STAT3106 Applied Machine Learning
Philip Tham (ht2546)

Project 1
Binary Classifier for employee departures

Problem specification
Atypical employee departures are costly for government agencies because, first, they need to restart a hiring process to replace the previous employee, and second, there is an additional cost of lost knowledge and experience as an experienced employee leaves the organization. Towards the aim of mitigating atypical employee departures, an applied machine learning solution would be to **create a binary classification model that predicts whether an employee will leave within the same fiscal year**. Based on the importance and directional impact of different features in the model, an agency might be able to take steps to retain employees. For instance, if base salary is found to be an important feature in the model that increases along with the probability that an employee stays, the agency could consider increasing base salary to retain the employee. The model can further be used to determine exactly how much of a base salary increase would be sufficient to retain the employee.

Overview of dataset
Data used in training the model was pulled from the New York City Citywide Payroll Data for fiscal years 2014 to 2024 via an API call. To keep the training dataset size manageable, only records for the **Administrator for Children's Services** employees that are paid **per Annum** and are currently either actively employed or no longer employed by the agency are considered. In other words, the model has only been trained on longer-term employees paid per year instead of per day or per week workers. In total, there were 82689 examples in the total dataset, which was rather imbalanced, with 84% being employed at the end of the fiscal year and the remaining 16% not being employed. Two thirds of the data was used for training while the last third was left as a test validation set. Stratified splitting was used to ensure that the proportion of employed and not employed examples remained similar.

Features
The following features were included in the original dataset:

Table 1. Features in original dataset

Feature name	**Description**
leave_status_as_of_june_30	Actively employed or ceased employment at the agency ceased (categorical data). This is the **target**, where 'ACTIVE' was converted into 1 and 'CEASED' was converted into 0, which was renamed 'active.'
fiscal_year	Fiscal year of the entry (June 30, fiscal year). After being used to generate time_in_pos_months (see below), this feature was removed from the training dataset.

agency_start_data	Date which employee began working for the current agency (discrete numerical data). After being used to generate time_in_pos_months (see below), this feature was removed from the training dataset.
work_location_borough	Borough of employee's primary work location (categorical data). This data was converted into dummy one-hot encoded variables.
title_description	Civil service title description of the employee (categorical data) After being used to generate means and standard deviations for zscore feature generation (see below), this feature was removed from the training dataset.
base_salary	Base salary assigned to employee. Exploratory data analysis (EDA) revealed that the base salary differs across individuals of the same title_description
regular_hours	Number of regular hours employee worked in the fiscal year. EDA revealed that the regular hours worked also differs across individuals of the same title_description
regular_gross_paid	Amount paid to the employee for base salary during the fiscal year. EDA revealed that regular_gross_paid is ***not*** simply base_salary * regular_hours.
ot_hours	Overtime hours worked by employee
total_ot_paid	Total overtime pay paid to employee during fiscal eyar
total_other_pay	Includes any compensation in addition to gross salary and overtime pay during fiscal year

Based on these given features, five other features were generated:

Table 2. Generated features

Generated feature	**Description**
time_in_pos_months	Total number of months since employee started work at the agency to June 30 of the fiscal year
overall_pay	Total pay received by employee in the fiscal year, including regular_gross_paid, total_ot_paid, and total_other_pay
std_from_mean_overall_pay	Z score (number of standard deviations away from mean) of

	example overall pay away from mean of all examples of the same job title. The mean and standard deviations were those of the training set.
std_from_mean_salary	Z score of example basetime salary away from mean of all examples of the same job title. The mean and standard deviations were those of the training set.
std_from_mean_ot_hours	Z score of example overtime hours away from mean of all examples of the same job title. The mean and standard deviations were those of the training set.

