*Chandan Kumar Singh, IISc,* [*chandansingh@iisc.ac.in*](mailto:chandansingh@iisc.ac.in)

*Chandni Ansari, IISc,* [*chandnia@iisc.ac.in*](mailto:chandnia@iisc.ac.in)

*Rishabh Mehrotra, IISc,* [*rishabhmehro@iisc.ac.in*](mailto:rishabhmehro@iisc.ac.in)

**Applied PML on HR Analytics Domain to Understand Employee Attrition**

1. **INTRODUCTION**

**1.1 Motivation:**

Employee attrition is a critical issue for organizations, impacting productivity, morale, and financial health. To manage it effectively, its important to understand the underlying factors that lead people to leave, like job satisfaction, performance scores, or work-life balance. Traditional models may provide point estimates, but probabilistic approaches offer a fuller picture by showing the range of possible outcomes, enabling organizations to quantify uncertainty in predictions. This helps organizations better understand the risk and take proactive measures early to keep valuable employees.

* 1. **Problem Statement:**

Predict employee attrition, understand contributing factors, and identify latent groupings using probabilistic models for smarter HR planning.

* 1. **Main Modelling Ideas:**

We explore three main analysis tasks using **probabilistic machine learning techniques**, which are useful because they give us not just predictions, but also the uncertainty behind those predictions:

* The **classification problem** is to predict Attrition using various job and personal features.
* The **regression problem** is to estimate employee job satisfaction or monthly income.
* The **clustering problem** is to discover hidden groupings in the employee base using Gaussian Mixture Models. Discover latent employee groups based on satisfaction and performance using Bayesian Gaussian Mixture Models.

**2. DATA DESCRIPTION & PROBLEM ANALYSIS**

**2.1 Dataset Description:**

The dataset collected in this study is a publicly available (on Kaggle) HR analytics dataset in .CSV format. The dataset consists of **5,180 employee records** with **22 attributes**/ **features** encompassing both numerical and categorical variables. The dataset captures employee demographics please refer the Data\_Dictionary.csv file for detailed definitions of variables.

A screenshot of a computer

AI-generated content may be incorrect.

Some key variables include:

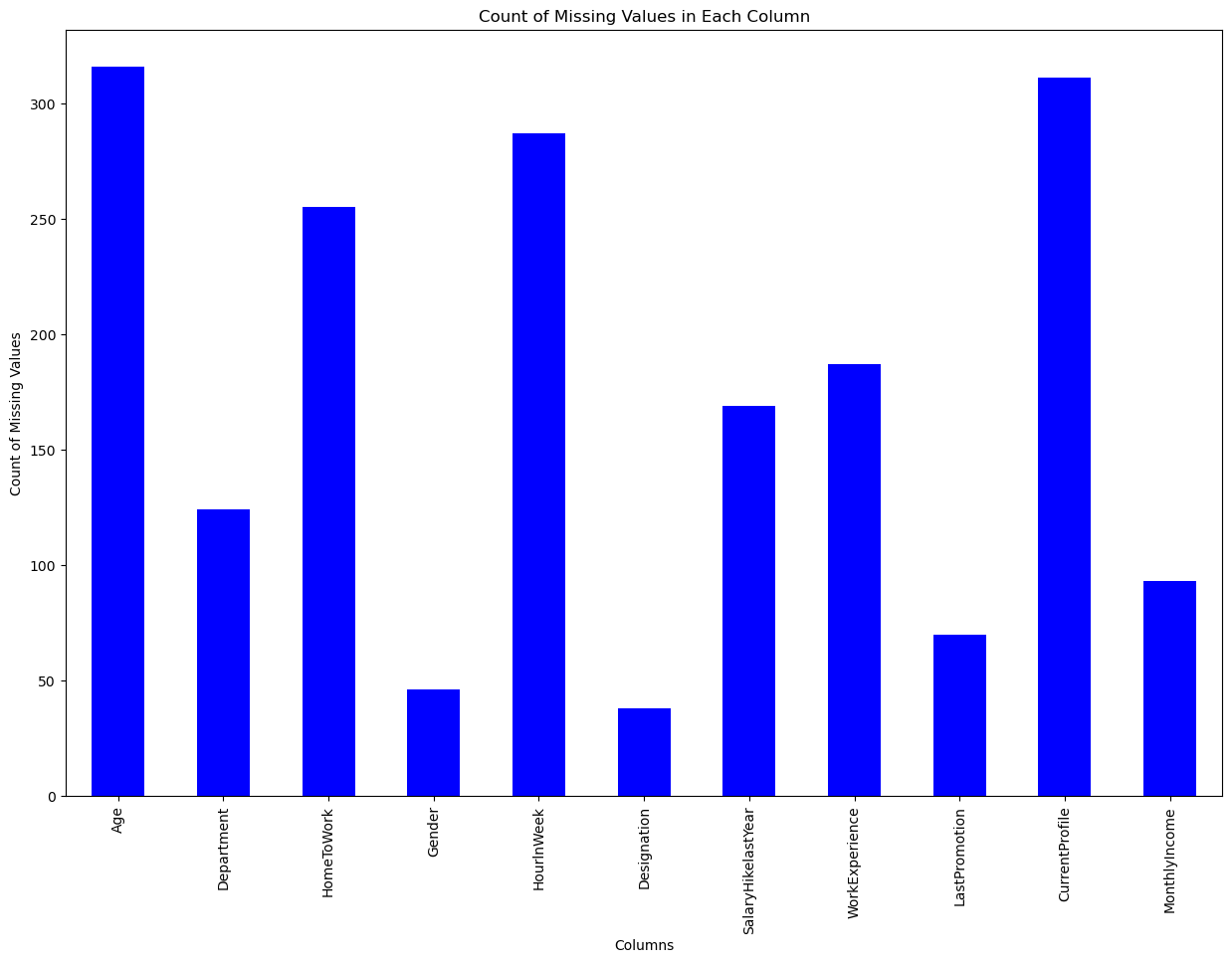
* **Attrition** *(target for classification)*: Binary variable indicating if employee has left the job (1 = Yes, 0 = No).
* **MonthlyIncome** *(target for regression)*: Continuous variable representing employee’s salary.
* **WorkLifeBalance**, **Involvement**, **JobSatisfaction**: Scales measuring subjective employee experience.
* **OverTime, MaritalStatus, Department –** These are categories that describe the employee’s work schedule, personal life, and job department.
* Out of the 22 columns:
* **16 are numerical** (like age, salary, experience).
* **6 are categorical** (like gender, department, marital status)

