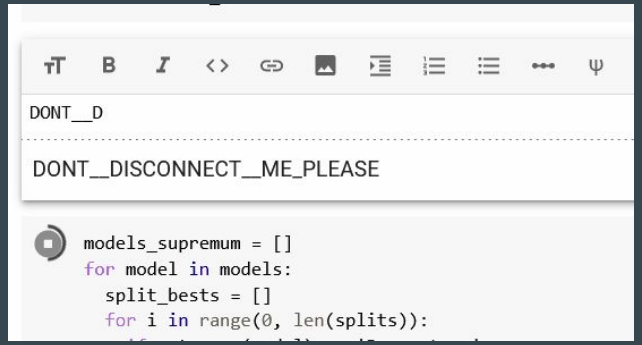


Man Vs. The Digital Oracle

•••

An attempt to predict belief in Zodiac signs with Machine Learning models using OkCupid data

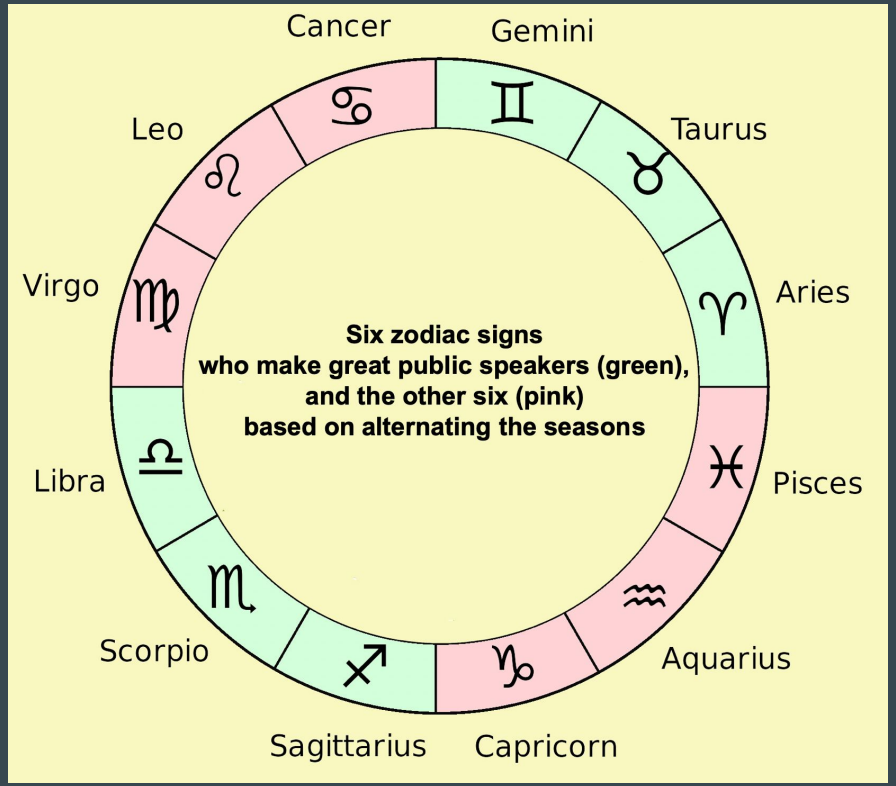
Mike Doan, Jack Ericson, Robert Tognoni

About Astrology/Horoscopes

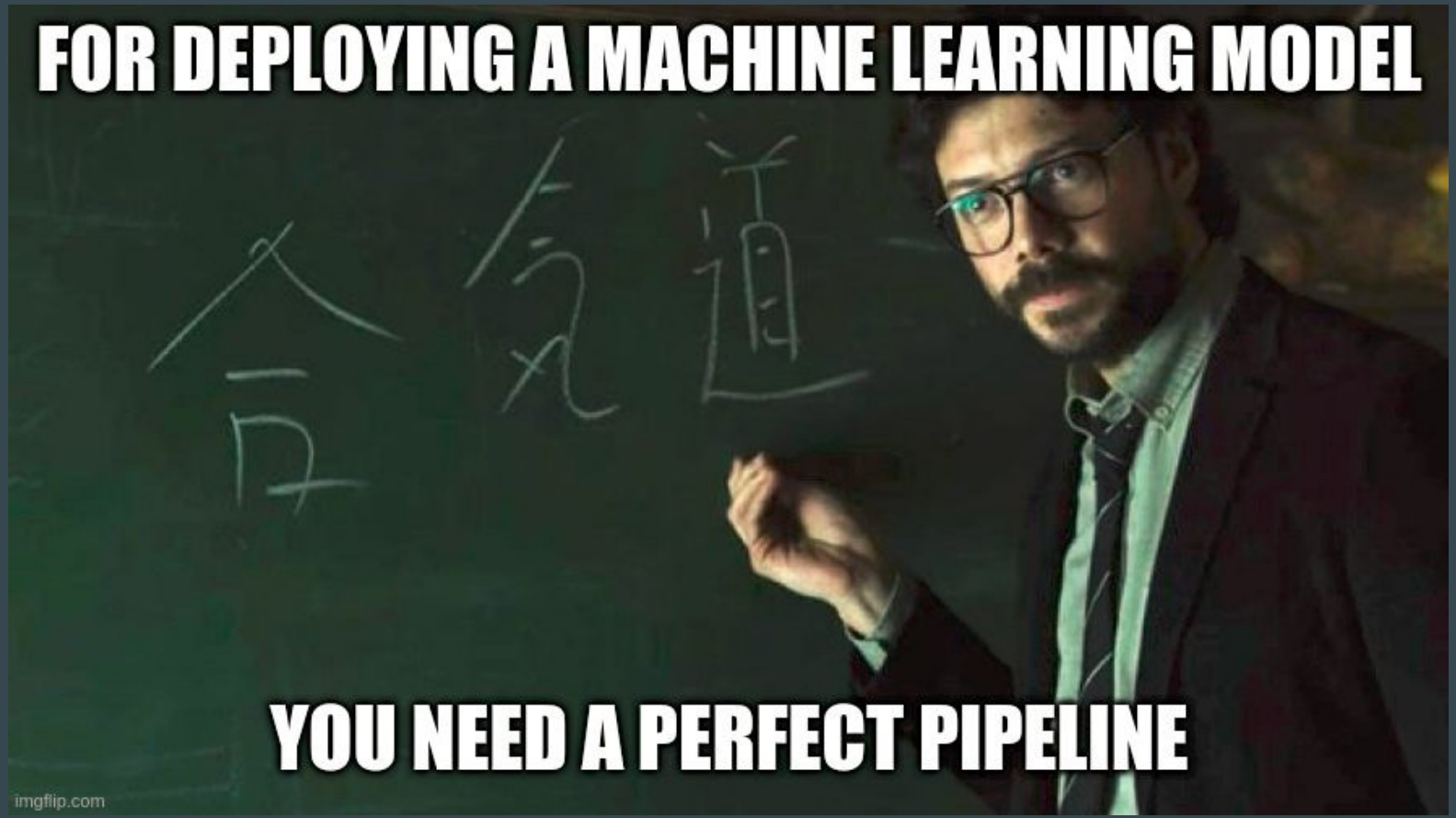
- “The study of the positions and motions of celestial bodies in the belief that they have an influence on the course of natural earthly occurrences and human affairs.” - *The first result on Google*
- An interesting result of the human pattern recognition applied to ‘random’ events.
- We intend to find out if a computer can do better.

Western ‘Astrology’ derived horoscopes

- Based on dominant star sign during birth month (12 possibilities)
- Deterministic personality traits
- Used to make predictions for love life, partner compatibility, finance *etc*
- Before PETA, better informed people would inspect entrails of slaughtered sheep
- Australians (and the rest of the Southern Hemisphere) are not real and, therefore, do not need Zodiac signs.



