GSJ: Volume 9, Issue 9, September 2021, Online: ISSN 2320-9186
www.globalscientificjournal.com

PREDICTING EMPLOYEE ATTRITION USING DECISION TREE ALGORITHM

Richard Eduvie, Johnwendy C. Nwaukwa, Felix Uloko, Eric Taniform

VERITAS UNIVERSITY, ABUJA

Abstract

Decision-making in an Organization is a necessity for the Human resource team in terms of predicting employee attrition. There are many complex, interrelated variables that impact the likelihood of employees quitting, which makes it extremely difficult—if not impossible—to manually predict which employees will quit, when they'll quit, and why they'll quit, especially at scale. This is where machine learning comes in, which finds statistical patterns in vast amounts of historical data, to uncover what humans would miss. In this case, machine learning can be used to find patterns in employee attrition, and create a model for predicting attrition. While much research focused on analyzing the IBM Dataset analytics based on the attrition feature as the subject given other features for prediction, this research seeks to contribute to knowledge by building a scalable decision tree model for employee attrition that can be used on any application which domain concern is for the employee for new prediction. Research science design method was carried out in this research. Secondary data source collection was used where we collected the IBM dataset for this research from Kaggle. The decision tree model was built using R and Transact SQL code in Microsoft SQL Server 2019 and then saved as a SQL procedure.ASP.NET which comprises HTML, CSS, and JavaScript was used to show how the model works and how it can be used to predict new employee data. C# was used to call the SQL procedure stored on the Microsoft SQL Server for prediction. Microsoft SQL Server 2019 was built with a lot of advanced features for big data analysis. Microsoft SQL Server 2019 big data cluster enables intelligence overall data and helps remove data silos by combining both structured and unstructured data across the entire data estate. Big Data Clusters integrates Microsoft SQL Server and the best of big data open-source solutions.

Keywords: IBM, HTML, CSS, Attrition, Decision tree

Introduction

The growing interest in machine learning among business leaders and decision-makers demands that researchers explore its use within business organizations (Darrell and Allen, 2018). One of the major issues facing business leaders within companies is the loss of talented employees (Alduayj and Rajpoot, 2018).

The key to success in an organization is the ability to attract and retain top talents (Rabbi, Ahad, Kousar, and Ali, 2015). The Human Resource (HR) department needs to identify the factors that keep employees and those which prompt them to leave (Frye, Boomhower, Smith, Vitovsky, and Fabricant, 2018). Organizations could do more to prevent the loss of good people.

Employee attrition is the most important factor that causes a huge loss to the company (Chowdhury and Mamun, 2017). There are many reasons for which the employees leave the company, such as; salary, dissatisfaction, stagnant career growth, etc. The Loss is not only in terms of the money but also the company sometimes losses the skilled employees who are the most valuable assets to the company (Will, 2021). If the company can predict the employees which are going to leave the company, in near future, they can also work on retention beforehand and avoid the loss of a valuable employee. The prediction of attrition is part of HR Analytics (Momanyi and Kaimenyi, 2015).

Companies lose trillion dollars a year from employee turnover (Shane and Ben, 2021). These losses can be devastating, especially amidst a time when many businesses are barely hanging on. Besides the cost of hiring and training new employees. The loss of an employee also leads to the breakdown of team morale, which can create a vicious downward spiral, with more employees following suit.

Many organizations assume this is the cost of doing business. Now and then, employees quit. The truth is, there are always ways to reduce employee attrition, saving costs, retaining valuable talent, and keeping up morale (Chiedu, Long, and Ashar, 2017).

There are many complexes, interrelated variables that impact the likelihood of employees quitting, which makes it extremely difficult—if not impossible—to manually predict which employees will quit, when they'll quit, and why they'll quit, especially at scale (Al Mamun and MD, 2017). This is where machine learning comes in, which finds statistical patterns in vast amounts of historical data, to uncover what humans would miss. In this case, machine learning can be used to find patterns in employee attrition (Fallucchi, Coladangelo, Giuliano, and De Luca, 2020).

1.2. Statement of Problem

It is critical to managing staff attrition to achieve low and healthy turnover rates to preserve organizational performance and, as a result, a competitive advantage (Shaw, 2010). Data-driven decisions and machine learning have been introduced as a result of the pressure on HR departments to give value to the firm (Tomassen, 2016). High employee attrition rates are now seen as a problem for businesses, putting more pressure on HR teams to keep attrition rates at a reasonable level (Park and Shaw, 2013).

1.4. Aim and objectives of the study

This research aims to design a prototype system for the HR department to be able to easily view employee attrition chart report when required and the objectives are:

1. Build the decision tree model using R and Transact SQL in Microsoft SQL Server 2019
2. Design the user interface and build the decision tree library using the ASP.NET and C# Dynamic Link Library.

2.1. Existing works of literature

Many researchers have proved the usefulness of human resource management (HRM) in working scenarios, production and management, and in identifying relationships with productivity. The results confirm that the impact of HRM on productivity has positive effects on a business's capital growth and intensity.

Shenghuan, Pradeep, and Timothy (2015), analyzed the dataset IBM Employee Attrition to find the main reasons why employees choose to re-sign. Firstly, they utilized the correlation matrix to see some features that were not significantly correlated with other attributes and removed them from our dataset. Secondly, they selected important features by exploiting Random Forest, finding monthly income, age, and the number of companies working significantly impacted employee attrition. Next, they also classified people into two clusters by using K-means Clustering. Finally, they performed binary logistic regression quantitative analysis: the attrition of people who traveled frequently was 2.4 times higher than that of people who rarely traveled.