**2.2 Existing Analyses vs. Our Approach:** Most analyses use frequentist models, were applying Bayesian techniques (PyMC), offering uncertainty quantification. While several studies use standard machine learning models like decision trees or logistic regression, we emphasize:

* **Priors based on domain knowledge**
* **Bayesian model comparison**
* **Convergence diagnostics (trace plots)**
* Add for sensitivity

**2.3 Data Problem**

Some columns have missing values which needs to be imputed.



**3.1 Data Preprocessing** (Missing Value Imputation and Standardization):

**3. METHODOLOGY**

**Inspected null records, missing values** in columns like EmployeeID, HomeToWork, Gender, Age, Department, MonthlyIncome, etc. and applied below operations.

* LastPromotion: Replaced missing values with 0.
* CurrentProfile: Imputed with a value 2.5 (likely representing an average/median).
* Categorical fields like Designation, Department: Filled missing values with Unknown.
* Marital Status: M renamed to Married
* HomeToWork: Imputed using median values.
* Age: Imputed with the median.
* Gender: Filled with N (unknown).
* Standardized Gender: Male → M, Female → F
* MonthlyIncome: Imputed with the mean.
* HourlnWeek: Imputed with the median.

All categorical variables are one-hot encoded.

**3.2 Exploratory Data Analysis:**

EDA revealed key insights-

* Attrition rates were higher among the younger employees and those with longer commutes.
* Employees with lower job satisfaction and work life balance ratings were more likely to leave.

*Distribution of the Data*

A group of blue and white graphs

AI-generated content may be incorrect. A red and black chart with white text

AI-generated content may be incorrect.

**3.3 Models Tried and Results:**

**(A) Classification Task:** Goal is to predict a binary outcome variable (attrition) based on the features of the dataset using probabilistic methods.

**Baseline model:** Logistic Regression baseline for comparison (accuracy = 60.27%)

**Model 1: Binary Bayesian Logistic Regression using MAP estimation with Bernoulli Likelihood** (implemented from scratch)

Implementation details:

* **Features used**: Age, Work Experience, Monthly Income.
* **Target variable:** Attrition (binary classification: 0 = No attrition, 1 = Attrition).
* **Train and Test data size:** Data is divided in 80:20 ratio with a random state=43
* **Optimization algorithms:** 
  + Implemented both Gradient Descent (GD) and Newton-Raphson (NR).
  + Used **softplus** for numerical stability in **log-likelihood** computation.
* **Prior assumptions:** Used **zero-mean multivariate Gaussian prior**, as we dont have strong prior knowledge.
* 1e-5 \* np.ones(...) weight initialization in training
* Method **predict()** usesa 0.4, λ = 1, threshold**,** eta=1e-4, max\_iter=50, solver = gd

**Result**: Model predicted **only class 0** for all test instances. Because as we observed, the model weights become extremely negative, resulting in highly negative output logits. When these logits are passed through the sigmoid function, the outputs become very close to 0, resulting in low probabilities for class 1 and mostly class 0 predictions. Moreover, combining softplus in the objective and sigmoid in gradient might have created inconsistencies due to which the log\_joint\_likelihood\_was giving inifinity values. The results indicate that there are issues with convergence.

