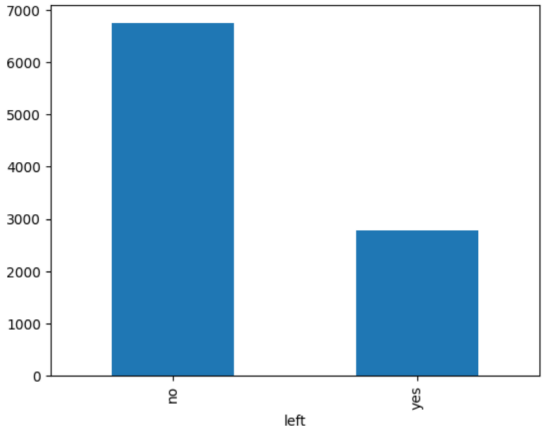
CIS 3207 Final Project: Predict Employee Turnover
Final Report

Amitai Goldmeer & John Currie

Employee retention and turnover are issues that companies globally are constantly attempting to understand and take control of. Knowing what causes an employee to stay or leave a company is valuable information, as it can be used in the future to entice employees to stay. Using that information to predict if an employee will leave a company is even more valuable, giving another chance to convince them to stay. For these reasons, among others, employee turnover is an issue that most, if not all major companies are always looking into.

We found an anonymous dataset from an unidentified company in the US with over 10,000 records of employees and various statistics, including their salary level, how long they worked for the company, their last satisfaction rating, and others that were part of an internal survey. This dataset also included whether, within the time period of 2016 - 2020, this employee had left the company or not. Our goal, as requested by the company who provided the dataset, was to find factors that were linked to employee turnover, so that the company could try to remedy these issues to lower future turnover rates.

This dataset used a binary classifier, with its ground-truth value being if an employee chose to stay, or chose to leave. Given this, our approach was to try multiple related models that would work for a supervised binary classification model, to see which one would have the best accuracy, as well as to see if they would give similar results for features that were most likely to cause an employee to stay or leave. To this end, we chose to use a logistic regression model, as it is a straightforward method for binary classification tasks. As we referenced in our proposal, a study done at UC Berkeley found it to be the most effective model they used in a study similar to this one. The next model we chose to use was a KNN model, as it has the potential to identify patterns through similarity. Lastly, we chose to use clustering in order to test its effectiveness for classification and to see if it could provide insights into how samples correspond with results. Data preprocessing and the KNN model were done by Amitai Goldmeer and the Logistic Regression model and Clustering were done by John Currie. Evaluation and drawn conclusions were done collectively.



As detailed previously, this data had about 10,000 rows of information, broken into nine feature rows and one row for the ground truth value. The majority of these features were already numerical, however the features ‘department’, which is the department that the specified employee worked in, ‘salary’ which was the salary range of the specified employee, separated into low, medium, and high, and ‘left’ which was the ground-truth value stating if an employee chose to leave the company or not, were all categorical. We used label and ordinal encoding to turn these categorical features into numerical ones. When these features were graphed, it was found that most employees had a salary categorized as ‘medium’ (out of low, medium, or high), that there was a spread of departments where these employees were from, with the largest being sales, and most importantly that about 70% of the sampled employees chose to stay with the company (Figure 1). Since the ground truth values were not balanced, we created a balanced version of the dataset by randomly selecting a subsection of the majority samples until the dataset was balanced. For the numerical features, there was little work needed, as no features in this dataset had null values, presumably because of how this data had been provided. All of these features were still graphed, to check for balance in the data.