Aseel, Asmaa, Munirah, Ruyan, and Hanan (2021), showed how several machine learning models are developed to automatically and accurately predict employee attrition. IBM attrition dataset is used in this work to train and evaluate machine learning models; namely Decision Tree, Random Forest Regressor, Logistic Regressor, Ada boost Model, and Gradient Boosting Classifier models. The ultimate goal is to accurately detect attrition to help any company to improve different retention strategies on crucial employees and boost those employee satisfaction.

Saeed, Naser, Ali, and Sarfaraz (2021), presented a three-stage (pre-processing, processing, post-processing) framework for attrition prediction. An IBM HR dataset is chosen as the case study. Since there are several features in the dataset, the "max-out" feature selection method is proposed for dimension reduction in the pre-processing stage. This method is implemented for the IBM Dataset. The coefficient of each feature in the logistic regression model shows the importance of the feature in attrition prediction. The results show improvement in the F1-score performance measure due to the "max-out" feature selection method. Finally, the validity of parameters is checked by training the model for multiple bootstrap datasets. Then, the average and standard deviation of parameters are analyzed to check the confidence value of the model's parameters and their stability. The small standard deviation of parameters indicates that the model is stable and is more likely to generalize well.

Shravan, Soham, Yash, Bhavesh, and Rahul (2017), proposed a decision support system that gives a deep insight into employee's behavior and whether he is willing to work in the organization or not. The attrition of the employees given in the open-source IBM dataset is predicted using three machine learning algorithms viz. Logistic Regression, Random Forest, and Artificial Neural Networks.

Kashyap and Kriti (2018), used the IBM employee dataset and found out that the characteristics of employees like Job Role, overtime, job level affect the attrition largely. They implemented various classification algorithms like logistic regression, LDA, ridge classification, lasso classification, decision trees, and random forests to predict the probability of attrition of any new employee and simultaneously tested them. Using various model evaluation metrics, they made a comparative analysis of the models and found out that LDA gave the highest accuracy, logistic gave the highest precision and ridge gave the highest recall.

2.2. Literature Findings

Machine learning can give important support to HRM applications. To the best of our knowledge, while much research focused on analyzing the IBM Dataset analytics based on the attrition feature as the subject given other features for prediction, this research seeks to contribute to knowledge by building a scalable decision tree model for employee attrition that can be used on any application which domain concern is for the employee for new prediction.

3.1. Research Methodology

The research methodology adopted in this study is the Design Science Research (DSR). DSR is a problem-solving paradigm that seeks to enhance human knowledge via the creation of innovative artifacts. Simply stated, DSR seeks to enhance technology and science knowledge bases via the creation of innovative artifacts that solve problems and improve the environment in which they are instantiated.

3.2. Decision Trees

Decision trees are graphical representations of choices that can be made by a business, which enable the decision-maker to identify the most suitable option in a particular circumstance. Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values. The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner. A greedy strategy is usually used because they are efficient and easy to implement, but they usually lead to sub-optimal models. A bottom-up approach could also be used. The algorithm is summarized as follows:

1. *create a node N;*
2. *if samples are all of the same class, C then*
3. *return N as a leaf node labeled with the class C;*
4. *if the attribute-list is empty then*
5. *return N as a leaf node labeled with the most common*

class in samples;

6. *select test-attribute, the attribute among attribute-list with the highest information gain;*
7. *label node N with test-attribute;*
8. *for each known value ai of test-attribute*
9. *grow a branch from node N for the condition test-attribute= ai;*
10. *let si be the set of samples for which test-attribute= ai;*
11. *if si is empty then*
12. *attach a leaf labeled with the most common class in samples;*
13. *else attach the node returned by Generate_decision_tree (si, attribute-list_test-attribute)*

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. In data mining, trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization, and generalization of a given set of data. Data comes in records of the form: $(x, Y) = (x_1, x_2, x_3... x_k, Y)$............. (1)

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector **x** is composed of the input variables, x_1, x_2, x_3, etc., that are used for that task.

Decision trees used in data mining are of two main types:

- Classification tree analysis is when the predicted outcome is the class to which the data belongs.
- Regression tree analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient’s length of stay in a hospital).

The term Classification and Regression Tree (CART) analysis is an umbrella term used to refer to both of the above procedures, first introduced by Breiman *et al*. Trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split. There are many specific decision-tree algorithms. Notable ones include:

1. ID3 algorithm
2. C4.5 algorithm
3.C5.0 algorithm

4. Chi-squared Automatic Interaction Detector (CHAID). Performs multi-level splits when computing classification trees

The most common types of decision tree algorithms are CHAID, CART, and C4.5. CHAID (Chi-square automatic interaction detection) and CART (Classification and Regression Trees) were developed by statisticians. CHAID can produce a tree with multiple sub-nodes for each split. CART requires less data preparation than CHAID but produces only two-way splits. C4.5 comes from the world of Machine Learning and is based on information theory. The most well-known algorithm in the literature for building decision trees is the C4.5 (Quinlan, 1993). C4.5 is an extension of Quinlan's earlier ID3 algorithm (Quinlan, 1979). One of the latest studies that compare decision trees and other learning algorithms has been done by (Tjen-Sien Lim et al. 2000). The study shows that C4.5 has a very good combination of error rate and speed.

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy.

