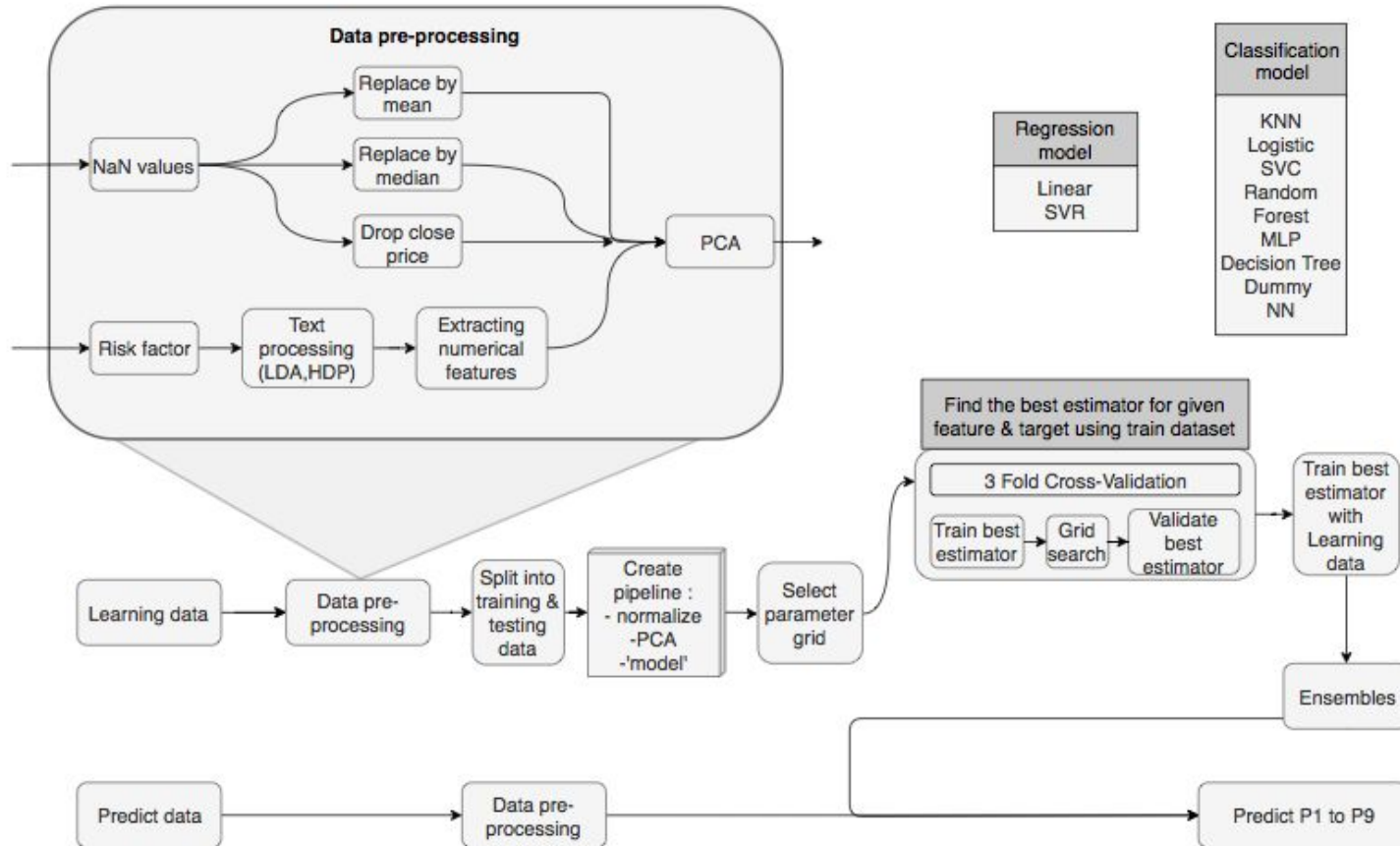


Data Science For Business

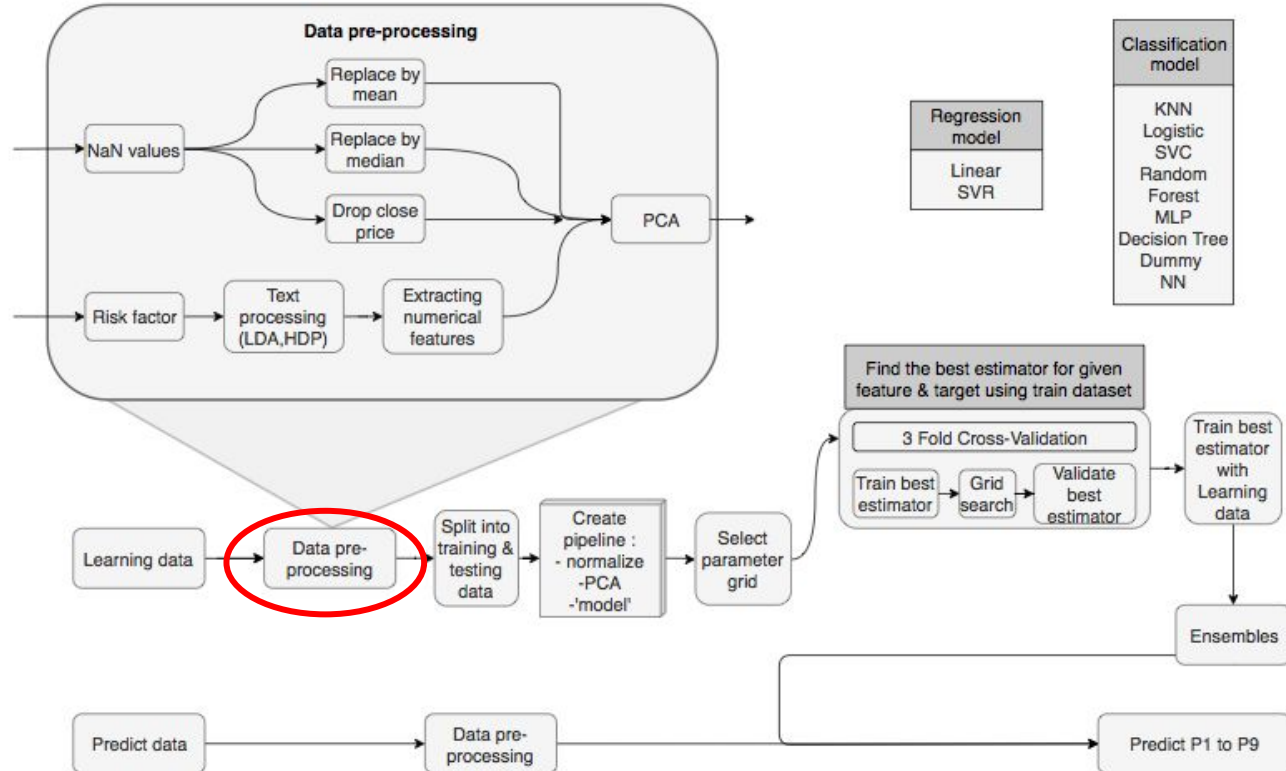
Learning to predict IPO performance... Data Science?