Models

Four classifier models were tested: logistic regression, decision tree, random forest, and adaboost. I wanted to include logistic regression in the models tested to see if there could be a simple linear relationship found between the features and the target. Decision tree, random forest, and adaboost were chosen because the goal behind the model was to generate actionable insights for employers to retain employees. An easy way of generating such actionable insights would be if the model were explainable (i.e. feature importance is clearly shown). Feature importance can be easily extracted from these three non-linear models. Because of this focus on explainability, methods that required dimensionality reduction like PCA were ruled out for consideration since extracting feature importance from such methods would be more tedious.

Hyperparameters were tuned using cross validation with five folds. Standardization was only applied within each fold. Only one hyperparameter was tuned for each model due to time constraints. The hyperparameters tuned for each model are:

Table 3. Hyperparameters tuned for each model

Model	**Hyperparameter and range of vals**	**Optimal (all scored on accuracy)**
Logistic regression	C (inverse of regularization strength): [1e-6, 1e-4, 1e-2, 1, 10, 100, 1000, 10000]	C=1
Decision tree	max_depth: 1 to 101 in increments of 10	max_depth=11
Random forest	max_depth: 1 to 101 in increments of 10	max_depth=21
Adaboost	n_estimators: 1 to 101 in increments of 10	n_estimators=101

The thresholding alpha hyperparameter that converts a continuous probability to a discrete category prediction was **not** tuned due to time constraints. In light of this, the default thresholding alpha of 0.5 was applied to each model when making class predictions.

STAT3106 Applied Machine Learning
Philip Tham (ht2546)

Results

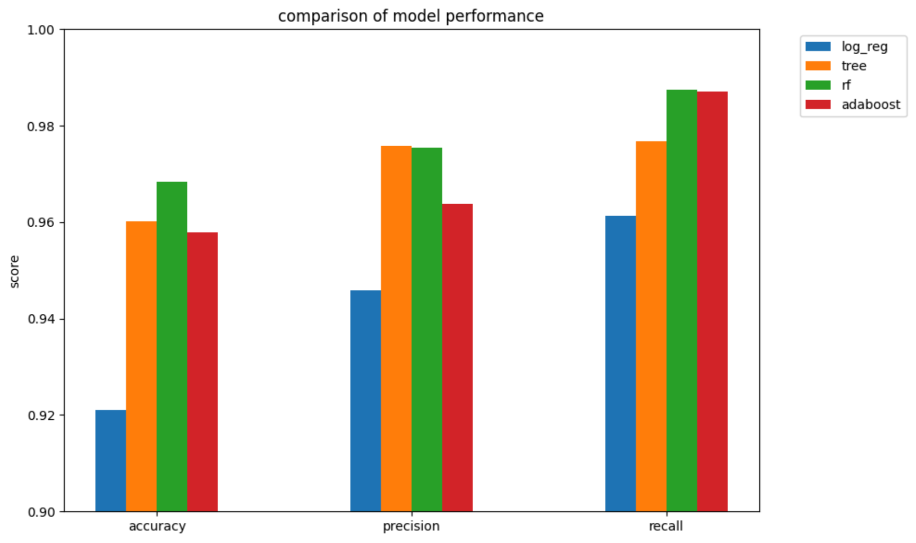
Based on the optimal hyperparameters, the four models were fit on the training dataset and then tested on the held-out testing data. The test set was first preprocessed based on data from the training set (ie the means and standard deviations of the training sets were used to calculate the z scores of the overall pay (std_from_mean_overall_pay), OT hours (std_from_mean_ot_hours), and base salary (std_from_mean_salary). The models were scored based on accuracy, precision, and recall. I chose to place more importance on the accuracy and recall scores when determining the best model. Accuracy indicates the overall correctness of the model, and recall was more important than precision because false negatives—ie predictions that wrongly classify an employed employee (a 1 prediction) as ceased (a 0 prediction)—would be costly since employers would then want to spend resources trying to retain the employee.

The results for accuracy, precision, and recall for the four models are shown in the table and graph below. Note that the dataset was imbalanced with 84% of examples being actively employed (1) and 16% of examples having ceased employment at the agency (0). Considering this, a baseline model that made random predictions based on a Bernoulli distribution with p=0.84 was also tested.

