

Consider the `turnover.csv` data file (posted under the In-Class 14 assignment link). This file contains basic employment information of employees from some company. The goal is to build a binary classification to predict employee turnover. **In Python**, answer the following:

1. (3 points) Using the pandas library, read the csv data file and create a data-frame called `turnover`.
2. (6 points) Change `sales`, and `salary` from labels to dummy variables.
3. (6 points) Engineer the interactions/features in-class 9 assignment (the ones from the decision tree).
4. (5 points) Using `satisfaction_level`, `last_evaluation`, `number_project`, `average_monthly_hours`, `time_spend_company`, `Work_accident`, `promotion_last_5years`, `sales` (dummy variables), and `salary` (dummy variables) and interactions/features (from part 3) as the input variables and `left` as the target variable, split the data into two data-frames (taking into account the proportion of 0s and 1s) `train` (80%) and `test` (20%).
5. (8 points) Using `train` data-frame, run the RFECV algorithm with `step = 1`, `min_features_to_select = 2`, `cv = 3` and base algorithm random forest (with 500 trees and the maximum depth of each tree equal to 3). After that, build a random forest (with 500 trees and the maximum depth of each tree equal to 3) and the selected features from RFECV algorithm. Then, use this model to make predictions on the `test` data-frame. Use the provided `precision_recall_cutoff.py` (posted under the In-Class 14 Assignment link) file to estimate the optimal cutoff value. Compute the classification report of this model.
6. (8 points) Using `train` data-frame and the selected features from part (5), perform a hyper-tuning job on the random forest model. Using the [GridSearchCV](#) function and the following dictionary:

```
RF_param_grid = {'n_estimators': [100, 300, 500],
                  'min_samples_split': [10, 15],
                  'min_samples_leaf': [5, 7],
                  'max_depth' : [3, 5, 7]}
```

perform the hyper-parameter job. After that, build a random forest model with the best hyper-parameter combination. Then, use this model to make predictions on the `test` data-frame. Use the provided `precision_recall_cutoff.py` (posted under the In-Class 14 Assignment link) file to estimate the optimal cutoff value. Compute the classification report of this model. Make sure to use `scoring = 'f1'` in the `GridSearchCV` function.

7. (3 points) Using the results from part 5 and 6, what model would use to predict `left`? Be specific.