Deliverable 3

1. Final Training Results

There is a lethal mistake which led to my 100% accuracy result. But it is fixed now thanks to Rick.(shout out to Rick!!!)

In the previous deliverable, I calculated the probability of each feature in training set, testing set and validation set separately, and use them to vectorize (replacing the string representation with the probability of those features) each set. SVM has the accuracy of 0.597 while Random Forest gets an accuracy of 0.624.

Data using local prob,RFC(`n_estimators=100, max_depth=1000, min_samples_split = 2, max_features = 8 ,random_state=0`*)*
Recall score of test set is 0.6077673025433386
Recall score of validation set is 0.5748963747723886
Accuracy score of test set is 0.6242633871380989
Accuracy score of validation set is 0.6136101499423299
Precision score of test set is 0.6028764158882232
Precision score of validation set is 0.572821170317417

Data using local prob,SVM
Recall score of test set is 0.5374966440230432
Recall score of validation set is 0.539767257076643
Accuracy score of test set is 0.5976172175249808
Accuracy score of validation set is 0.5838139177239523
Precision score of test set is 0.5437842200970274
Precision score of validation set is 0.538901916433604

This time, I calculated the probability of each feature across the whole data set(prob_x_overall) and map them to each set. And Random Forest has the accuracy of 0.667 and SVM has the accuracy of 0.580 with global probability.

Data using global prob,SVM
Recall score of test set is 0.5432800784163668
Recall score of validation set is 0.5595469293163383
Accuracy score of test set is 0.5803228285933897
Accuracy score of validation set is 0.5934256055363322
Precision score of test set is 0.5437349260494736
Precision score of validation set is 0.5565993680220014

Data using local prob,RFC(`n_estimators=100, max_depth=1000, min_samples_split = 2, max_features = 8 ,random_state=0`*) so far the best hyperparameter*
Recall score of test set is 0.6486947691801853
Recall score of validation set is 0.6218586326767092
Accuracy score of test set is 0.6673071995900589
Accuracy score of validation set is 0.6434063821607074
Precision score of test set is 0.6441838660788095
Precision score of validation set is 0.6139071728678148

Next use local prob on training but global on test and valid

SVM
Recall score of test set is 0.5456733277998936
Recall score of validation set is 0.5435389836119848
Accuracy score of test set is 0.5935178068152703
Accuracy score of validation set is 0.5822760476739716
Precision score of test set is 0.5489918611208486
Precision score of validation set is 0.5418213986386926

RFC
Recall score of test set is 0.6169296613516422
Recall score of validation set is 0.5874179771560999
Accuracy score of test set is 0.6360491929285165
Accuracy score of validation set is 0.6128412149173394
Precision score of test set is 0.6126948706431417
Precision score of validation set is 0.5817556822685479

Finally use global on training but local on test and valid
SVM
Recall score of test set is 0.5412367244869201
Recall score of validation set is 0.5542436682668432
Accuracy score of test set is 0.5894183961055598
Accuracy score of validation set is 0.6026528258362168
Precision score of test set is 0.5442329030193108
Precision score of validation set is 0.5544133937391535

RFC
Recall score of test set is 0.6392760098249222
Recall score of validation set is 0.612001324284059
Accuracy score of test set is 0.6578273123238535

Accuracy score of validation set is 0.6299500192233757
Precision score of test set is 0.6347094823918228
Precision score of validation set is 0.6034774130646734

Basically, to control the variable, we should only use a unified prob,which is the global prob. It doesn't make any sense to use local prob.

Now focused on RFC, I tried to adjust hyperparameters. To do this, I used the ideas in https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74 using RandomSearchCV

Implementing RFC `rfc = RandomForestClassifier(n_estimators=20, max_depth=406, min_samples_split = 2, max_features = 8 ,random_state=42)`

Recall score of test set is 0.6497395889762116
Recall score of validation set is 0.6216859791425261
Accuracy score of test set is 0.6651293876505252
Accuracy score of validation set is 0.6418685121107266
Precision score of test set is 0.6437820987942775
Precision score of validation set is 0.6133299015465703

Very close to my really randomly set hyperparameters.