Table 4. Evaluation metrics of models against baseline

	Accuracy	**Precision**	**Recall**
log_reg	0.921064	0.945835	0.961240
tree	0.960312	0.975858	0.977005
rf	0.968594	0.975683	0.987283
adaboost	0.957930	0.963812	0.987066
baseline	0.735342	0.843009	0.842348

Graph 1. Comparison of model performance (excluding baseline)



*baseline model not included in graph because the scores were too low and would distort the scale, making differences between the four non-baseline models difficult to distinguish

All models outperformed the random Bernoulli baseline model on all three metrics, indicating that even though the dataset was imbalanced, the results of each model was still better than the baseline (though might require further statistical testing to determine significance). While the decision tree slightly outperformed other models on precision, the random forest outperformed other models on accuracy and recall. Adaboost performed only slightly worse than random forest for recall, but more significantly worse than random forest on accuracy. Given these results and the focus on accuracy and recall (give costly fase negatives), the **random forest** model was selected as the best model.

Feature importance and directionality

The feature importance of each feature in the optimized random forest model was extracted and ranked below as seen in the graph and table below.

Graph 2. Feature importance for optimized random forest model

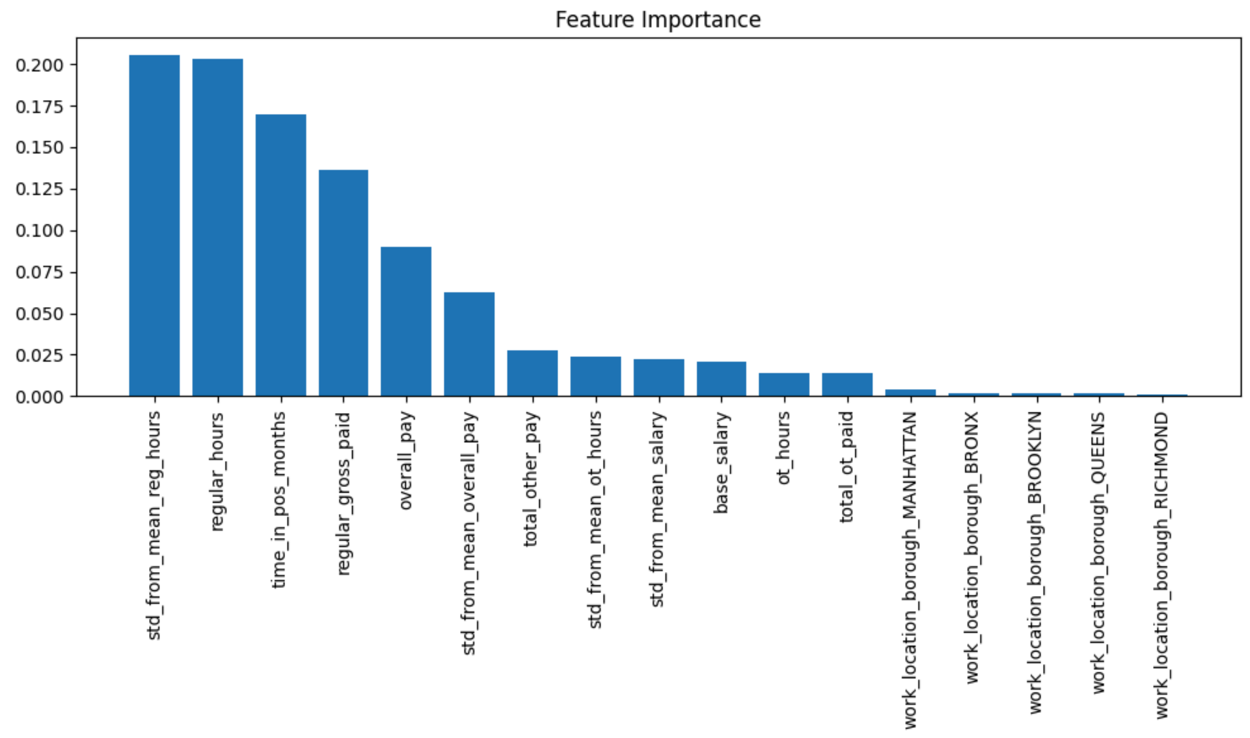


Table 5. Tabularized feature importance for optimized random forest model

Rank	**Feature**	**Importance**