At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sub lists. This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- The instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

3.2. Building the Predictive Model

Classification techniques were used to develop the prediction models used in the study. Classification is a classic data mining technique based on machine learning. Basically, classification is used to classify each item in a set of data into one a predefined set of classes or groups. The classification method makes use of mathematical techniques such as decision trees,

3.3. System Architecture

During the system analysis, the analysis of system data is very important (Dhiraj, 2021). Analysis of data is made up of more than one level at the beginning (first level) and different ideas are used at each level. At the first level, we develop the architecture of the system.

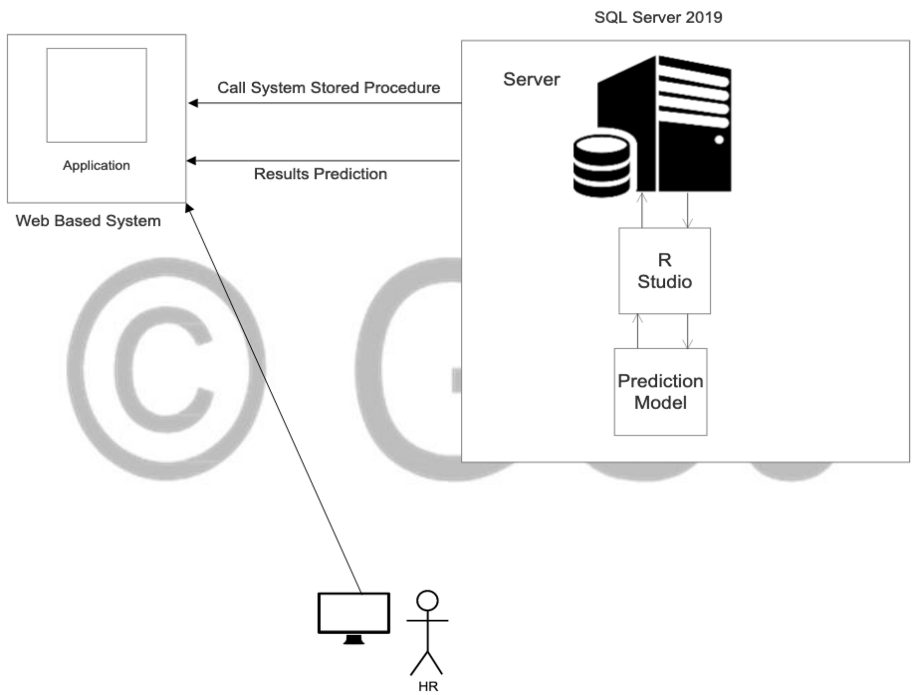

Figure 3.1: Architecture Design of the system

The IBM employee dataset for this research work was downloaded from Kaggle, 2017. The dataset was populated with the aid of the storage service to the database. The records from the database table are trained using the Decision Tree, due to the fact that the problem is actually a classification problem. In this research, we will use the algorithm to make our prediction. The model is trained using the training procedure call and then tested using the testing dataset. After that, the records are refined and the predicted model is then moved to another storage table for prediction. A web service written in C# calls the predicted model that is deployed for use to the system where it is used for predicting employee attrition.

3.4. Source of Data collection

Secondary sources were used to gather information for this research project. The data collection instrument issued for this study is the IBM employee dataset downloaded from Kaggle, 2017.

3.4.1. Data Exploration and Visualization

Dataset Structure: 1470 observations (rows), 35 features (variables)

Missing Data: There is no missing data in our dataset.

Data Type: We only have two datatypes in this dataset: factors and integers

Label: Attrition is the label in our dataset, and we would like to find out why employees are leaving the organization.

Table 1: Attributes of Employee Attrition

S/No	Attributes
1	Attrition
2	Age
3	DailyRate
4	DistanceFromHome
5	Education
6	EnvironmentSatisfaction
7	HourlyRate
8	JobInvolvement
9	JobLevel
10	JobSatisfaction
11	MonthlyIncome
12	MonthlyRate
13	NumCompaniesWorked
14	PercentSalaryHike
15	PerformanceRating
16	RelationshipSatisfaction
17	StandandHours
18	StockOptionLevel
19	TotalWorkingYears
20	TrainingTimesLastYear
21	WorkLifeBalance
22	YearsAtCompany
23	YearsInCurrentRole
24	YearsSinceLastPromotion
25	YearsWithCurrManager

3.5. Software Development Methodology

The software methodology adopted in this research is the iterative development model of the System Development Life Cycle (SDLC) method. The requirements are not finished in the iterative model, and the iterative process begins with a minimal set of requirements. Each iteration creates a little version of the product, which is then repeated until the final version is ready. The implementation of an iterative process model begins with a subset of required specifications.

The graphical representation of the iterative model is given below

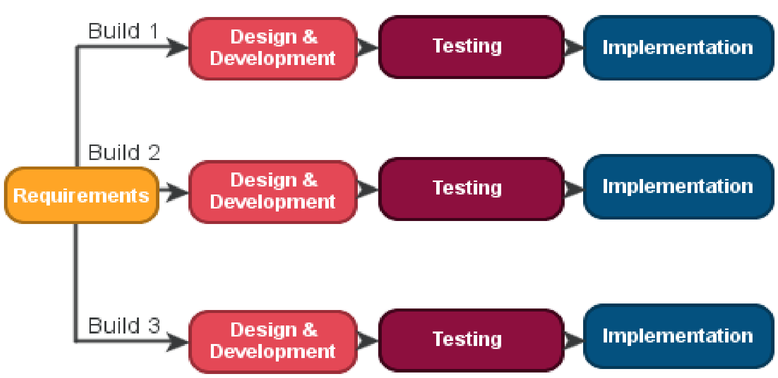

Figure 3.2: Iterative Model

3.6. Requirement Analysis

Requirement's analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications and is an important aspect of project management.