Section 1 - The Dataset



Raw Data

- 68k profiles, with 14k full (non-NaN) entries.
- The dataset comes from the (once) popular online dating platform **OKCupid**.
- The entries are **anonymous** and only contain information about the dating profiles themselves, not the interactions those profiles had on the platform.
- The overwhelming majority of the profiles in the dataset have their locations based in the state of **California**.
- This is bound to result in a highly biased dataset that is **not representative** of the global dating platform user base.

Pipeline

- For each model, we run them on **3 splits** of the entire data set (50/50), (70/30), (80/20)
- For every split, we do **10-fold cross validation** on all the models
- From the metrics from the CV, we choose the **best split**
- From the best splits, we choose the model that performs the best.
- From all the best tuned models, we choose the best overall classifier.

Pipeline specifics

- Due to different models requiring different metrics to evaluate, we have opted to **manually pick the best models** from the metrics instead of using an automatic tool such as GridSearchCV (ex: Linear Regression can only use RMSE)
- This specific pipeline that we have implemented forces each model to have the same metrics for every split. Not only did this simplify the implementation process, but also standardized results for ease of comparison between models.

Original Data fields

age
status
sex
orientation
body_type
diet
drinks
drugs
education
ethnicity
height
income
job
last_online
location
offspring
pets
religion
sign
smokes
speaks
essay0-9

- 31 raw data fields, 10 of which are user 'essays', leaving 21 standard categories including predictions: the Zodiac 'sign' field
- Some have modifiers such as 'Leo (is my sign), but I do not take it seriously' which will serve as the predictive labels of our models
- Most data is categorical. Exceptions: age (numeric), income (numeric), last_online (date), height (float)
- Most of this data is volunteered and assumed to be what they want seen, rather than truth.
- Users can withhold information in almost all fields if desired
- Some are not balanced, like location: Majority are from California

Sign Intensity As Prediction

- This is a modifier for the ‘sign’ attribute, which signifies how much a user believes in their star sign.
- Technically speaking, there are 4 possible options for this field: None, “doesn’t matter”, “it’s fun to think about” and “it matters a lot”
- We have decided to discard the None answers due to ambiguity (though they do have interesting correlations with some other ‘None fields’) and to combine “it’s fun to think about” with “it matters a lot”.
- This reduces our problem from a multi-class to a simple binary Yes/No

About the ‘Essays’

- Each of the 10 essays can were filled out based on user selected prompts
- Each row and each column can be a custom response any prompt
- Individuals chose to fill as many as they want or none
- Prompts are not provided, only the text entered

-This leaves the following options-

1. Discard all 10 fields
2. **Total word count**
3. # of prompts filled out
4. Letter count or length of response

Feature Selection

- Important features picked based on raw correlation with the sign intensity.
- Repeat information was filtered using general intuition about the features and dataset itself (e.g. drinks, drugs, and smokes strongly correlated)

	sign_intensity
sign_intensity	1
age	0.021602
status_data	0.001815
sex_data	0.145914
orientation_data	0.077667
body_type_data	0.050065
diet_data	0.02528
drinks_data	-0.011177
drugs_data	0.044252
education_data	-0.07444
height_data	-0.011825
sign_data	0.007643
income	-0.006411
job_data	-0.02206
offspring_data	0.013163
religion_data	-0.080336
religion_intensity	0.000466
speaks_data	0.012155
smokes_data	0.076484
essay_len	0.041466