1	regular_hours	0.222080
2	std_from_mean_reg_hours	0.210711
3	time_in_pos_months	0.166794
4	regular_gross_paid	0.113935
5	overall_pay	0.083639
6	std_from_mean_overall_pay	0.065946
7	total_other_pay	0.030975
8	std_from_mean_salary	0.024808
9	base_salary	0.020743
10	std_from_mean_ot_hours	0.020630
11	total_ot_paid	0.014801

12	ot_hours	0.014722
13	work_location_borough_MANHATTAN	0.004079
14	work_location_borough_BROOKLYN	0.001899
15	work_location_borough_QUEENS	0.001836
16	work_location_borough_BRONX	0.001814
17	work_location_borough_RICHMOND	0.000587

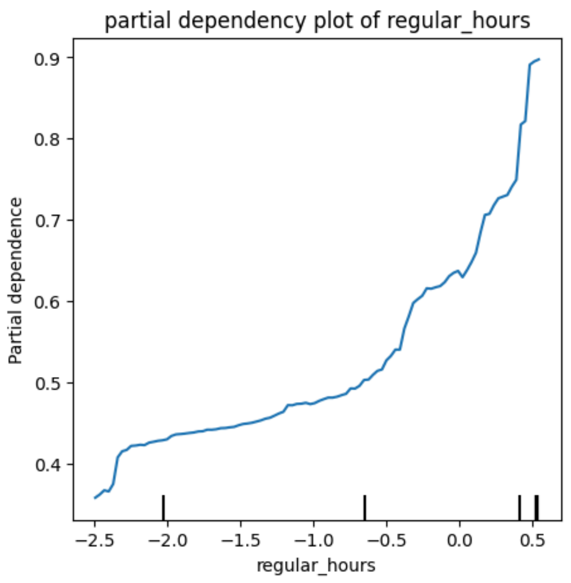
Eyeballing the feature importances from the graph, the four most important features—that is, **regular_hours, std_from_mean_reg_hours, time_in_pos, and regular_gross_paid**—seem to be the most important subset of features.

Partial dependency graphs were plotted for each of these four features to measure the impact of each feature on the prediction. Generally, there is a positive relationship between regular_hours, std_from_mean_reg_hours, and regular_gross_paid against the probability of staying at the agency (i.e. a 1 prediction). The relationship is more complex for time_in_pos_months. Ignoring a sudden initial drop in partial dependence, as the feature increases towards 0 (ie the mean across all examples since this numerical feature was scaled and centered column-wise in the preprocessing step), the relationship between time_in_pos_months and the probability of staying at the agency increases. This plateaus slightly as time_in_pos_months increases above the mean.