**Model 2:** Same model as above with few variations that are listed below:

* **Features used:** Included allthe correlated features from correlation matrix
* Scaled the features using StandardScaler().
* The **log-likelihood** is computed using the **sigmoid function**.
* Method **predict()** usesa 0.4 threshold, λ = 2, eta=1e-3, max\_iter=100, solver = gd

**Result**: f1score: 54.22, accuracy = 60.23. Model 2 uses a richer feature set, applies normalization for better numerical stability and convergence for gradient-based optimizers, drops low-correlation features, and uses standard logistic regression loss. λ = 2 in Model 2 means **stronger regularization** which prevents overfitting.

**Model 3: Bayesian Logistic Regression using PyMC using a normal prior on weights and a Bernoulli likelihood**

Implementation details:

* Features used: Age, Work Experience, Monthly Income.
* Scaled the features using StandardScaler().
* Posterior sampling: Uses **MCMC (NUTS sampler)** via **pm.sample()** – samples from the true posterior distribution over weights and gives many possible sets of weights w that are consistent with the data
* Uses **multiple weight samples** from entire posterior distribution and average over all posterior samples to get **final prediction**.
* Final predictions are made by averaging and thresholding probabilities at 0.5

**Result:** accuracy = 55.058. To evaluate the **convergence** of the Bayesian Logistic Regression model using PyMC, we have compared MCMC sampling results with different configurations of samples and tune parameters:

Two MCMC configurations were tested using PyMCs pm.sample() to evaluate the **convergence**.

**Trace Plot Observations**:

A graph of a graph of a person

AI-generated content may be incorrect.

* **Config1:** (pm.sample(100, tune=100)) produced noisier, less stable chains with poor mixing, indicating insufficient exploration of the posterior. (Prediction results: Class 0 = 477, Class 1 = 818)
* **Config2** (pm.sample(1000, tune=1000)) produced smoother curves, more stable estimates and well-mixed chains, indicating better convergence. (Prediction results: Class 0 = 531, Class 1 =764)

A close-up of a graph

AI-generated content may be incorrect.

* The change in the prediction distribution shows a more robust posterior estimation leads to better-calibrated predictions. The model with 100 samples overpredicted class 1, while model with 1000 samples achieved a better balance. This reflects **posterior uncertainty** over parameters translates into **predictive uncertainty**. Hence, more samples in Bayesian inference improve prediction accuracy and reduce bias, with trace plots validating sampling effectiveness.

**Model 4:** Same model as above with variation in feature selection. Used features are - Age, WorkExperience, MonthlyIncome, CurrentProfile, OverTime, MaritalStatus\_Single.

**Results:** accuracy= 66.718. Since we have included more features as per correlation with target, the model is able to capture more complex patterns, so we can see a better result. However, trace plots show more noise hence slower convergence due to added complexity.

**Trace Plot Observations**:

* **Config1:** pm.sample(100, tune=100) produced noisier, less stable chains with poor mixing, indicating insufficient exploration of the posterior. (Prediction results: Class 0 = 673, Class 1 = 622)

A graph of a person's body

AI-generated content may be incorrect.

A graph of a function

AI-generated content may be incorrect.

* **Config2:** pm**.**sample(1000, tune**=**1000) produced smoother curves, more stable estimates and well-mixed chains, indicating better convergence. (Prediction results: Class 0 = 674, Class 1 = 621)
* The prediction distribution shows model with more features appears to **generalize better** and is **less sensitive** to sampling variation i.e. additional features are helping to stabilize decision boundaries. Model’s robustness aligns with a **tighter posterior distribution**, indicating higher confidence in its learned parameters.

**Sensitivity Analysis:** With varying prior obtained accuracies are:   
Accuracy with prior N(0, 1²): 0.6486, with prior N(0, 5²): 0.6479, with prior N(0, 10²): 0.6486

As we can see **Model is not very sensitive** to the choice of prior variance — all accuracies are **very close**. This suggests the features are likely not suffering from severe multicollinearity or overfitting that stronger regularization would help with.

|  |
| --- |
|  |

**(B) Regression Task:**

**Model: Bayesian Linear Regression (using conjugate priors)**-Goal is to predict monthly income based on the features of the dataset using probabilistic methods.

