

Consider the `turnover_train.csv`, `turnover_val.csv`, and `turnover_test.csv` data files (posted under the In-Class 17 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. (4 points) Using the pandas library, read the csv data files and create three data-frames called `train`, `validation` and `test`, respectively.
2. (7 points) Change `sales`, and `salary` from labels to dummy variables in the three data-frames.
3. (7 points) Engineer the interactions/features in-class 9 assignment (the ones from the decision tree) in the three data-frames.
4. (10 points) Based on the different models built on this dataset, it seems that `interaction_3`, `interaction_1`, `satisfaction_level`, `time_spend_company`, and `number_project` are the top 5 important variables. Using `train` data-frame and the top 5 features, 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 with 3 folds. Note that based on historical data, the company estimated the following:

		Actual Class	
		0	1
Predicted Class	0	\$0	-\$1,000
	1	-\$1,500	\$500

Using the information from the above table, the cost function is given by:

$$\text{cost} = -1000 \times Y - 1500 \times Z + 500 \times W$$

where Y , Z and W represent the number of times the model predicted 0 and it was actually 1, number of times the model predicted 1 and it was actually 0, and number of times the model predicted 1 and it was actually 1, respectively. Identify the hyper-parameter combination that produces the highest cost. Then, use that model to predict the likelihood of `left` on the `validation` and `test` data-frames. Find the optimal cutoff value by comparing the likelihoods of `left` in `validation` and the actual `left` values in the `validation`. Use this cutoff to change the likelihoods of `left` in the `test` data-frame to label. Compute the cost of this prediction on the `test` data-frame.

6. (10 points) Based on the different models built on this dataset, it seems that `interaction_3`, `interaction_1`, `satisfaction_level`, `time_spend_company`, and `number_project` are the top 5 important variables. Using `train` data-frame and the top 5 features, perform a hyper-tuning job on the support vector machine model. Using the [GridSearchCV](#) function and the following dictionary:

```
SVM_param_grid = {'kernel': ['rbf', 'poly', 'sigmoid'],  
                  'C': [0.01, 0.1, 1, 10],  
                  'gamma': [0.01, 0.1, 1]}
```

perform the hyper-parameter job with 3 folds. After that, build a support vector machine model with the best hyper-parameter combination that produces the highest cost (defined in part 5). Then, use that model to predict the likelihood of **left** on the **validation** and **test** data-frames. Find the optimal cutoff value by comparing the likelihoods of **left** in **validation** and the actual **left** values in the **validation**. Use this cutoff to change the likelihoods of **left** in the **test** data-frame to label. Compute the cost of this prediction on the **test** data-frame.

7. (10 points) Using the predictions on the **validation** data-frame from parts 5 & 6, build an ensemble model (using the random forest model). Perform a hyper-parameter tuning job on the ensemble model (using the same set of hyper-parameters from part 5) and identify the model that produces the highest cost (defined in part 5). Then, use the ensemble model to predict the likelihood of **left** on the **test** data-frame. Find the optimal cutoff value by comparing the likelihoods of **left** in **validation** and the actual **left** values in the **validation**. Use this cutoff to change the likelihoods of **left** in the **test** data-frame to label. Compute the cost of this prediction on the **test** data-frame.
8. (3 points) Based on your results from parts 5, 6, and 7, what model would you use to predict **left**? Be specific.