For features that were not binary classifiers, the data was mostly balanced, with items such as reviews given by employees, satisfaction levels, tenure with the company, and average hours worked all showing a normal distribution. Binary features were less balanced, with about 90% of employees having been promoted within the time frame, and about 75% of employees having received a bonus within the time frame. Nearly all of the features did not show any type of correlation, with the exception of tenure (years at the company) and average hours worked per month, which had a correlation of 0.97 (Figure 2). While very high for 2 data points, given the context it is not very surprising that a correlation exists. Additionally, the initial data showed a slight correlation between the reviews an employee gave, with a correlation of 0.3 (Figure 2). Once again, this is not a surprising correlation to find. Overall, however, the initial data showed no significant correlation to itself. Our goal was to ideally find relationships between the data, and a way to predict values from this data, that initial findings could not find.

	promoted	review	salary	tenure	satisfaction	bonus	avg_hrs_month	department	left
promoted	1.000000	0.001879	0.002926	0.001410	-0.011704	0.001072	-0.002190	0.003617	-0.036777
review	0.001879	1.000000	-0.001292	-0.184133	-0.349778	-0.003627	-0.196096	0.004471	0.304294
salary	0.002926	-0.001292	1.000000	-0.007428	0.005547	-0.008002	-0.010633	-0.018294	0.009589
tenure	0.001410	-0.184133	-0.007428	1.000000	-0.146246	-0.000392	0.978618	0.005470	0.010521
satisfaction	-0.011704	-0.349778	0.005547	-0.146246	1.000000	0.000704	-0.143142	-0.009292	-0.009721
bonus	0.001072	-0.003627	-0.008002	-0.000392	0.000704	1.000000	-0.000370	0.000378	-0.011485
avg_hrs_month	-0.002190	-0.196096	-0.010633	0.978618	-0.143142	-0.000370	1.000000	0.001829	0.009008
department	0.003617	0.004471	-0.018294	0.005470	-0.009292	0.000378	0.001829	1.000000	0.000270
left	-0.036777	0.304294	0.009589	0.010521	-0.009721	-0.011485	0.009008	0.000270	1.000000

With the data preprocessed, the next step was to split and normalize the data for logistic regression. Using sklearn's train_test_split function, we split the data into 88% training and 12% testing. Following the data split, the next step was data normalization to ensure that all features contribute equally to model training. Using sklearn's StandardScaler function, we normalized the data. This normalization process ensures that each feature aligns with the same scale so that they can be weighted as equally as possible.

Moving forward with logistic regression modeling, our objective was to identify the optimal regularization coefficient for the model using the K-fold cross-validation approach. We used the KFold and LogisticRegression functions from sklearn to handle the cross-validation. We employed 5 folds during cross-validation and utilized the testing data to evaluate the performance of each fold, measuring accuracy as our primary metric for assessing the model's effectiveness under different regularization coefficients. Through this iterative process, we discovered that a regularization coefficient of 0.1 yielded the most favorable results.

Having determined the optimal regularization coefficient, the final step was to retrain the logistic regression model using the entire dataset. This retraining procedure was conducted not only on the original dataset but also on a balanced dataset, addressing any class imbalances that might affect the model's predictive performance. By training the model on both datasets, we ensured its robustness and ability to generalize well to new, unseen data instances.

```
unbalanced:
accuracy: 0.717, recall: 0.193, precision: 0.717, f1: 0.305
balanced
accuracy: 0.667, recall: 0.657, precision: 0.695, f1: 0.675
```

While the balanced dataset has a worse overall accuracy than the unbalanced dataset, the recall score is much higher, which means that the results should be more useful. We also wanted to gain some insights into the impact that each feature holds on the prediction of results.

Since the datasets were normalized, we took the weights of each feature and ranked them from highest to lowest.

```
Feature: review, Coefficient: 0.8302269157041582
Feature: satisfaction, Coefficient: 0.30957118884828194
Feature: avg_hrs_month, Coefficient: 0.13149715416968574
Feature: promoted, Coefficient: -0.0904785347562236
Feature: tenure, Coefficient: 0.08211665418556058
```

Review and satisfaction stand out with far larger weights, which would make sense given that those are employee evaluations, which would comprise reflections from various other features.