This is the first phase of software development in Iterative model, where all possible requirements of the system to be developed is captured and documented in a SRS (System Requirement Specification) document.

3.6.1. Functional and non-functional requirements

The system's functional requirements are in one phase. The HR logs into the system, upload employee data and then make prediction.

The Non-functional requirements include the following:

1. Usability: The proposed system should be easy for the user to operate, enter data, and interpret the output
2. Compatibility: the proposed system should be compatible with all web browsers.

4.1. Software design Modelling

Design modeling in software engineering represents the features of the software that helps engineer to develop it effectively, the architecture, the user interface, and the component level detail (Swati, 2021). The Unified Modelling Language used to model this system are the use case, activity, sequence, and the class diagram. It is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems (Visual, 2020).

4.1.1. Use-case diagram of the system

A use-case model describes a system's functional requirements in terms of use cases. It is a model of the system's intended functionality (use cases) and its environment (actors). Use cases enable you to relate what you need from a system to how the system delivers on those needs (Visual, 2020). The system comprises one actor, which is the Human resource manager.

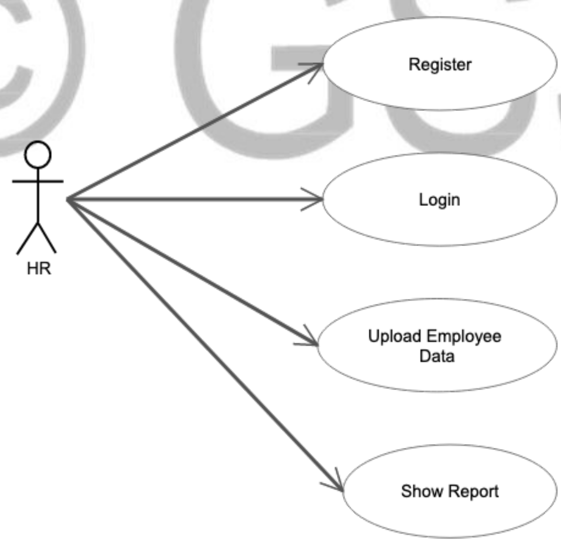

Figure 4.1: System Use Case

4.1.2. Activity Diagram of the system

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency (Visual, 2020). It describes the flow of control of the target system, such as exploring complex business rules and operations, describing the use case also the business process (Visual, 2020).

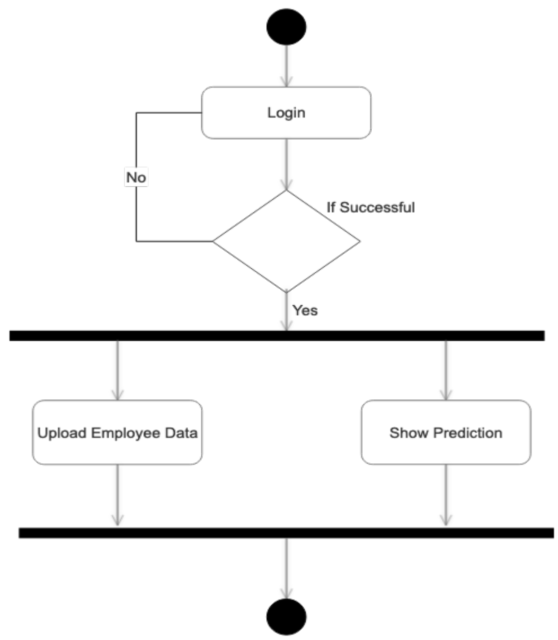

Figure 3.2: Activity Diagram of the system

4.1.3. Sequence Diagram of the system

The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case. With the advanced visual modeling capability (Visual, 2020).

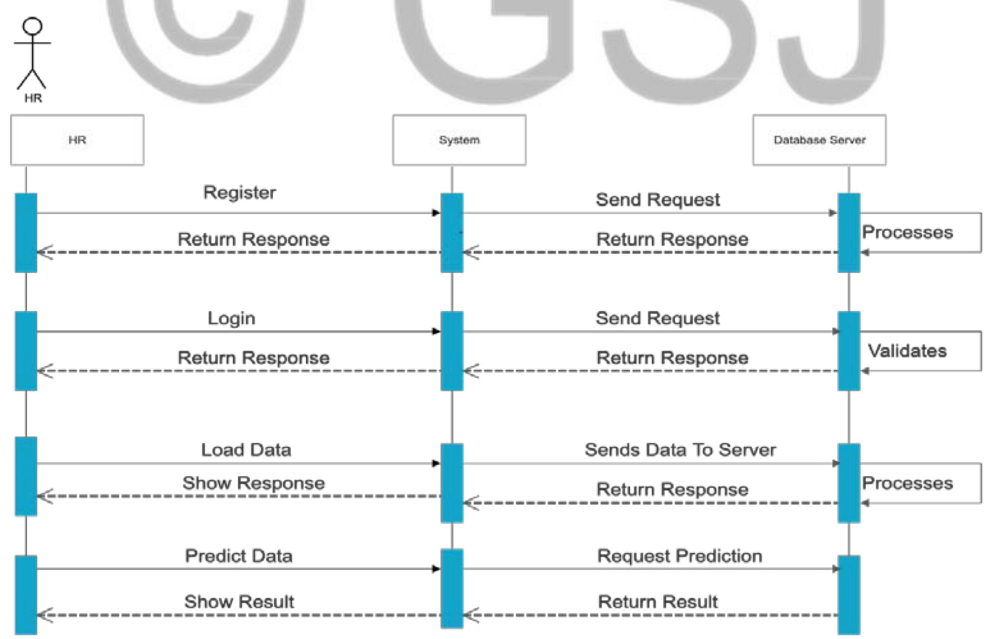

Figure 4.3: Sequence Diagram

4.1.4. Class Diagram of the system

The class diagram is a central modeling technique that runs through nearly all object-oriented methods. This diagram describes the types of objects in the system and various kinds of static relationships which exist between them.