Claire Laurent, Devakumar Thammisetty, Cyprien Mercier

Prediction process : Overview



Data pre-processing



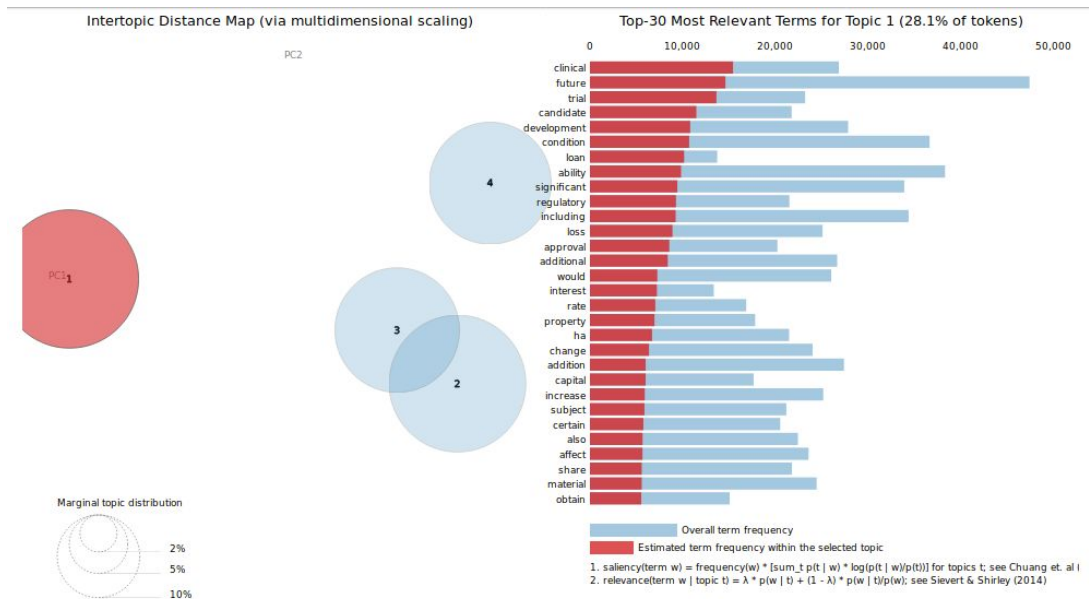


Given Learn data

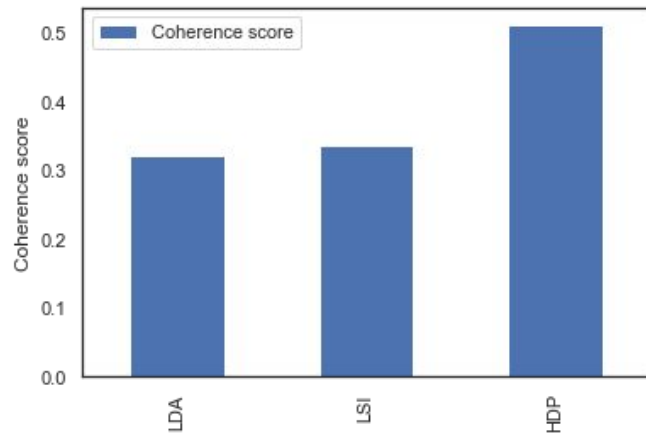
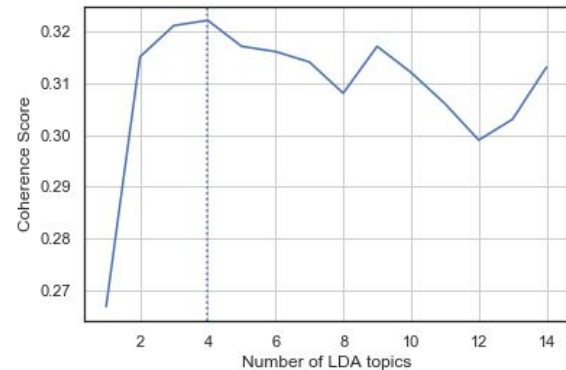


Learn data **After** pre-processing

Processing of the text data (LDA)



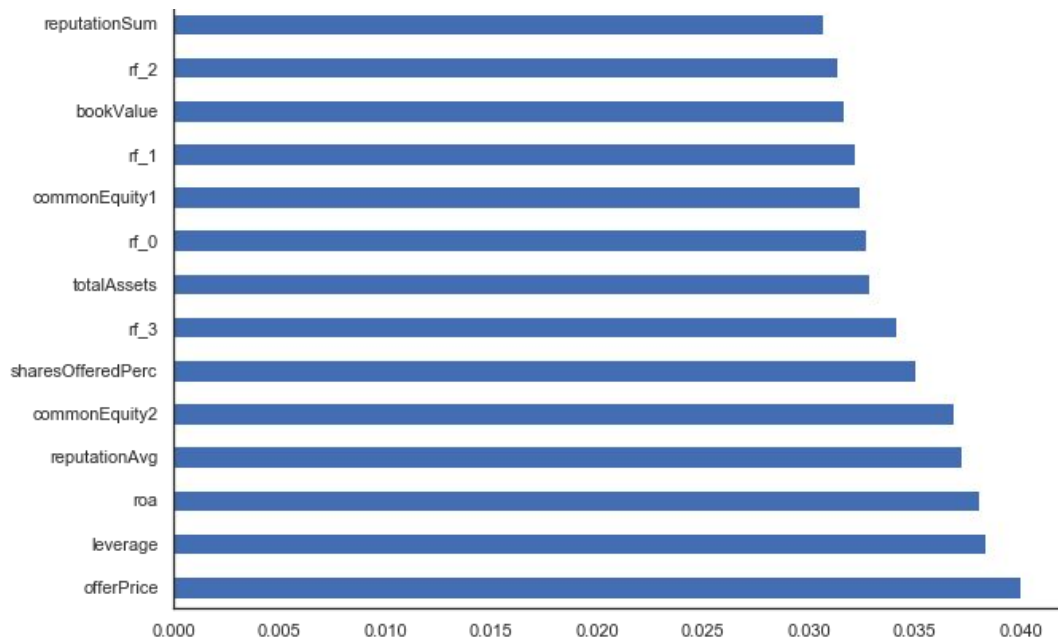
	LDA	LSI	HDP
Coherence score	0.321608	0.33685	0.511364



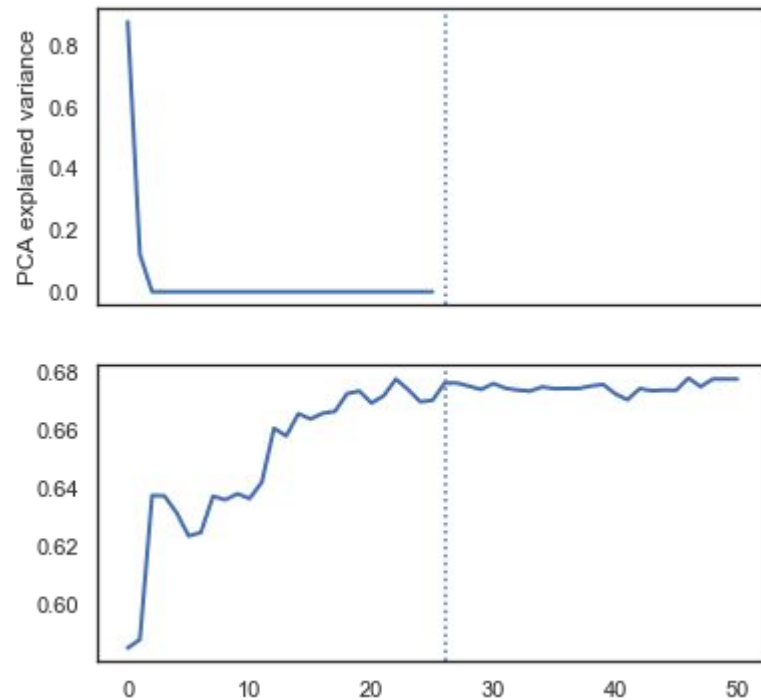
Data reduction and extraction of important features



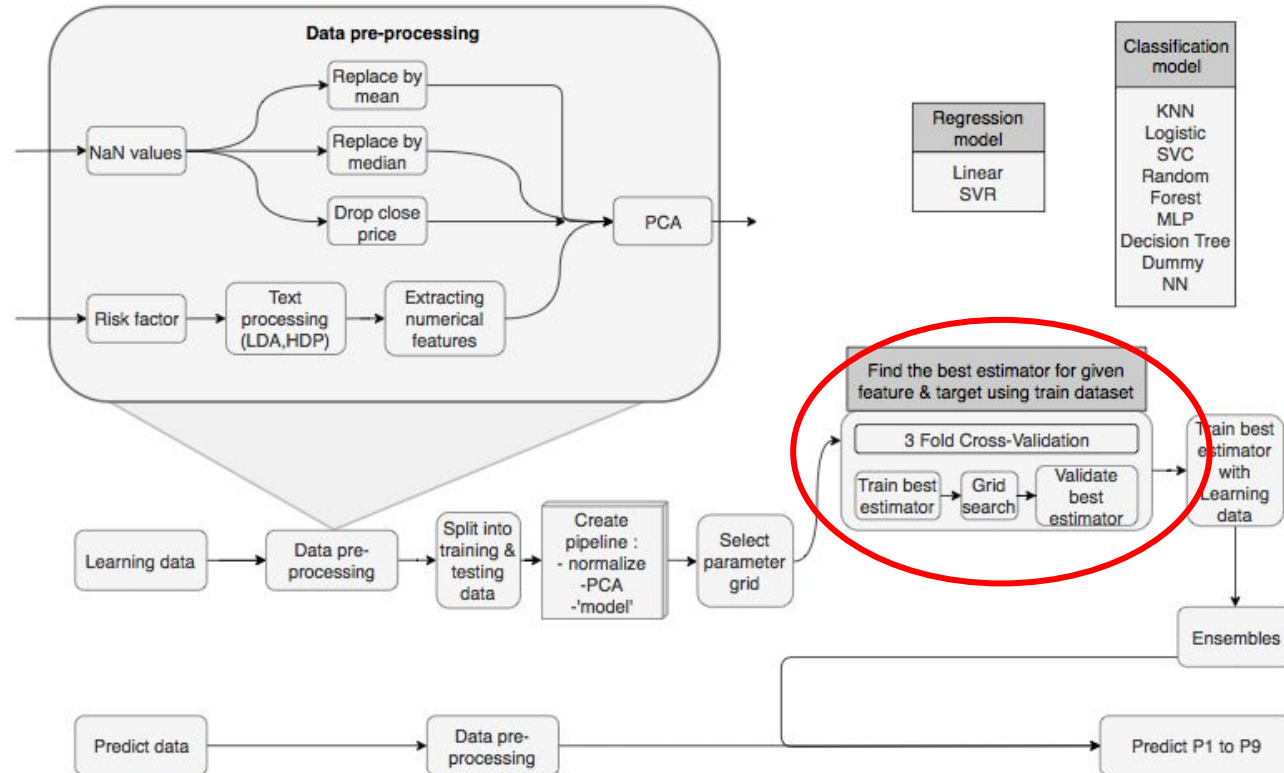
Features importance from Random Forests



PCA



Training - cross validation - selection of best estimator



Process used for the predictions

Define X and Y

For each prediction, either the predictors or the value to predict change. Define them for the 9 predictions

Define the models to try

Have tried:

- random
- baseline
- linear
- logit (lasso)
- decision tree
- random forest
- KNN
- SVC
- SVR
- CNN

Define best hyper-parameter

For **each** model: use Stratified Kfold and Grid search to determine the best hyper-parameters. Scoring metrics used: ROC, AUC, R2 and F1-score.

