## Algorithm outline

**Input parameters:**

* full\_dataset (default=’CSL\_d5.csv’)
* t**arget\_var**(default=“transfus\_yes”)
* **corr\_var\_list** [previous list of highly correlated vars; must contain target\_var]
* **under\_alg** (‘random’ | ‘cohort’)
* **cohort\_var** (default=None | “high\_Age” | 'Insurance' , etc)
* **pred\_alg** ( ‘LR’ | ‘SVC’ | ‘NB’ | ‘MLP’ , etc)
* **pred\_params** ( dict of algorithm-specific hyper-parameters)
* **random\_seed** (default=None)

**General Algorithm:**

1. Load **full\_dataset**
2. Filter dataset by **corr\_var\_list**
3. Set **random\_seed**
4. Create Test/Train split (use stratify = **target\_var**)
5. Perform undersampling on Training set
   1. if **under\_alg** == ‘cohort’ use **cohort\_var** [create function for this]
6. Train **pred\_alg** on Training set using p**red\_params**
7. Create prediction based on Test set
8. Output confusion matrix, F1\_score, ROC\_AUC, MCC and precision/recall for both minority and majority classes.
9. Save output to filename based on input params.

**Notes:**

* Need to do Test/Train split *before* doing undersampling. We didn’t do this with SMOTE, but should have according to what I’ve read
* Undersample only the Training set, but create prediction based on full Test set; this prevents information or data leakage
* Perform test/train split with stratify= target variable. This ensures that the testing and training data distributions with respect to the target variable are similar
* Possibly consider encoding non-binary categories using one-hot encoding.