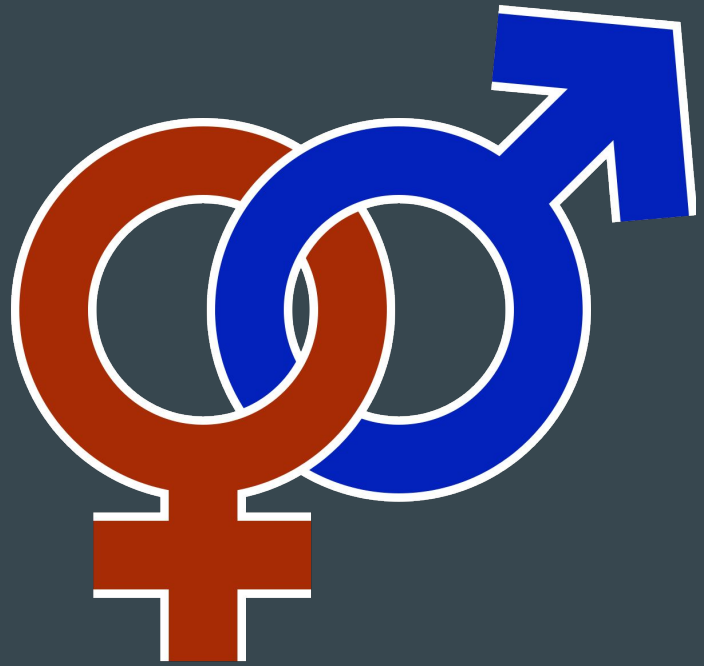
Correlation Data

Sex

- Unique Values: ['m' 'f'], which are encoded as [0,1]
- Correlation W/ Sign Intensity: 0.145
- High correlation with height, as well as some sizable correlations with other predictable metrics

"At this time our group has determined that it would best not to comment on this particular feature regardless of correlations noticed in dataset. "

- *The Team*



Education

- Unique Values Examples:
  - working on college/university'
  - 'working on space camp',
  - 'graduated from masters program',
  - 'graduated from college/university'
- Correlation W/ Sign Intensity: -0.07
- Grouped values into general level of education, such as primary, secondary, tertiary education.
- Relevancy: Higher educated people are assumed to be less likely to place weight on astrological signs.

Orientation

- Unique Vals
  - 'straight',
  - 'bisexual',
  - 'gay'
- Correlation W/ Sign Intensity: 0.077
- Relevancy -

“At this time our group has determined that it would best not to comment on this particular feature regardless of correlations noticed in dataset. “

- *The Team*

“No comment is the best comment”

Body Type

- Unique Vals- 'a little extra', 'average', 'thin', 'athletic', 'fit', 'skinny' 'curvy'
- Correlation W/ Sign Intensity - 0.05
- Presumed Relevancy- No intuition we could think of, but there has been some scientific inquiry into the relationship of weight and spirituality:

The Relationship Between Spirituality, Health-Related Behavior, and Psychological Well-Being - https://pubmed.ncbi.nlm.nih.gov/32922340/

Drugs

- Unique Values:
  - 'Never',
  - 'Sometimes',
  - 'Often'
- Correlation W/ Sign Intensity: 0.044
- People that do, or condone, psychoactive substances may be exposed to alternative mind states. Perhaps this could predispose them to take an interest in astrology or other spiritual activities.
- Regardless, the correlation with Sign Intensity is *higher* than most 😉.

“Never program sober” -Limus Torshalds, creator of Windows Penguin Edition

Age

- Unique Values: numeric 0-100
- Correlation W/ Sign Intensity: 0.021
- Could be relationship with old age or something to do with cultural generational changes.

“You should Descend the Gradient off my damn lawn, kid”