Validation

Draw the validation curve in order to check if the hyperparameters are good or not

Test

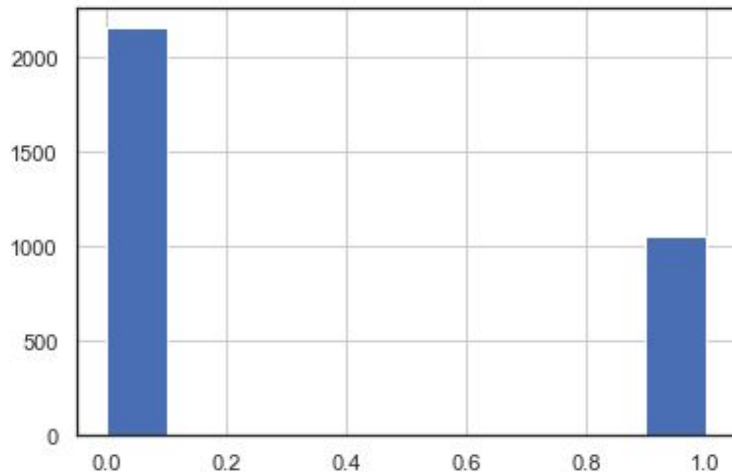
Apply the model to the test set, compute metrics and compare models with each others.

Do the steps for each model and decide which model to use for each prediction. Then use it to predict Y.

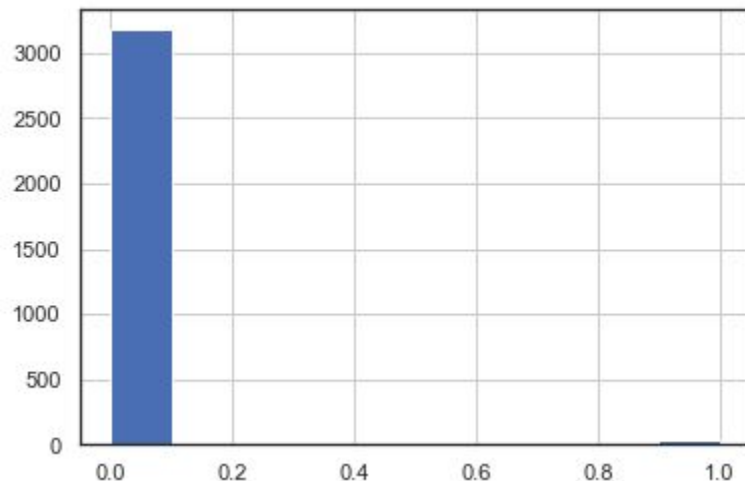
Scoring metrics - Selection



The price go up by more than 20%
(P4)

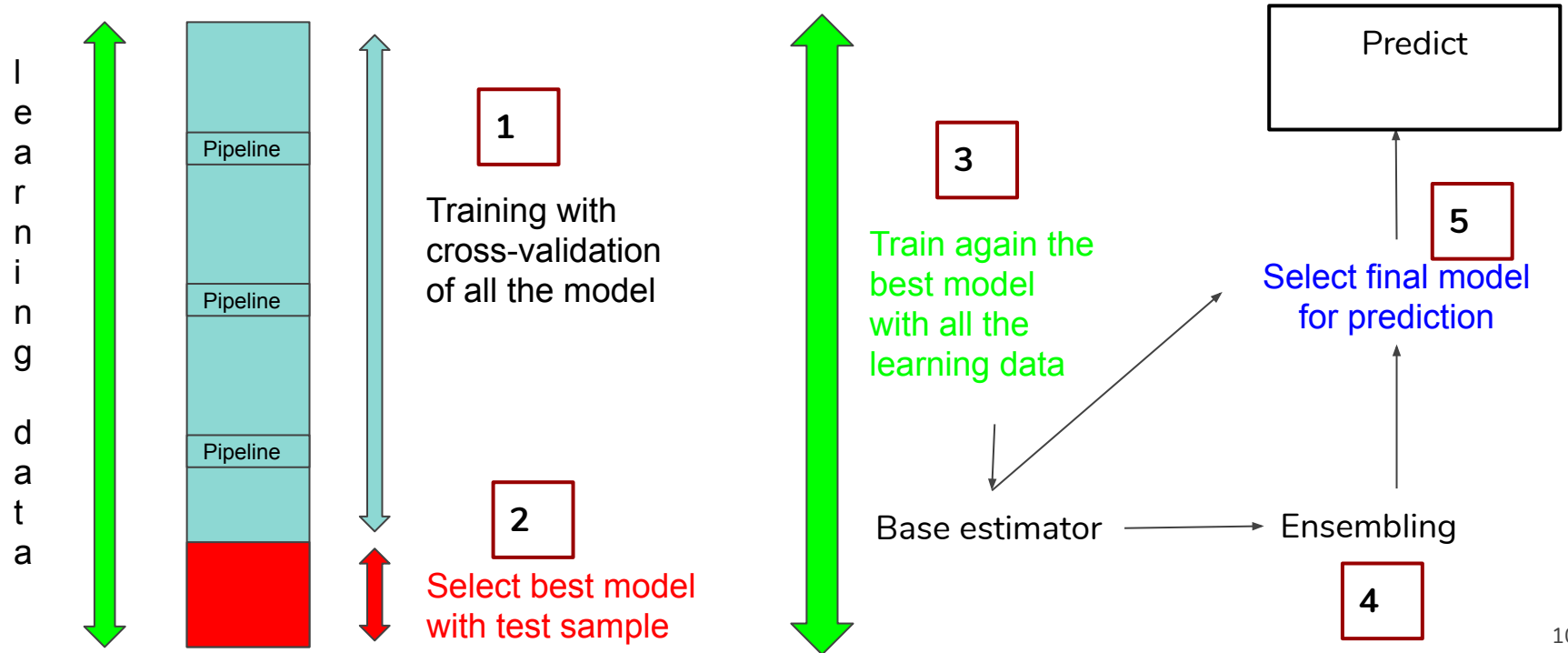


The price go down by more than 20%
(P5)



→ Two target variable distribution widely different, necessity of choosing the right metric (Accuracy will choose dummy classifier for P5)

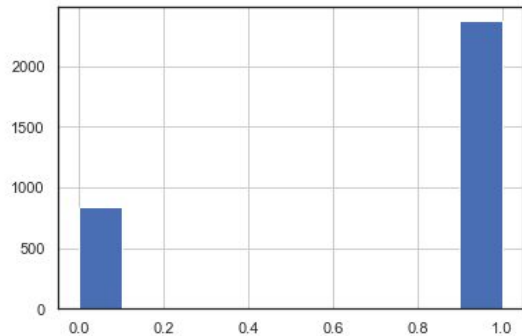
Final prediction - Steps from best estimator



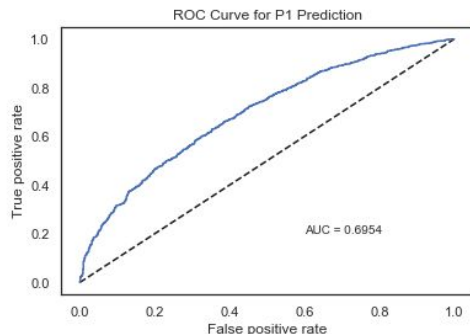
Best Model selection process: Example for P1



our objective:



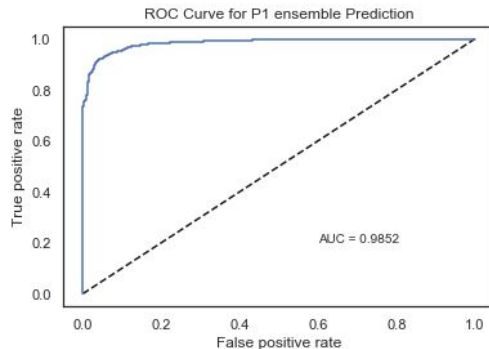
AUC with MLP classifier



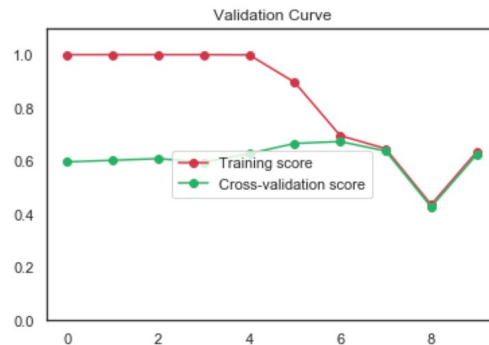
Hyper-parameter tuning

	roc_auc
MLPClassifier	0.673
LogisticRegression	0.670
RandomForestClassifier	0.670
KNeighborsClassifier	0.641
SVC	0.640
DecisionTreeClassifier	0.591
DummyClassifier	0.500
DummyClassifier-1	0.500