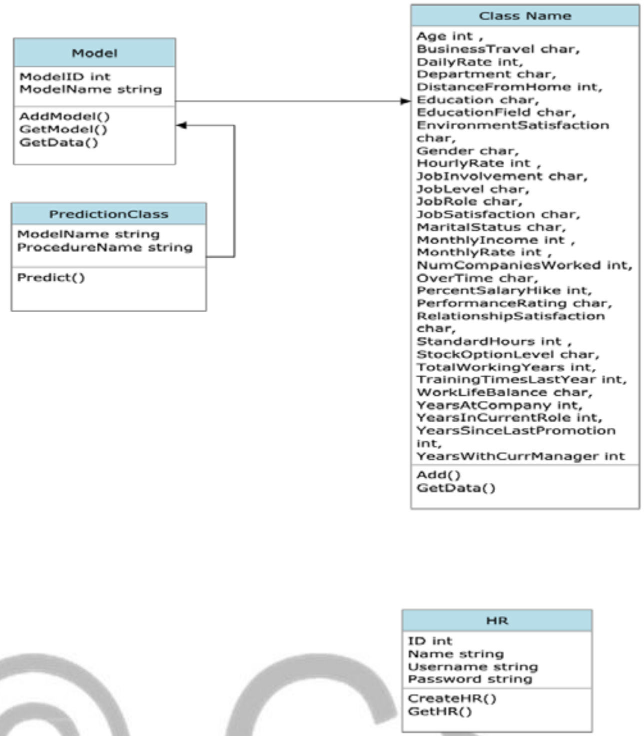

Figure 4.4: Class Diagram of the System

4.1.5. Database Schema and Design

During the research, the IBM employee dataset was collected from Kaggle, 2017 to gain knowledge towards the designing of the system. The dataset collected online was refined manually and some of the main attributes were selected for this research based on our scope of the study. We created our database schema and design using Microsoft SQL Server 2019 based on the attributes for our prediction and populate the database table with the refined dataset.

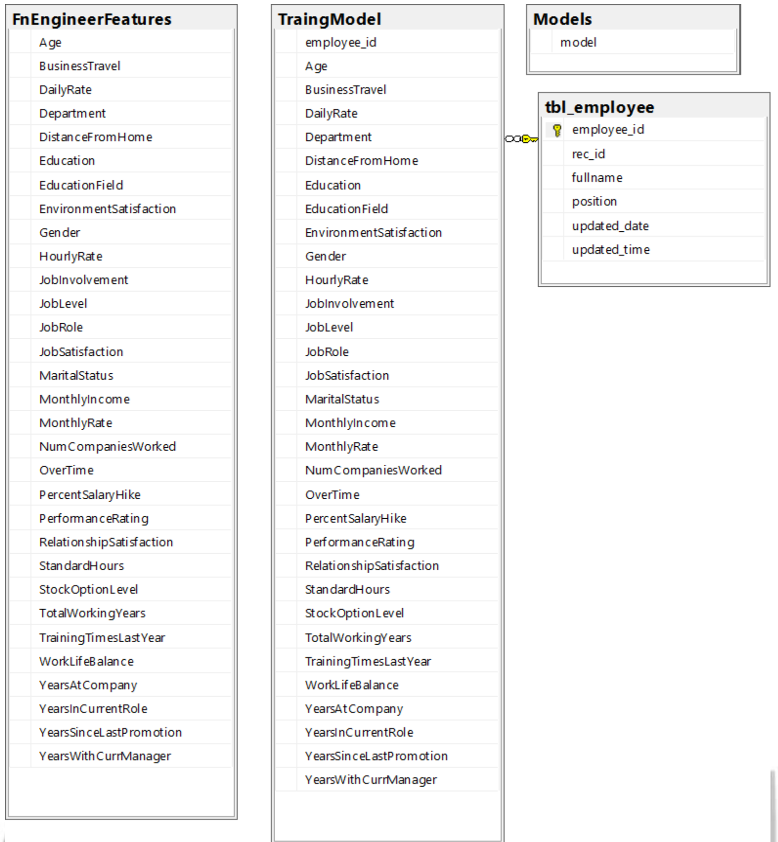

Figure 4.5: Database Schema and Design

The database tables are created and normalized for the efficient production of this system. The FnEngineering table where the dataset is populated is not normalized, this is as a result of having a huge dataset and extracting the relevant attributes for the system. The FnEngineering table is used to aid us in the model prediction. The employee table stores employee data and the training table makes predictions of the new employee records.

4.2. Hardware Requirements

These refer to the computer's physical features required to implement the system. Features are as follows: at least 250GB HDD, 4GB RAM, and at least Intel Pentium Dual-Core.