Now change `random_state to 0`

```
Recall score of test set is 0.6526024442382095
Recall score of validation set is 0.6224914749213706
Accuracy score of test set is 0.6665385600819882
Accuracy score of validation set is 0.6399461745482506
Precision score of test set is 0.6460147321537821
Precision score of validation set is 0.613335952415414
```

Doesn't really have any impact

Changing n_estimators=100, basically comes back to initial model.
Therefore, n estimator is still the most important parameter.

```
Recall score of test set is 0.6486947691801853
Recall score of validation set is 0.6218586326767092
Accuracy score of test set is 0.6673071995900589
Accuracy score of validation set is 0.6434063821607074
Precision score of test set is 0.6441838660788095
```

Precision score of validation set is 0.6139071728678148

Now run grid search, found the following best combination

Implementing RFC`rfc = RandomForestClassifier(n_estimators=300, max_depth=456, min_samples_split = 2, max_features = 8 ,random_state=0)`

Performance dropped a little bit.

Recall score of test set is 0.6488521688509497
Recall score of validation set is 0.6198308227114717
Accuracy score of test set is 0.6669228798360236
Accuracy score of validation set is 0.6407151095732411
Precision score of test set is 0.6440865032664119
Precision score of validation set is 0.6117487829784476

Again use 406 for max_depth: `rfc = RandomForestClassifier(n_estimators=300, max_depth=406, min_samples_split = 2, max_features = 8 ,random_state=0)` didn't change from last at all

Recall score of test set is 0.6488521688509497
Recall score of validation set is 0.6198308227114717
Accuracy score of test set is 0.6669228798360236
Accuracy score of validation set is 0.6407151095732411
Precision score of test set is 0.6440865032664119
Precision score of validation set is 0.6117487829784476

Try `rfc = RandomForestClassifier(n_estimators=150, max_depth=406, min_samples_split = 2, max_features = 8 ,random_state=0)`

Recall score of test set is 0.6498414316428966
Recall score of validation set is 0.6199637477238868
Accuracy score of test set is 0.6673071995900589
Accuracy score of validation set is 0.6403306420607459
Precision score of test set is 0.6447977882619155
Precision score of validation set is 0.6117259343640719

Basically, 0.667 is the upper limit. And GridSearch doesn't seem to provide any help. As n-estimator goes up beyond 100 to 150, the performance decreases, though only by 0.001 to 0.002 or so. And max_depth

didn’t affect the result too much, because with the same n-estimator, max_depth = 406 or 1000 doesn’t make much difference.

Wait!!!
rfc = RandomForestClassifier(n_estimators=500, max_depth=406, min_samples_split = 2, max_features = 8 ,random_state=0)
There is 0.001 improvement

Recall score of test set is 0.6497469421091044
Recall score of validation set is 0.6201324284058931
Accuracy score of test set is 0.6688444786062003
Accuracy score of validation set is 0.641676278354479
Precision score of test set is 0.6454783195662257
Precision score of validation set is 0.6122364518854957

Run Grid search again, this time it tells me the best n-estimator is 1000
rfc = RandomForestClassifier(n_estimators=1000, max_depth=406, min_samples_split = 2, max_features = 8 ,random_state=0)
But it took too long to run, so it’s not very practical,also the performance worsened

Recall score of test set is 0.6512309606253588
Recall score of validation set is 0.6203426585002483
Accuracy score of test set is 0.6661542403279529
Accuracy score of validation set is 0.6382160707420222
Precision score of test set is 0.6450651002579069
Precision score of validation set is 0.6113976537216829

rfc = RandomForestClassifier(n_estimators=750, max_depth=406, min_samples_split = 2, max_features = 8 ,random_state=0)
So much worse than 1000 estimators

Recall score of test set is 0.6471825103709368
Recall score of validation set is 0.6164875020691938
Accuracy score of test set is 0.6637202152190622
Accuracy score of validation set is 0.6357170319108035

```
Precision score of test set is 0.6417295561169648
Precision score of validation set is 0.608075718935954

No,500 is the maximum
```

2. Final demonstration proposal

- Key idea: This model is to predict a prisoner's tendency to reoffend after they are released so that the judge can decide how much sentence to give. Different factors, including ethnicity, age and type of crime they committed and so on, have within them more subtypes with different probabilities of a recidivist.

- Method: Introduction of SVM and Random Forest working mechanism based on lectures and sklearn documentations.
- Results and future research: For results there should be two parts, the first is prediction's accuracy is approximately 0.667, the second is the probability distribution of different factors of recidivists.
  For future research, the dataset can be further expanded and the model can be deployed into practice as an assistive tool for legal procedures.
- References as convention.