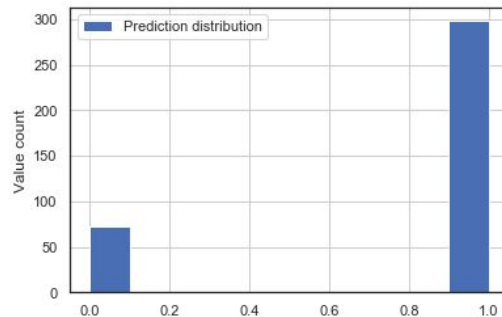
Ensembles



cross-validation curve :



Our Prediction



Our prediction - Selection of best estimators

Features	'rf'	'rf'	all	all	all	all	all	all	all
Targets	+/-	+/-	+/-	>20%	<-20%	\$	p>5%	p>50%	p<10%
ROC-AUC score	P1	P2	P3	P4	P5	P6	P7	P8	P9
DecisionTreeClassifier	0.596	0.559	0.621	0.744	0.492	nan	-126	-126	-1.18e+03
DummyClassifier	0.5	0.5	0.5	0.5	0.5	nan	nan	nan	nan
DummyClassifier-1	0.5	0.5	0.5	0.5	0.5	nan	nan	nan	nan
KNeighborsClassifier	0.641	0.576	0.641	0.79	0.738	nan	-122	-122	-1.18e+03
LinearRegressor	nan	nan	nan	nan	nan	-0.524	nan	nan	nan
LogisticRegression	0.67	0.591	0.681	0.8	0.791	nan	-121	-121	-631
MLPClassifier	0.673	0.594	0.655	0.791	0.804	nan	-120	-121	-1.18e+03
RandomForestClassifier	0.643	0.569	0.646	0.788	0.716	nan	-124	-120	-1.17e+03
SVC	0.633	0.535	0.643	0.782	0.779	nan	-123	-122	-1.18e+03
SVR	nan	nan	nan	nan	nan	-0.341	nan	nan	nan



Conclusions

- ❑ Learned IPO processes and predicted various events using data science techniques.
- ❑ Built robust classifiers for various predictions and cross validated them on learn data, provided best possible predictions :).
- ❑ Obtained AUC as high as 0.8 to predict IPOs that give >20% profit. In addition, also obtained similar prediction capability for IPOs that go down by 20%.
- ❑ Improved performance with Ensembling
- ❑ Logistic regression, Random forests and Neural net based classifiers are found to give similar performance. However, Neural net based classifier turned out good for (4 of 9 predictions) followed by Logistic regression (3 out of 9).



Thank you

Our prediction using Heirarchical Dirichlet

Features	'rf'	'rf'	all	all	all	all	all	all	all
Targets	+/-	+/-	+/-	>20%	<-20%	\$	p>5%	p>50%	p<10%
ROC-AUC score	P1	P2	P3	P4	P5	P6	P7	P8	P9
DecisionTreeClassifier	0.608	0.558	0.556	0.75	0.687	nan	-129	-129	-1.19e+03
DummyClassifier	0.5	0.5	0.5	0.5	0.5	nan	nan	nan	nan
DummyClassifier-1	0.5	0.5	0.5	0.5	0.5	nan	nan	nan	nan
KNeighborsClassifier	0.641	0.612	0.657	0.791	0.678	nan	-122	-122	-1.18e+03
LinearRegressor	nan	nan	nan	nan	nan	-0.521	nan	nan	nan
LogisticRegression	0.67	0.611	0.69	0.798	0.714	nan	-120	-121	-631
MLPClassifier	0.674	0.598	0.688	0.799	0.721	nan	-119	-120	-1.18e+03
RandomForestClassifier	0.665	0.623	0.636	0.784	0.739	nan	-124	-122	-1.17e+03
SVC	0.59	0.556	0.622	0.765	0.829	nan	-125	-124	-1.18e+03
SVR	nan	nan	nan	nan	nan	-0.0892	nan	nan	nan

Our prediction - F1 score

Features	'rf'	'rf'	all	all	all	all	all	all	all
Targets	+/-	+/-	+/-	>20%	<-20%	\$	p>5%	p>50%	p<10%
F1 score	P1	P2	P3	P4	P5	P6	P7	P8	P9
DecisionTreeClassifier	0.84	0.84	0.84	0.485	0	nan	-126	-126	-1.17e+03
DummyClassifier	0.84	0.84	0.84	0.494	0.0185	nan	nan	nan	nan
DummyClassifier-1	0.84	0.84	0.84	0	0	nan	nan	nan	nan
KNeighborsClassifier	0.84	0.84	0.841	0.546	0	nan	-122	-122	-1.18e+03
LinearRegressor	nan	nan	nan	nan	nan	-0.522	nan	nan	nan
LogisticRegression	0.838	0.84	0.839	0.572	0	nan	-121	-121	-631
MLPClassifier	0.84	0.84	0.834	0.595	0	nan	-120	-120	-1.19e+03
RandomForestClassifier	0.827	0.791	0.819	0.552	0	nan	-122	-121	-1.18e+03
SVC	0.84	0.84	0.831	0.525	0	nan	-123	-123	-1.17e+03
SVR	nan	nan	nan	nan	nan	-0.348	nan	nan	nan



Back up

	P1	P2	P3	P4	P5	P6	P7	P8	P9
DecisionTreeClassifier	0.608	0.558	0.556	0.75	0.687	nan	-129	-129	-1.19e+03
DummyClassifier	0.5	0.5	0.5	0.5	0.5	nan	nan	nan	nan
DummyClassifier-1	0.5	0.5	0.5	0.5	0.5	nan	nan	nan	nan
KNeighborsClassifier	0.641	0.612	0.657	0.791	0.678	nan	-122	-122	-1.18e+03
LinearRegressor	nan	nan	nan	nan	nan	-0.521	nan	nan	nan
LogisticRegression	0.67	0.611	0.69	0.798	0.714	nan	-120	-121	-631
MLPClassifier	0.674	0.598	0.688	0.799	0.721	nan	-119	-120	-1.18e+03
RandomForestClassifier	0.665	0.623	0.636	0.784	0.739	nan	-124	-122	-1.17e+03
SVC	0.59	0.556	0.622	0.765	0.829	nan	-125	-124	-1.18e+03
SVR	nan	nan	nan	nan	nan	-0.0892	nan	nan	nan

Model vs Predictor

	P1	P2	P3	P4	P5	P6	P7	P8	P9
Logit									
Baseline									
Linear									
Knn									
SVC									
Decision Tree									
Random F									
SVR									
MLP									



Processing of the data training set

- drop columns with high cardinality (city and manager)
- decide to only use industryFF12: drop industryFF5 and industryFF48
- create dummy variables with industryFF12 and exchange
- create new variables:
 - **return** = closeDay1-offerPrice/OfferPrice
 - **raisingPrice** = 1 if return > 0 and 0 otherwise
- Processed text data inside rf field: remove punctuation, lemmatize the text, remove a custom list of stopwords
- Use LDA model to define topics and extract feature vector for each observation

Processing of the data to predict

Same process but don't drop any lines

Profile report and observations

- Presence of missing values: we have to either drop or process the missing fields
- High correlation among 5 fields : we may decide to ignore them, drop them, or use PCA to reduce the dimensionality
- Different scales, ranging from 0 to 1e9: we need to normalize the data
- Missing outcome: offerPrice(3.5%) and closeDay1(3.5%). Since there is no outcome, it may not be useful to use this data, so we may drop the corresponding rows.
- Only 22% of the companies are marked as emerging growth companies so there may have a bias.
- Most of the companies are listed in NASDAQ(2368), followed by NYSE(895)
- Data is present from 1996 to 2018, there is more data in the late 90s, but data is well spread across years.
- Five fields are skewed (totalProceeds, InvestmentReceived, commonEquity1, totalRevenue, nPatents)
- 19 fields out of 47 have missing entries. Highest missing entries in investmentReceived(45%) followed by nExecutives, priorFinancing, nVCs, patRatio, managementFee(32.9%) in descending order

Profile report and observations

Dataset info

Number of variables	47
Number of observations	3330
Total Missing (%)	8.8%
Total size in memory	1.1 MiB
Average record size in memory	341.0 B

Variables types

Numeric	28
Categorical	7
Boolean	6
Date	0
Text (Unique)	1
Rejected	5
Unsupported	0