4.3. Software Requirements

These are the computer programs and procedures required to implement the system Table 1 indicates the minimum software requirements.

Table 2: Software Requirements

Requirements	**Software**
Operating System	Microsoft Windows (from window 8 and above) or Microsoft Server
DBMS	Microsoft SQL Server 2019
Programming Languages	R, ASP.NET, C#
Development Tool	Visual studio 2019

4.4. The Web Application

The web application was built using ASP.Net which comprises of HTML, CSS, and JavaScript and C #. HTML, CSS, and JavaScript was used to design the user interface of the system. C# was used to call the SQL server procedure from the database storage and represent the data in a graphical form.

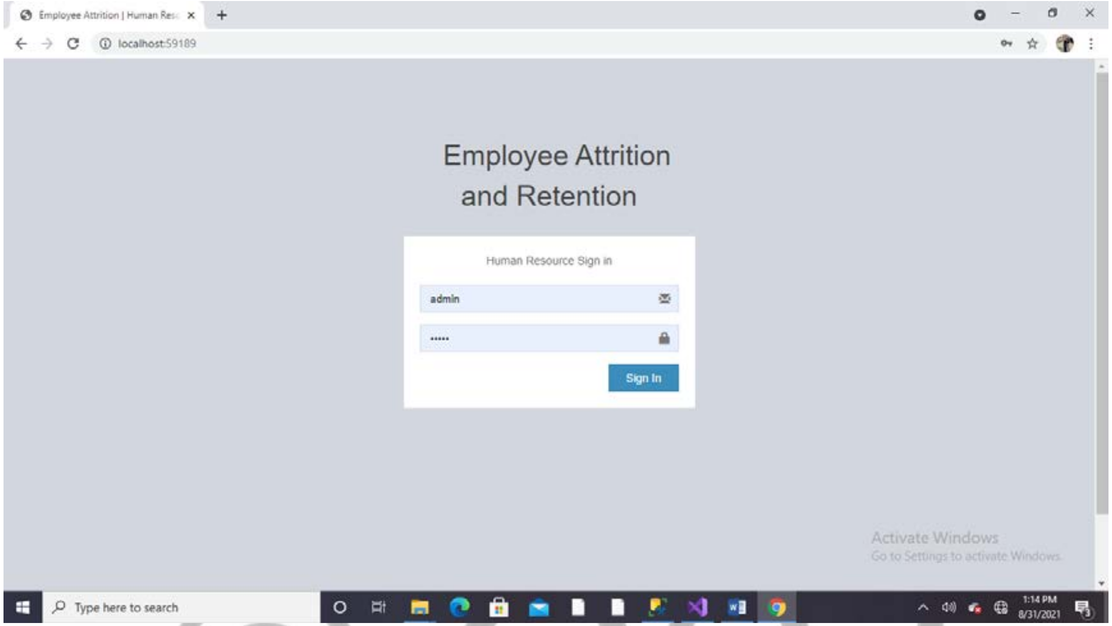

Figure 4.6: Admin Login Interface

Figure 4.6 shows login interface for the administrator. The admin are required to login with their registered username and given password for the system.

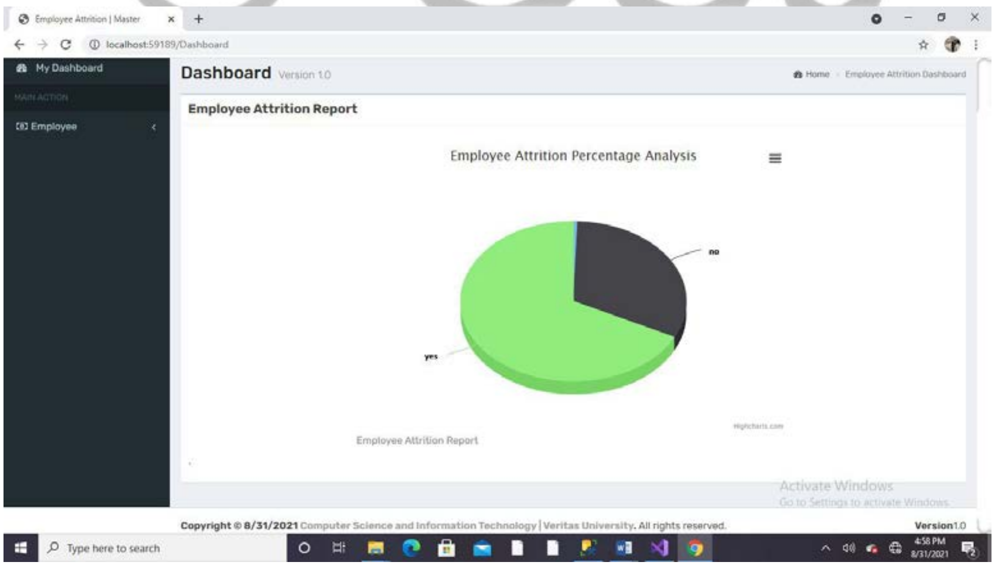

Figure 4.7: Graph Representation of Training Dataset

Figure 4.7 shows the graphical representation of the decision tree training datase