The second model that we chose to use was a KNN model. As we did for the logistic regression model, the sklearn train_test_split function was used to split the data into training and testing sets. This was followed by the normalization of the data using sklearns' StandardScaler function. This normalization process ensures that each feature aligns with the same scale so that they can be weighted as equally as possible.

From there, a hyperparameter was needed, in order to find a potential 'k' number of neighbors in the KNN model. Using the KNeighbors classifier, and the sklearn function GridSearchCV, options from 1 to 100 were sampled over the testing dataset, on the features and the ground-truth values. When scoring for accuracy to find the best parameter and k-value, a best score of 0.838 was found, with a k-value of 19.

Having found the appropriate data from the testing set to create a model, the sklearn KNeighborsClassifier was once again implemented, this time over the optimal k-value found previously. Implementing the new classifier over the training data, and then using it to create predictions based on the test data yielded overall good results. Accuracy, Recall, Precision, and F1 scores were all calculated manually so that the number of true positives, true negatives, false positives, and false negatives could all be observed as well. The model outputs a high accuracy of 81.6%, as well as a high precision of 80.5%. We were particularly interested in accuracy, as the amount of true positives and true negatives was very high. Out of the test set (1145 records), 728 true negatives were found, and 206 true positives were found. Not only does this encompass the vast majority of the testing set, with only 211 records incorrectly attributed, but it also matches the percentage breakdown seen in data preprocessing between employees who chose to stay with or leave the company. The other two scores, F1 and Recall, showed average scores of 66.1% and 56.1% respectively. While not ideal metrics of our data, these findings were representative of similar findings from our other models.

```
True Positive =  206

True Negative =  728

False Positive =  50

False Negative =  161

accuracy: 0.816, recall: 0.561, precision: 0.805, f1: 0.661
```

We wanted to experiment with the effectiveness of clustering for a classification task, even though clustering is not traditionally used in this manner. Using sklearn's PCA and KMeans functions, we were able to reduce the balanced dataset to two components and then make two clusters using Kmeans. To measure the results we take the ratio of each target in different clusters. There was about a 57% clustering of each feature within their respective clusters. While it is better than randomly selecting a cluster for each sample, it is not the optimal choice for this task.

Another approach for the clustering model was to apply clustering to each feature with each other feature. This way each feature would be isolated and compared to other isolated features. While the NMI scores were not particularly impressive, the ranking of each feature did align fairly well with the weights of the logistic regression model. The largest five from each model included the same set of features, and the review was by the largest value measured from each model. This observation shows that clustering can reveal insights into the weights of each feature.

Overall, these models produced satisfactory results for our data. Both the logistic regression model and the k nearest neighbor model are ones that are relatively resource-cheap, meaning that they do not take too much time or computational power to make or use. This also means that in larger scale testing, they could be scaled with more data to provide hopefully more accurate results that could be used in other circumstances, for other companies, or more.

While both models produced satisfactory results, the KNN model produced results with both a higher accuracy and precision than the logistic regression model. Although this defied our initial expectations, that result may not hold when looking at a scaled-up dataset. Improvements to both of these models could be achieved through several different means:

increasing the sample size, layering models, or fine-tuning features to be more representative. Additionally, while our clustering was by no means the most accurate way to look at this data, it did reveal some trends in the data that would be useful information for those running this type of test. It, along with our other models, showed that some of the most important features out of the ones given were review and average hours worked. These are not overly surprising, as the review given by an employee, and how much they chose to work on average, are logically representative signs of their commitment and feelings towards a given company.

If we were to continue this project, the next step would be to try using a deep neural network was this would almost certainly produce much better results. Furthermore, we may want to test models without 'review' and 'satisfaction', to make predictions with raw data, especially since it is less likely that companies would have these metrics on hand for every employee. However, given the data that was provided by the company initially, we believe that the correlations we have found, and the models that we created, satisfy their initial request of what to do with this data.

Acknowledgements:

Dataset:
https://www.kaggle.com/datasets/marikastewart/employee-turnover

Sklearn:
https://scikit-learn.org/stable/user_guide.html