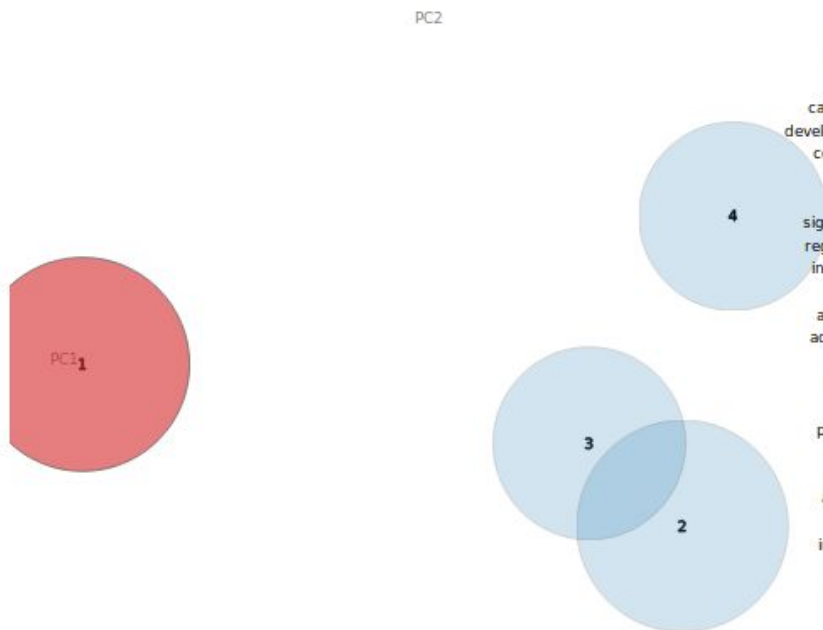
Dataset info

Number of variables	55
Number of observations	370
Total Missing (%)	22.7%
Total size in memory	146.5 KiB
Average record size in memory	405.3 B

Variables types

Numeric	24
Categorical	7
Boolean	6
Date	0
Text (Unique)	1
Rejected	17
Unsupported	0

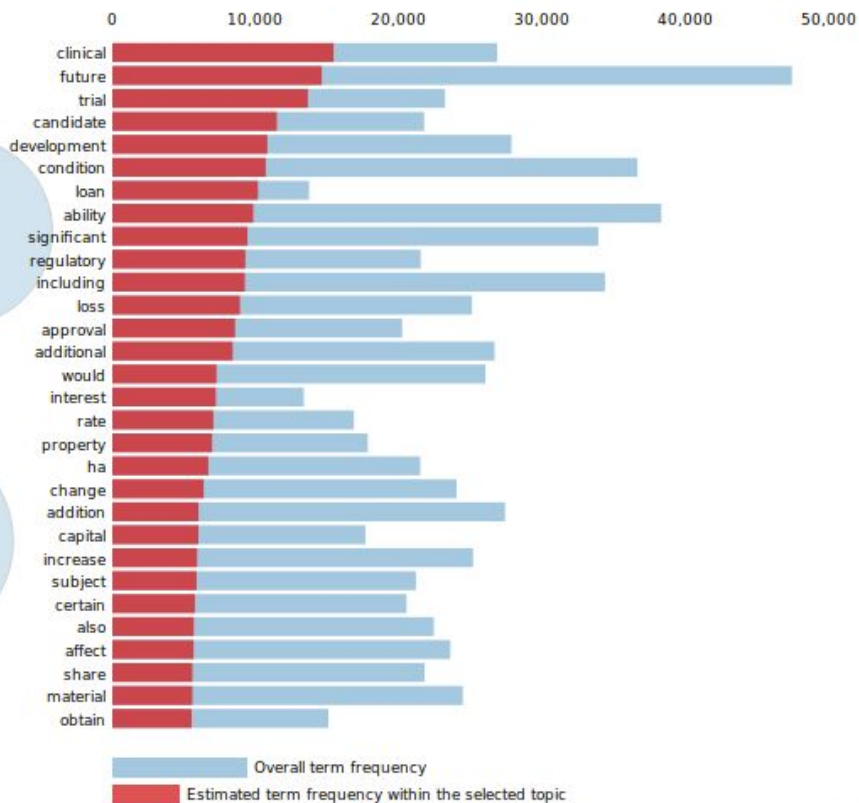
Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

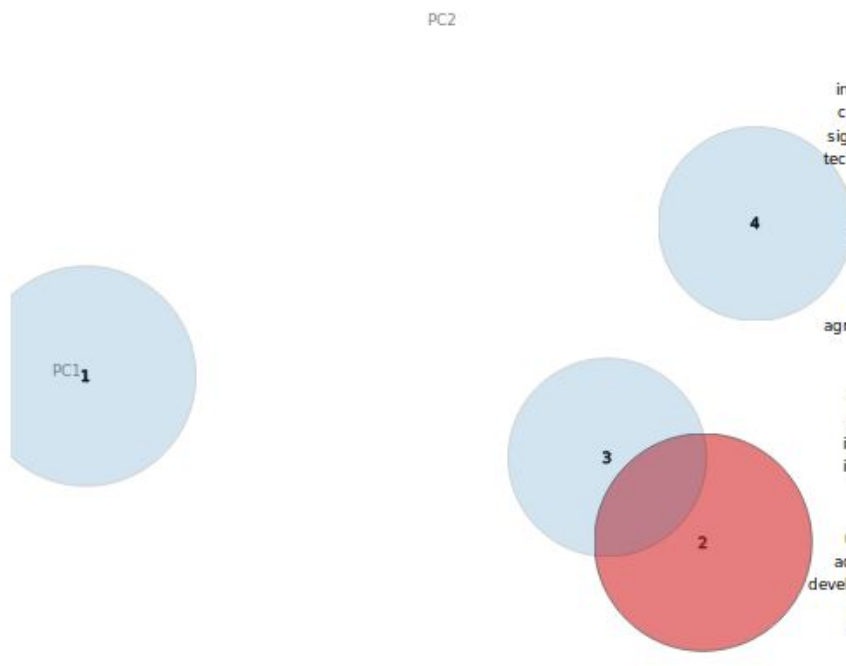


Top-30 Most Relevant Terms for Topic 1 (28.1% of tokens)



1. $\text{saliency}(\text{term } w) = \text{frequency}(w) * [\sum_t p(t | w) * \log(p(t | w)/p(t))]$ for topics t ; see Chuang et. al
2. $\text{relevance}(\text{term } w | \text{topic } t) = \lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

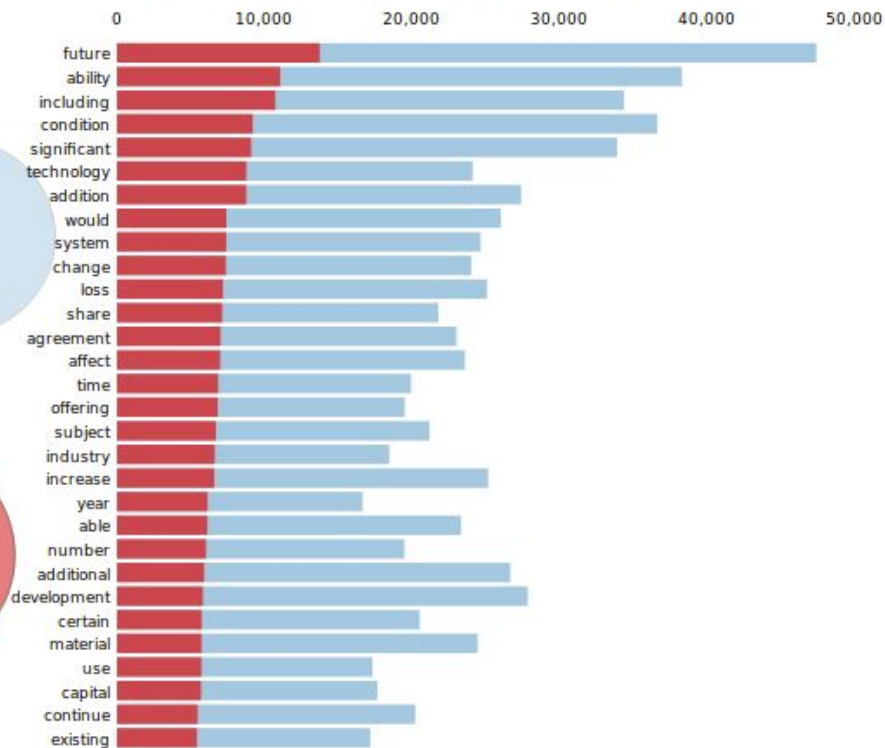
Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution



Top-30 Most Relevant Terms for Topic 2 (27.4% of tokens)