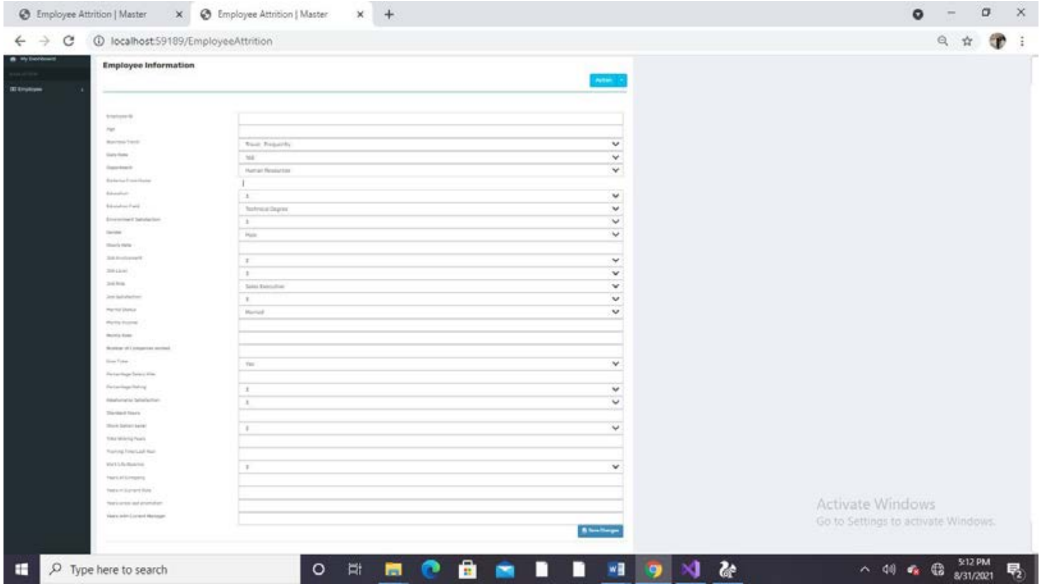

Figure 4.8: Form Template to Add New Employee Records

Figure 4.8 shows the form template to add new employee records to predict employee attrition

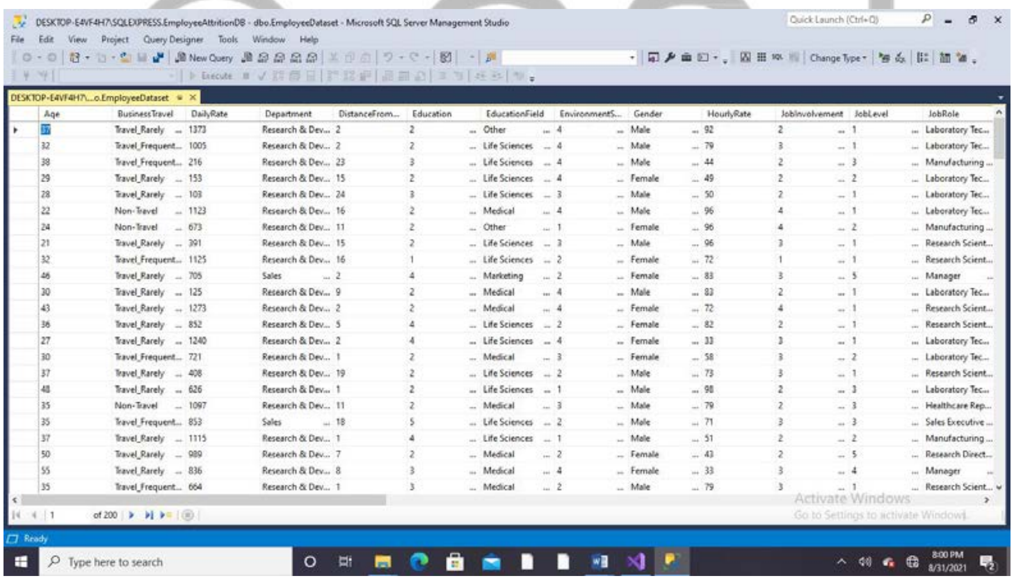

Figure 4.9: Dataset Report

Figure 4.9 shows the IBM employee dataset populated to the Microsoft SQL Server

5.1 Findings and Discussions

During this study, we reviewed different works of works of literature done on employee attrition. Various machine learning techniques have been used for this research. Most of the works done on this research involve complete data analysis for the employee datasets. The scope of our data source collection is the IBM employee dataset.

5.2 Contribution and Recommendations

To the best of our knowledge, while much research focused on analyzing the IBM Dataset analytics based on the attrition feature as the subject given other features for prediction, this research seeks to contribute to knowledge by building a scalable decision tree model for employee attrition that can be used on any application which domain concern is for the employee for new prediction. The Microsoft SQL Server 2019 relational database was chosen for this research because of the scalable features it has. Microsoft SQL Server 2019 big data cluster enables intelligence overall data and helps remove data silos by combining both structured and unstructured data across the entire data estate. Big Data Clusters integrates Microsoft SQL Server and the best of big data open-source solutions. It is deployed on scalable clusters using Apache Spark, HDFS containers with Kubernetes. This system can be used in any organization and the attributes for prediction may be further expanded or modified based on how different HR team accesses their employees.

We recommend researchers whose area is on data science to try and use Microsoft SQL Server for building any kind of machine learning model for prediction.

5.3 Conclusion

In conclusion, we successfully built our decision tree model using R and Transact SQL code and then stored the model in the SQL procedure. We then called the SQL procedure using C# to our web application to test it on new employee data. ASP.NET which comprises HTML, CSS, and JavaScript.

5.4 Area for Further Research

For further research, we will look at evaluating deep artificial neural networks, deep reinforcement algorithms for very large employee datasets using the scalable Azure Cosmos DB built by Microsoft.