Implementation details:

* **Features used**: All except EmployeeID, MonthlyIncome and Attrition.
* **Target variable**: MonthlyIncome (continuous variable).
* **Train and Test data size:** Data is divided in 80:20 ratio with a random state=43
* **Feature engineering:** Target and Numerical feature variables are scaled using MinMaxScaler() to the [0,1] range. Categorical feature has already been one-hot encoded. Intercept term is added to feature matrix allowing model to learn an intercept/bias term during training.
* We have used a Gaussian prior over the weights and assume Gaussian noise in the observations, leading to a conjugate prior setting. This choice allows for a closed-form posterior distribution, enabling efficient computation of predictions and uncertainty estimates. It avoids costly sampling methods and provides interpretable model diagnostics. α controls regularization; low α → loose prior, high α → more regularization. β adjusts noise level; higher β = lower observation noise.
* **Evaluation Metrics: Log Likelihood** on test set, **BIC** and **AIC** for model complexity and fit trade-off, **RMSE** for standard regression error metric.
* **Hyperparameter Tuning: Grid search** on alpha and beta, **5-fold cross-validation** based on average log marginal likelihood (log\_evidence), BIC, and AIC.

**Convergence & Sensitivity:**

**Convergence: Closed-form posterior means** no iterative optimization**,** guaranteed convergence(numerically stable as long as matrix inversion is stable).

**Sensitivity:** Results can be sensitive to alpha and beta.That's why a grid search is used to find optimal prior α and noise precision β. Multiple folds average out overfitting.

**Results:** Best hyperparameters obtained using k-fold cross-validation, α =10.0, β =100.0 and the model has been trained using these hyper parameters. Results Obtained on Test set: Log Likelihood: 391.73, BIC: -339.11, AIC: -655.47, RMSE: 0.1418. These scores indicate a well-fitting model with low residual error and strong model evidence i.e. model has a good trade-off between fit and complexity.

**Insights**: Use of conjugate priors allows for efficient computation of the posterior and predictive distributions. Cross-validation ensures that the selected hyperparameters generalize well to unseen data.

The graph demonstrates that the model not only predicts values but also quantifies how sure it is about them.

In regions with less data or more variability, the gray shaded region widens, showing the model is less confident in its predictions. Where data is denser or more consistent, the band is narrower, indicating higher certainty.

A few **extreme outliers** exist (e.g., a person in their 30s with 90,000+ income or one near 0) which are not captured by the model and lie **outside the ±2σ band**

A graph with blue and orange lines

AI-generated content may be incorrect.

**(C) Clustering Task: Bayesian Gaussian Mixture Model (BGMM) –** Goal isto **identify and understand employee risk profiles** using unsupervised learning (clustering) and then validate how well those clusters can be predicted using a supervised model.

Implementation details:

* We are **clustering employees** based on work-related and behavioural features (e.g., job satisfaction, salary hike, etc.) then labeling those clusters as **High**, **Medium**, or **Low Risk**. training a classifier to **predict the risk label**, which can be useful for future employees without needing to re-cluster everything.
* **StandardScaler**: Normalize features before clustering.
* Gaussian Mixture Models (GMM) uses **Bayesian inference** to estimate the number of clusters, although we've manually set it to n\_components=3.
* We have used PCA to reduce the dimension of **feature** **space into 2D** to help you **visualize clusters** on a scatter plot. It computes two principal components (PC1, PC2). These are **linear combinations of original features** that capture the most variance.
* Obtained the component loadings to interpret which features influence PC1/PC2 most.

A screenshot of a computer screen

AI-generated content may be incorrect.

* weight\_concentration\_prior=0.01: This is a **Dirichlet Process parameter** that controls how much the model prefers balanced clusters. Lower values (like 0.01) **encourage the use of more components** with similar weight — helpful when clusters are not very distinct.

A screen shot of a graph

AI-generated content may be incorrect.

* Silhouette Score measures **how well clusters are separated**, we obtained Silhouette Score: 0.1099 indicating overlapping or unclear clusters.
* Random Forest Classifier is trained after clustering to classify future employee data into risk categories.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Conclusion:**

* Feature selection based on correlation also helped us here.
* Data Cleaning and standardization also helps to improve the model.
* Hyper Parameter tuning is important to improve the model.
* Exploring advanced techniques like ensemble models or deep learning could enhance predictive performance but not always jump directly on them.