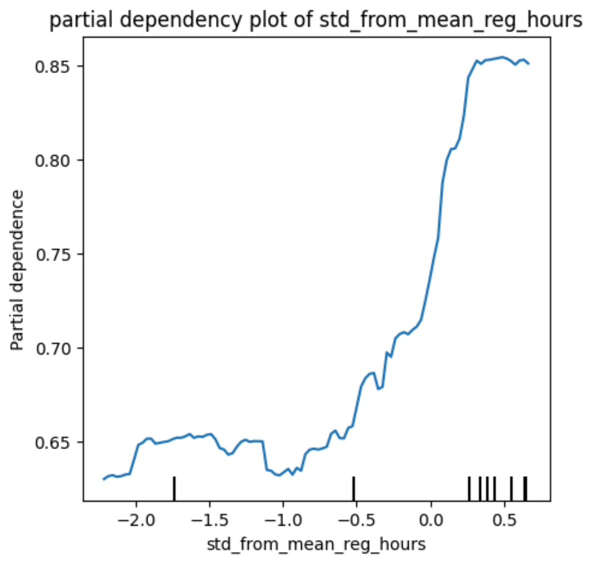
I had initially hypothesized that overall_pay and std_from_mean_overall_pay would be one of the more important features for this binary classification task. I had also hoped that this would be the case because it would make incentivizing employees to stay at the agency easier—changing the overall pay would make workers stay. Yet, these features were slightly less important to the random forest model, ranking fifth and sixth in feature importance. Nonetheless, sut of curiosity, partial dependence plots of overall_pay and std_from_mean_overall_pay (z score of each example based on the mean and standard deviation of all examples of the same job description, *not* across all examples in the dataset) were also plotted. For these two features, an increase in either feature towards 0 (the mean across all examples since these numerical features were scaled and centered column-wise in the preprocessing step as well) leads to an increase in probability to stay employed at the agency.