References

Al Mamun, A. C., and MD, N. H. (2017). Factors affecting employee turnover and sound retention strategies in business organization: a conceptual view. *Problems and Perspectives in Management*, *15*(1), 63-71.

Albrecht, S. L., Bakker, A. B., Gruman, J. A., Macey, W. H., and Saks , A. M. (2015). Employee engagement, human resource management practices and competitive advantage: An integrated approach. *Journal of Organizational Effectiveness: People and Performance*, *2*(1), 7-35.

Alduayj, S. S., and Rajpoot, K. (2018). Predicting Employee Attrition using Machine Learning. *2018 International Conference on Innovations in Information Technology (IIT)* (pp. 93-98). Al Ain, United Arab Emirates: IEEE.

Aseel , Q., Asmaa , A.-M., Munirah , A.-H., Ruyan , A., and Hanan, A. (2021). Prediction of Employee Attrition Using Machine Learning and Ensemble Methods. *International Journal of Machine learning and computing, 11*(2), 110-114.

Basariya, S. R., and Ahmed, R. R. (2019). A Study On Attrition – Turnover Intentions Of Employees. *International Journal of Civil Engineering and Technology ER, 10*(01), 2594-2601.

Brunton, S. L., Kutz, N. J., Manohar, K., Aravkin, A. Y., and Morgansen, K. (2021). Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning. *AIAA Journal*, 1-28.

Chiedu, K. C., Long, C. S., and Ashar, H. B. (2017). The Relationship Among Job Satisfaction, Oreganizational Commitiment and Employees' Turnover at Unilever Corporation in Nigeria. *European Journal of Multidisciplinary Studies*, 370-383.

Chowdhury, A., and Mamun, N. H. (2017). Factors affecting employee turnover and sound retention strategies in businessorganization: a conceptual view. *Problems and Perspectives in Management*, *15*(5), 63-69.

Darrell, M. W., and Allen , R. J. (2018). *How artificial intelligence is transforming the world.* Washington: Brookings.

Fallucchi, F., Coladangelo , M., Giuliano, R., and De Luca , E. W. (2020). Predicting Employee Attrition Using Machine Learning Techniques. *Computers, 9*, 86-92.

Francesca , F., Marco , C., Romeo , G., and Ernesto , W. D. (2020). Predicting Employee Attrition Using MachineLearning Techniques. *Computers*, 1-17.

Frye, A., Boomhower, C., Smith, M., Vitovsky, L., and Fabricant, S. (2018). Employee Attrition: What Makes an Employee Quit? *SMU Data Science Review*, 1-29.

Ibidunni, S., Osibanjo, O., Adeniji, A., Salau, O. P., and Falola, H. (2016). Talent Retention and Organizational Performance: A Competitive Positioning in Nigerian Banking Sector. *Periodica Polytechnica Social and Management Sciences*, 1-13.

Ingersoll, M. R., May, H., and Collins, G. (2019). Recruitment, Employment, Retention and the Minority Teacher Shortage. *Education Policy Analysis Archives, 27*(37).

Jain, D. (2017). *Evaluation of Employee Attrition by Effective Feature Selection using Hybrid Model of Ensemble Methods.* National College of Ireland.

Karnouskos, S. (2017). Massive open online courses (MOOCs) as an enabler for competent employees and innovation in the industry. *Computers in Industry, 91*, 1-10.

Kashyap, B., and Kriti, S. (2018). Comparative Study of the Machine Learning Techniques for Predicting the Employee Attrition. *International journal of research and analytical review, 5*(3), 568-577.

Momanyi, N. B., and Kaimenyi, C. K. (2015). An Investigation into Factors Causing High Nurse Turnover in Mission Hospitals in Kenya: A Case for Picea Chogoria Hospital. *International Journal of Academic Research in Business and Social Sciences*, 55-86.

Nausheen, S., and But, S. (2015). Factors Influencing Employee Retention: An Integrated Perspective. *Journal of Resources Development and Management, 6*, 32-49.

Rabbi, F., Ahad, N., Kousar, T., and Ali, T. (2015). Talent management as a source of competitive advantage. *Journal of Asian Business Strategy*, 208-214.

Saeed, N.-Z., Naser , S.-G., Ali , A.-N., and Sarfaraz , H. Z. (2021). An Improved Machine Learning-Based Employees AttritionPrediction Framework with Emphasis on Feature Selection. *Journal of Mathematics*, 1-14.

Shane, M., and Ben, W. (2021). *This Fixable Problem Costs U.S. Businesses $1 Trillion*. Retrieved July 2021, from WorkPlace: https://www.gallup.com/workplace/247391/fixable-problem-costs-businesses-trillion.aspx#:

Shenghuan, Y., Pradeep, R., and Timothy, S. (2015). *IBM Employee Attrition Analysis.* Scientific.

Shravan, S., Soham, A., Yash, B., Bhavesh, K., and Rahul, P. (2017). Analysis of employee attrition and implementing a decision support system providing personalized feedback and observations. *Journal of critical review, 7*(19), 2372-2380.

Travis, B. (2021). *What Successful People (Who Are Happy) Do Differently*. Retrieved July 2021, from huffpost.com: https://www.huffpost.com/entry/what-successful-people-wh_b_14137794

Will, K. (2021, July). *Intrapreneurship*. Retrieved 2021, from Investopedia: https://www.investopedia.com/terms/i/intrapreneurship.asp

Yarkoni, T., and Westfall, J. (2017). Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. *PubMed, 12*(6), 1100-1122.