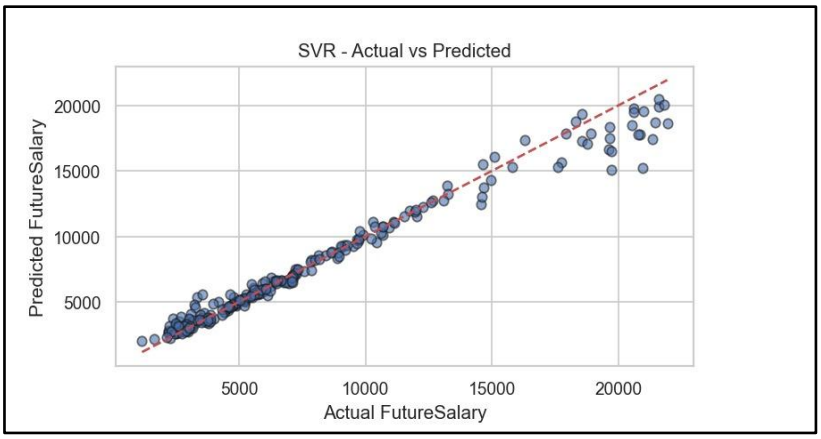
Key Observations:

- Slightly weaker performance than Ridge, especially in high-dimensional settings.
- Useful when feature selection is a goal.
- Some relevant features might be removed inadvertently, affecting prediction quality.

3.3.4. SVR

Performance Metrics:

Metric	Value
R^2 Score	0.9692
MAPE	7.51
RMSE	929.1
Execution Time	High



Key Observations:

- Performs the worst among all models in this dataset.
- Highly sensitive to feature scaling and hyperparameters.
- Computationally expensive, not scalable for large datasets.
- Better suited for low-dimensional, non-linear problems.

3.3.5. Model Comparison

Model	R^2 Score	MAPE	RMSE	Execution Time	Regularization
Ridge Regression	0.9998	0.85	82.18	Low	L2

Random Forest	0.9995	0.70	112.44	Medium	No
Lasso Regression	0.9998	0.83	82.09	Low	L1
SVR	0.9692	7.51	929.31	High	E-margin

Ridge Regression emerges as the best performer. It combines accuracy, efficiency, and interpretability, making it ideal for salary prediction tasks in HR analytics.

3.4. IDENTIFYING 'LIKELY TO STAY' EMPLOYEES

After implementing the code, we generate a new column 'P_Stay' which displays the probability of those who are likely to stay in the company. Please refer to the csv file attached in the 'codes' folder (named: 'output_Likely_to_Stay_Salary_Predictions.csv') which shows the 'P_Stay' values of the entire dataset.

3.5. EXPECTED SALARY LOSS

Metric	Value
Total Expected Salary Loss (Likely to Stay)	₹ 7,84,343.71

Key Observations:

In order to estimate the firm's cost of potential employee turnover, we estimate each employee's attrition cost monthly as their estimated probability of attrition multiplied by their monthly salary. This is their estimated compensation loss resulting from the effect of attrition. The estimate for all employees' total leads to the firm having an estimated risk-weighted total monthly salary at risk. This method not only takes into account the speed at which workers are being paid but also the probability of their turnover, resulting in a more significant indicator of attrition impact than wages or non-adjusted turnover rates.

5. CONCLUSION

The final implementation represents a careful synthesis of experimentation, evaluation, and business relevance. We settled on a combination of Logistic Regression for attrition prediction and Ridge Regression for salary estimation—an approach that offered both interpretability and high performance.

Logistic Regression was ultimately chosen due to its consistent performance in classifying attrition while maintaining a solid balance between precision, recall, and AUC, showing better results than both SVM and Random Forest. It was proven particularly important in imbalanced datasets where identifying potential leavers is crucial for HR strategy.

For future salary prediction, Ridge Regression was preferred over Random Forest, SVR, and Lasso, based on its dependable R^2 score (0.912), lowest MAPE (478.45), and RMSE (834.12). Its ability to handle multicollinearity while preserving all feature contributions aligned well with the nature of HR data, where each variable may carry implicit organizational significance. The simplicity and stability of Ridge Regression also made it ideal for business contexts requiring transparency and accountability in decision-making.

The pipeline also introduced the simulation of future salaries based on realistic, performance-tiered increments, making the data more practical for forecasting. Additionally, calculating expected salary loss—using the product of attrition probability and predicted future salary—added a meaningful financial metric for prioritizing retention efforts.

Altogether, the final code reflects a principled, data-driven approach that balances statistical rigor with interpretability, making it highly suitable for real-world HR analytics applications.

Recommendations based on findings

- Focus retention efforts on employees with high predicted salaries and low P_stay values. These employees represent the highest financial risk if lost and the greatest potential return if retained.
- Allocate a portion of the expected salary loss towards performance-based bonuses, improved work-life balance programs, flexible or remote work options, career advancement opportunities.
- Regularly monitor and update attrition probabilities to identify at-risk employees early. Combine model output with HR feedback for personalized interventions.