Graph 3. Partial dependence plots of features in optimized random forest model

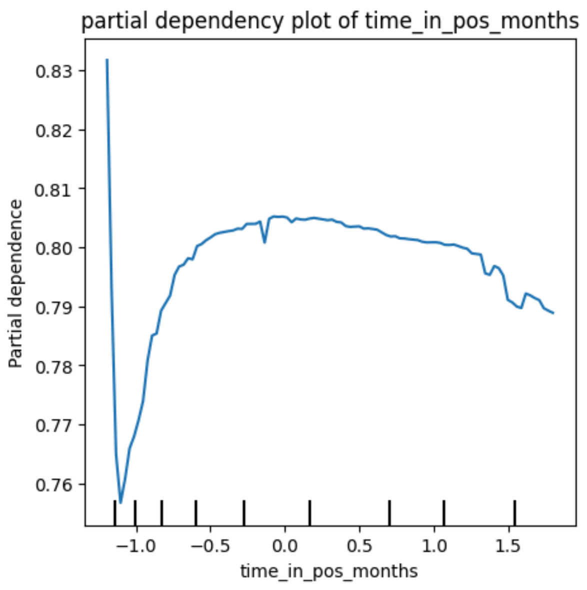
3.1 regular_hours



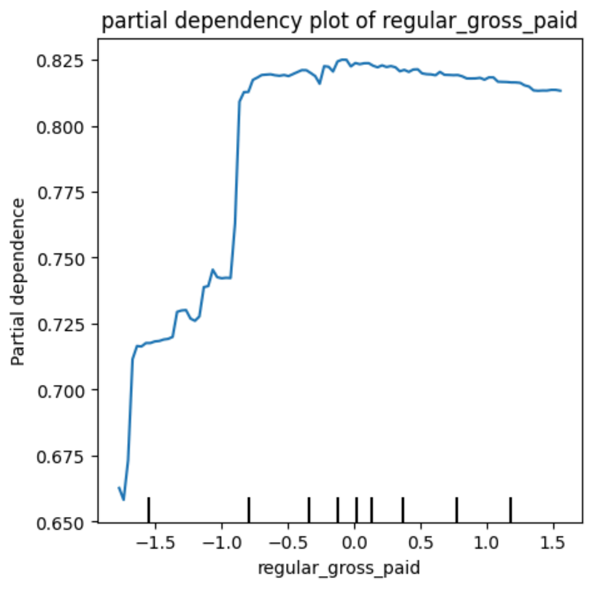
3.2. std_from_mean_reg_hours

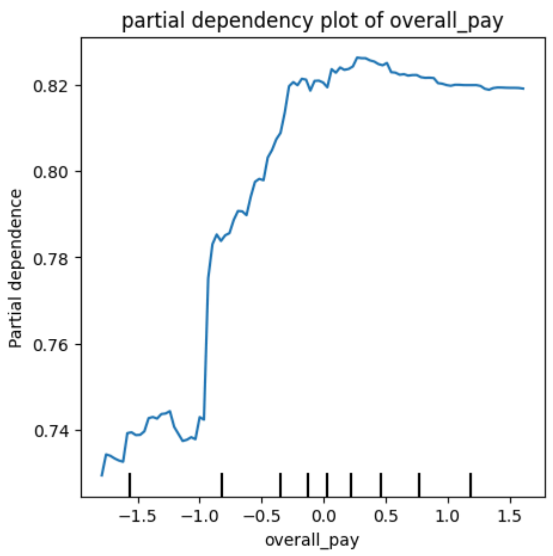
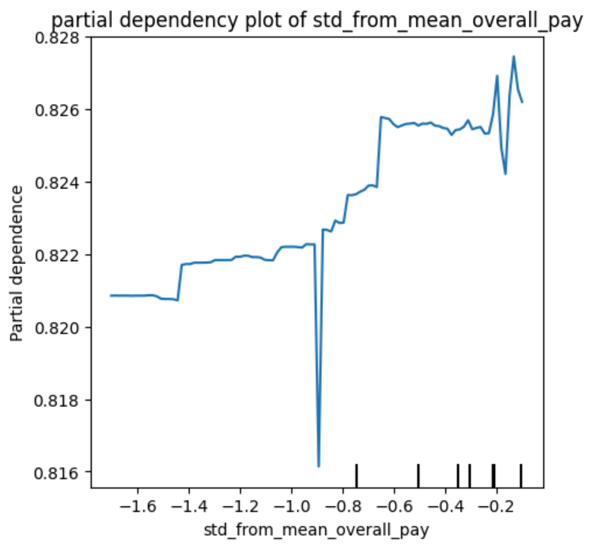


3.3 time_in_pos_months



3.4 regular_gross_paid



3.5 overall_pay 	3.6 std_from_mean_overall_pay 

Discussion and extension
This model could be further refined in several ways.

Investigate variation in feature importance across models. An initial OLS model was run on all features in the data as well to gain intuition into which features would be important in making predictions. The p values and coefficients of each feature were calculated to get a sense of feature importance. While regular_hours was a consistently important feature across the OLS and random forest models, the importance of other features varied (see the accompanying python notebook in the github repository for specific details).

- First, the OLS model calculated very low p values (highly significant) and high coefficients for dummy variables for the categorical feature work_location_borough whereas the random forest model indicated that these dummy variables were the least significant. These dummy variables were clearly important in explaining the variance in the target y—when the R-squared value of the OLS model was 0.876 but decreased to 0.091 when the dummy variables were removed.

- Second, the importance of other non-dummy features also varied between the OLS and random forest.

A first preliminary explanation could be that the OLS was only capturing linear relations between the features and the target whereas the random forest could make non-linear cuts in the data space. This explanation could make sense given the relatively worse performance of the logistic regression model (a linear model) on accuracy, precision, and recall on the test set (see *Graph 1*).

Another explanation could be that the random forest model was able to pick up on interdependencies between different features (for instance, between work_location_borough and regular_hours) and consistently decided to split by one feature over another even though both features conveyed the same information. Regression models would not be able to pick up on such interdependencies. To check if this were the case, more analysis on the correlations between different features could be done by running a tree classifier only on several important features and then examining the purity of the leaf nodes,

Investigate collinearities in features. Several features might be collinear with other features, such as the Z score features and their non-Z score counterparts. Evidence that this could be the case lies in the feature importance of Z score features generally only being slightly different from their non-Z score counterparts (for instance, regular_hours and std_from_mean_reg_hours, or overall_pay and std_from_mean_overall_pay). To further analyze this, the correlation coefficients of these different variables could be calculated to check for collinearity. If there were substantial collinearity, perhaps one solution could be to use PCA to reduce the dimensions, determining the feature importance of each reduced dimension, and then using the loadings of the rotation matrix to determine which features were decomposed to form each reduced dimension. This would still allow the model to have some form of explainability in terms of feature importance.