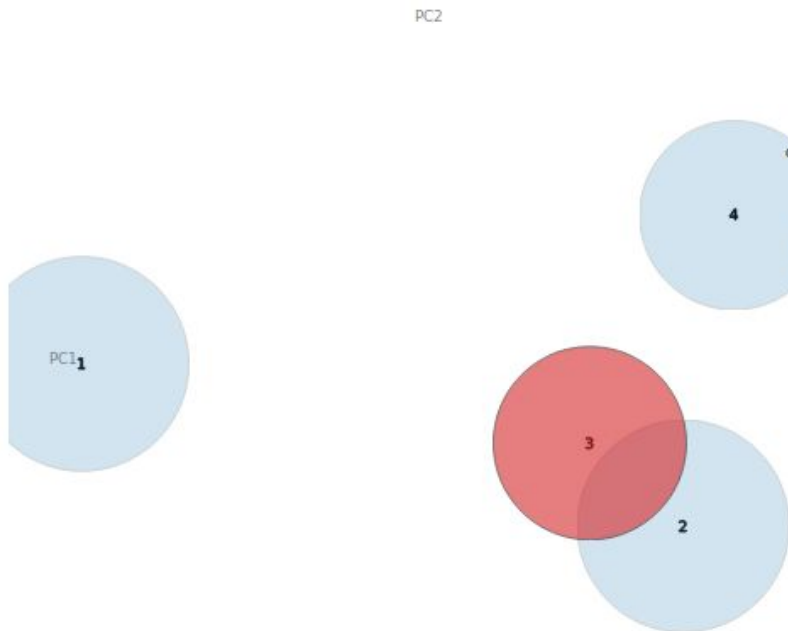
Overall term frequency

Estimated term frequency within the selected topic

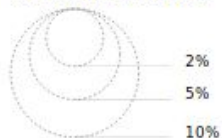
1. $\text{saliency}(\text{term } w) = \text{frequency}(w) * [\sum_t p(t | w) * \log(p(t | w)/p(t))]$ for topics t ; see Chuang et. al

2. $\text{relevance}(\text{term } w | \text{topic } t) = \lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

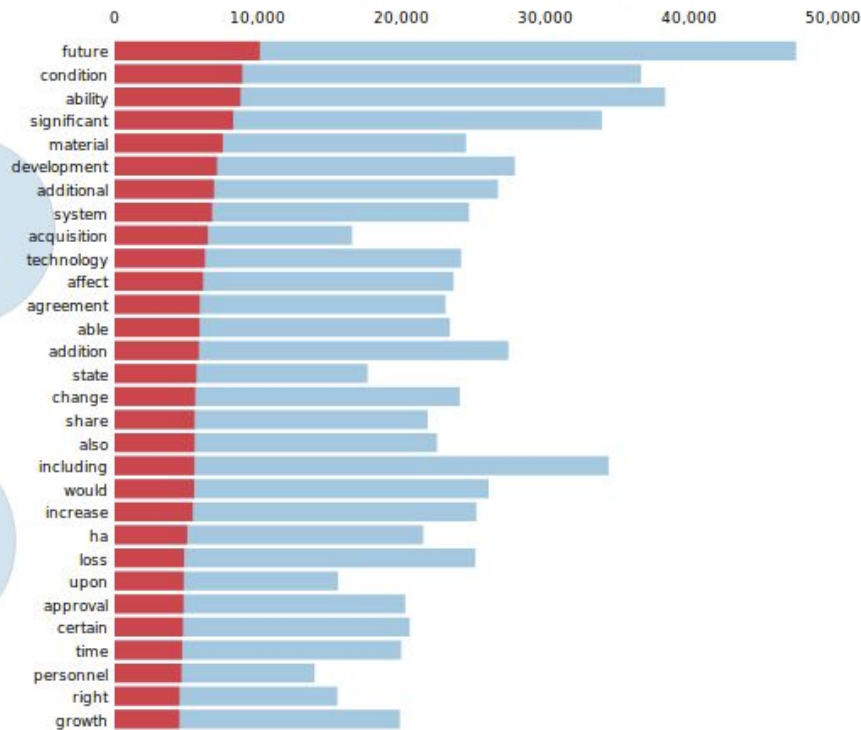
Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution



Top-30 Most Relevant Terms for Topic 3 (22.7% of tokens)



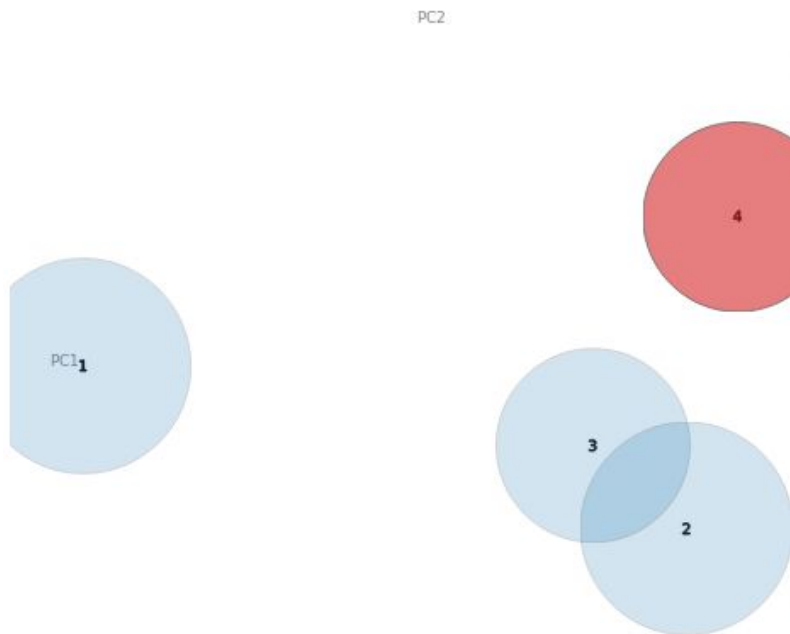
Overall term frequency

Estimated term frequency within the selected topic

1. $\text{saliency}(\text{term } w) = \text{frequency}(w) * [\sum_t p(t | w) * \log(p(t | w)/p(t))]$ for topics t ; see Chuang et. al

2. $\text{relevance}(\text{term } w | \text{topic } t) = \lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

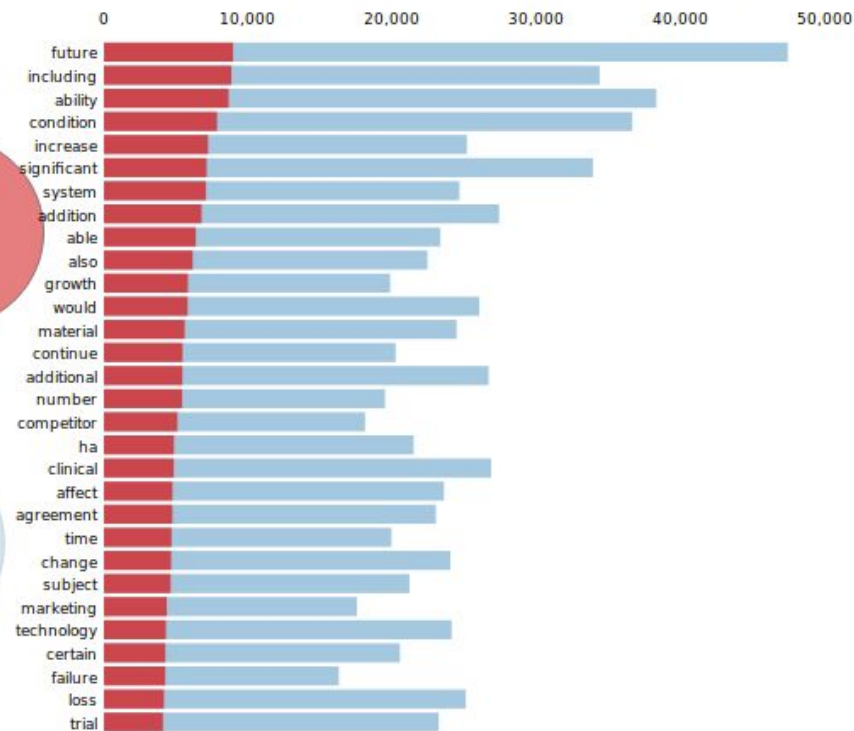
Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution



Top-30 Most Relevant Terms for Topic 4 (21.8% of tokens)

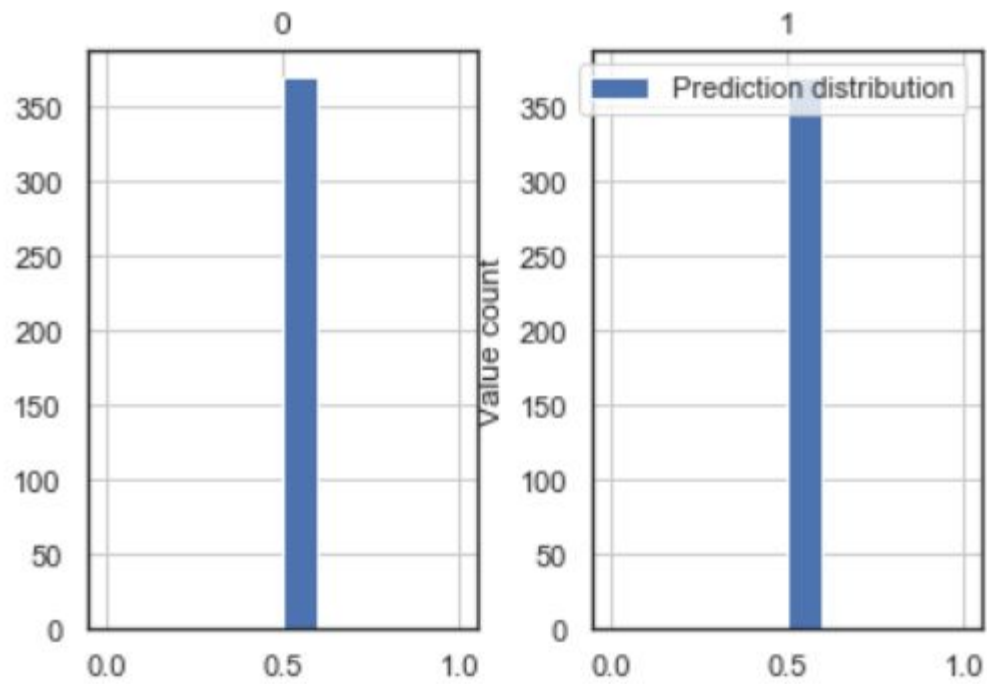


Overall term frequency

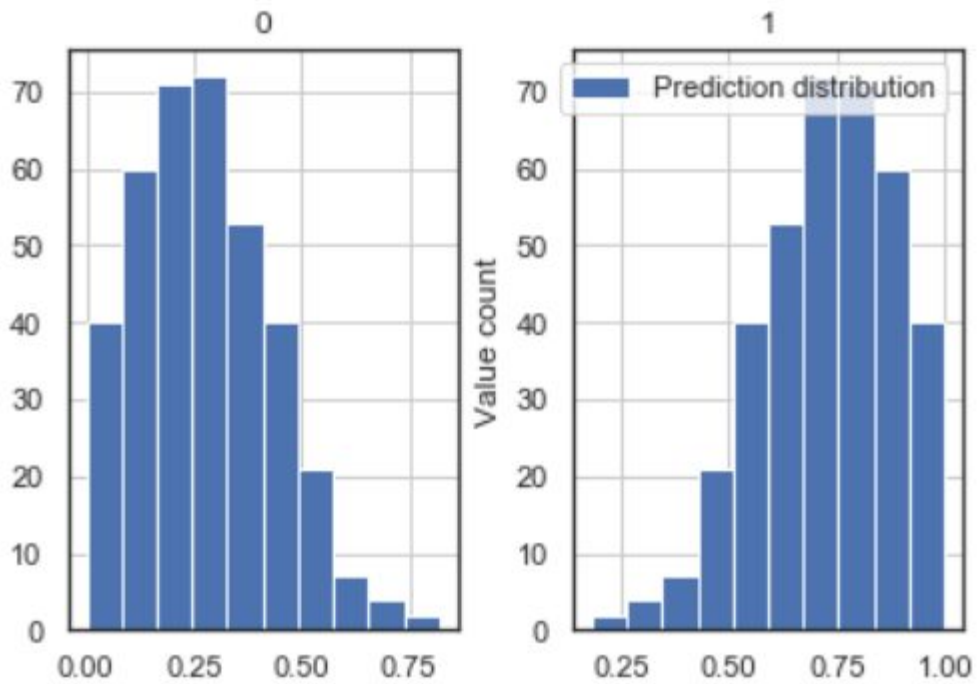
Estimated term frequency within the selected topic

1. $\text{saliency}(\text{term } w) = \text{frequency}(w) * [\sum_t p(t | w) * \log(p(t | w)/p(t))]$ for topics t ; see Chuang et. al
2. $\text{relevance}(\text{term } w | \text{topic } t) = \lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

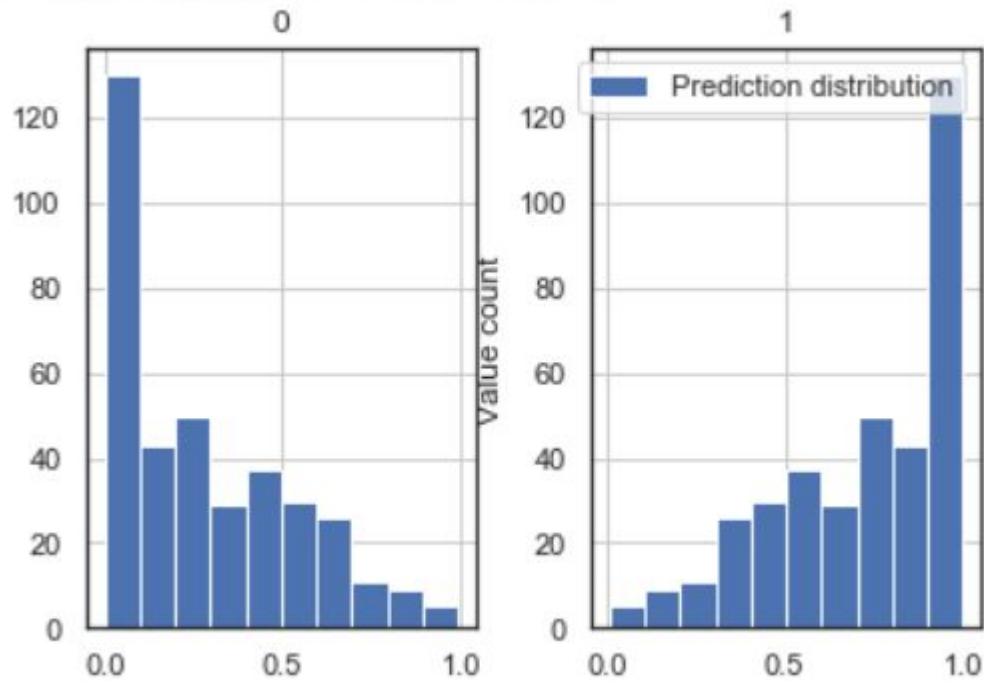
P9



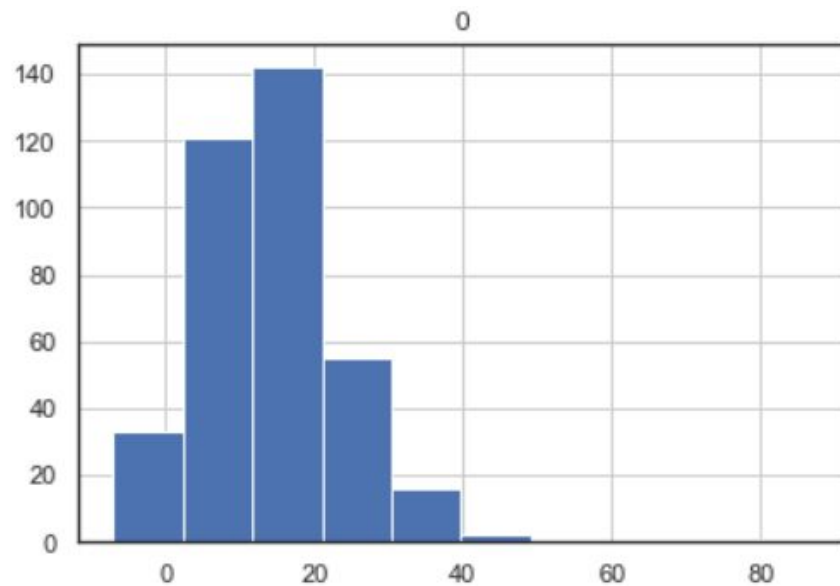
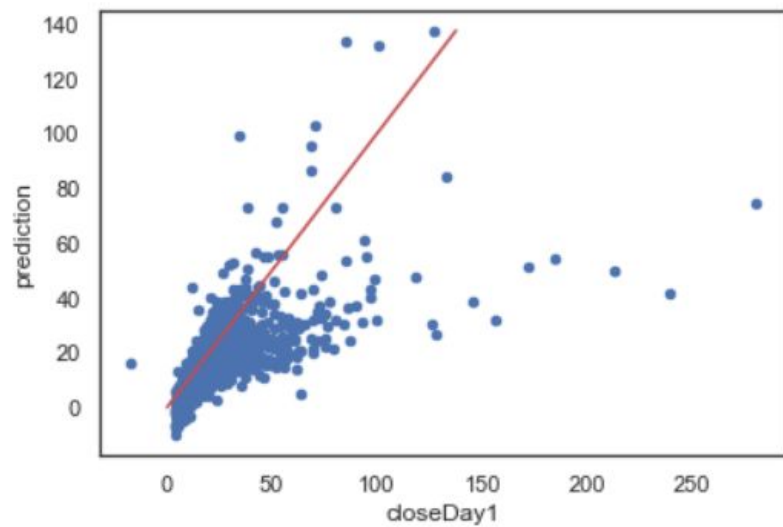
P8



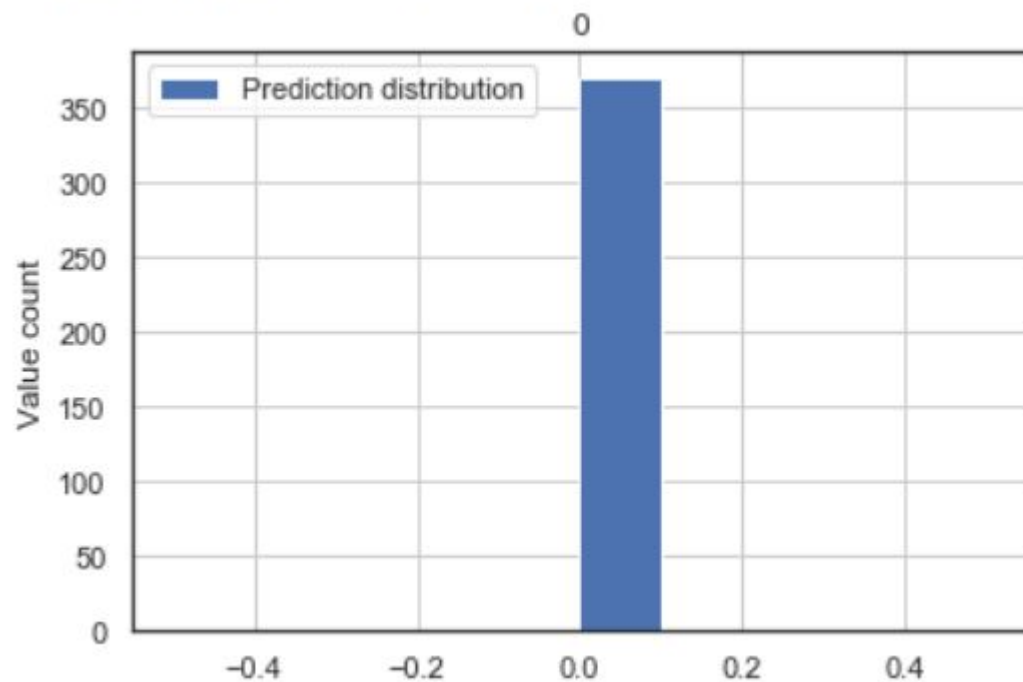
Using bagging classifier for P7 ...



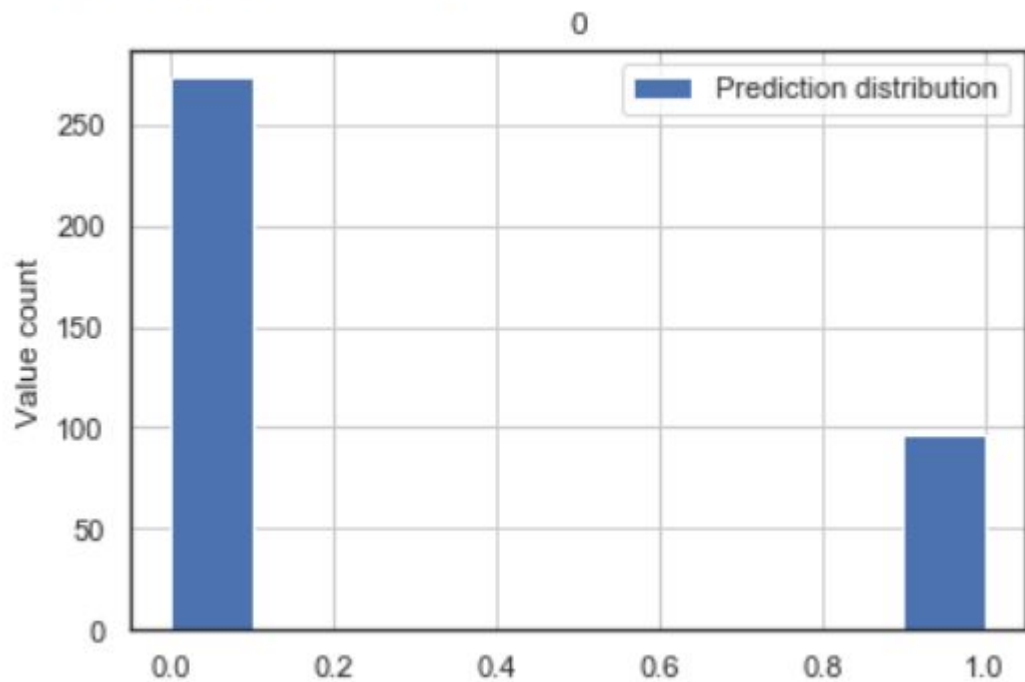
P6



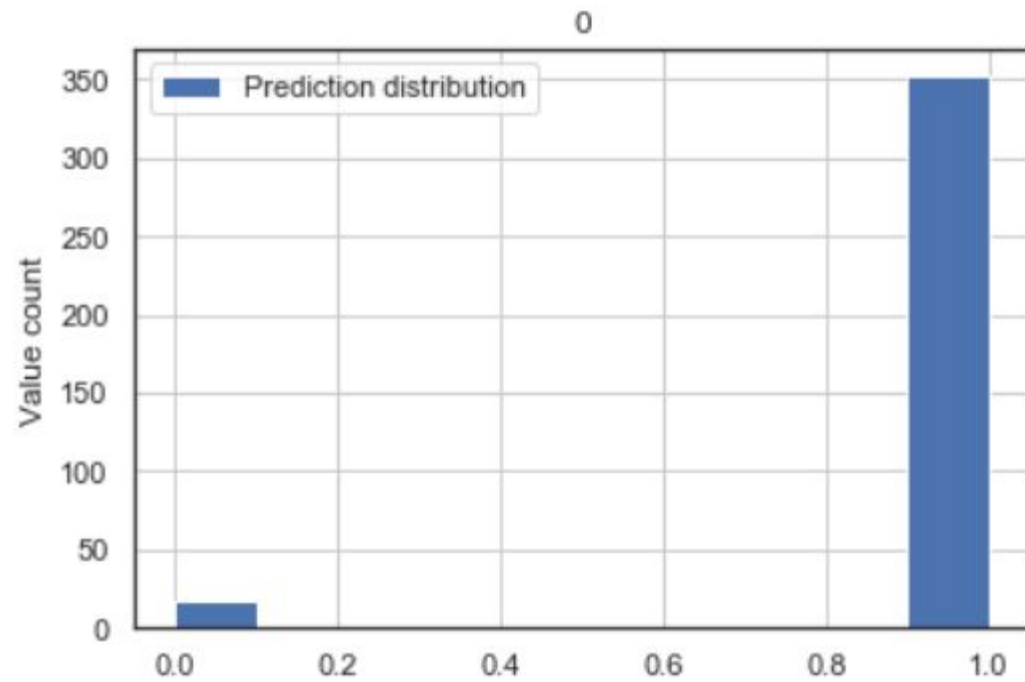
Using bagging classifier for P5 ...



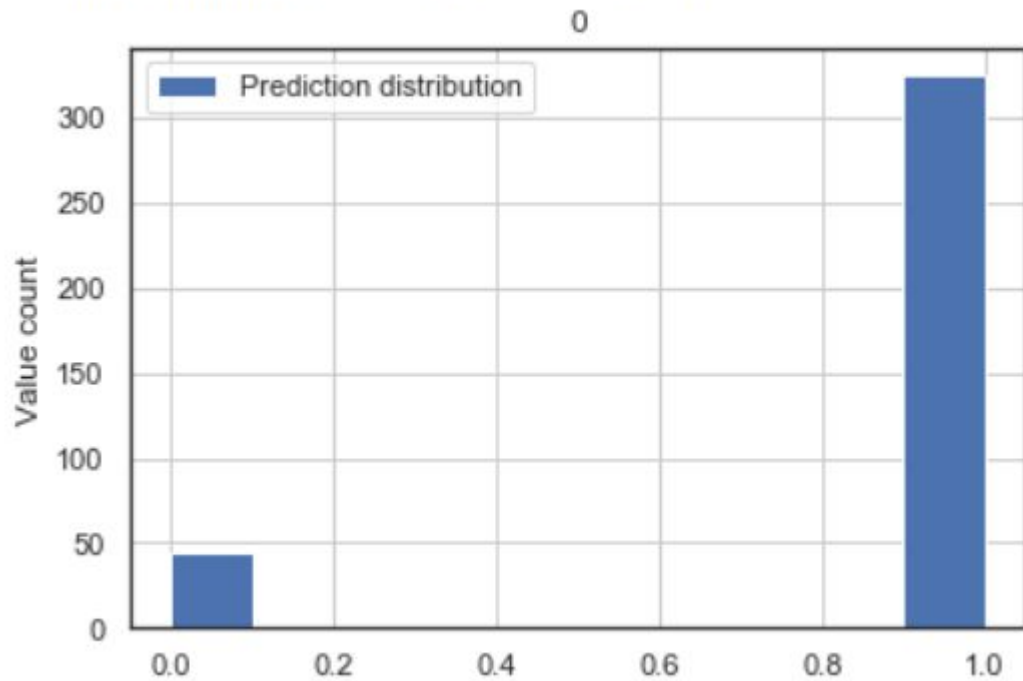
Using bagging classifier for P4 ...



P3



Using bagging classifier for P2 ...



Using bagging classifier for P1 ...