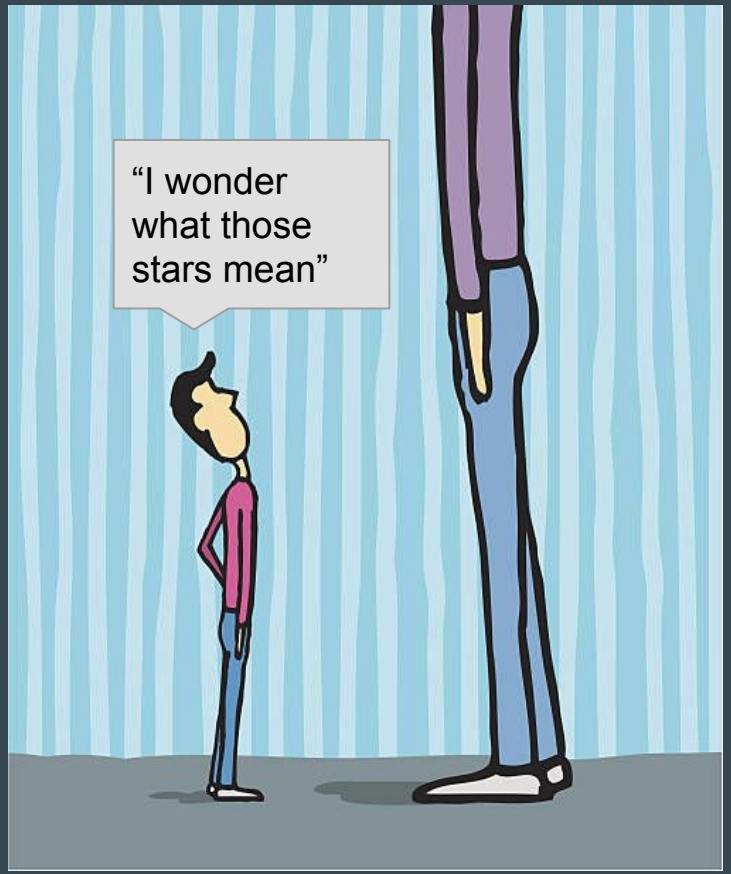
Offspring

- Unique Vals - Converted to 3 categories: 0 kids, 1 kid, >1 kid based on the prompts.
- Correlation W/ Sign intensity: 0.013
- Presumed Relevancy- It could be assumed that people without children trend more skeptical.

```
['none' "doesn't have kids" "doesn't have kids, but might want them"
 "doesn't have kids, but wants them" 'has kids'
 "doesn't have kids, and doesn't want any" "doesn't want kids"
 "has kids, but doesn't want more" 'has a kid' 'might want kids'
 "has a kid, but doesn't want more" 'has a kid, and wants more'
 'has a kid, and might want more' 'has kids, and might want more'
 'wants kids' 'has kids, and wants more']
```

Height

- Unique Values- Range between 1.0 and 100.0 (inches)
- Correlation W/ Sign Intensity: -0.012
- Presumed Relevance - The correlation is low, but maybe shorter people spend more time looking up at the sky?



Last_Online

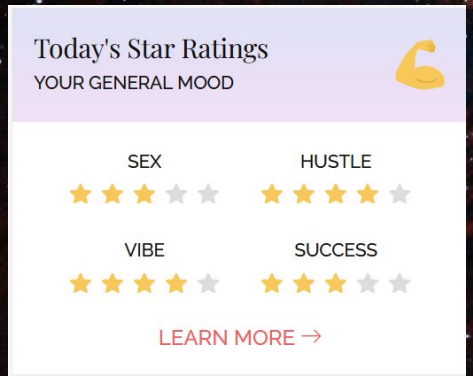
- Unique Vals- Date of last log on.
- Correlation W/ Sign - 0.034
- Presumed Relevancy- The more active a member is the more likely they are to believe in astrological concepts?

Religion

- Unique Vals-
  - 'atheism' 'christianity' 'catholicism' 'agnosticism' 'buddhism' 'other' 'hinduism' 'judaism' 'islam'
- Separated Religious Intensity from -
  - laughing about it' ‘very serious about it' 'not too serious' ‘somewhat serious’
- Correlation W/ Sign Intensity: -0.080336
- Presumed Relevancy- This relates through the idea of spirituality directly

Section 2 - About The Models

I predict that humans born under Aries shall have good vibes, but will not succeed in mating.



The Zodiac Perceptron Model

List of models

-Decision Tree

-Perceptron

-Multi-layer Perceptron

-Naive Bayes

-Logistic Regression

-Linear Regression

-SVM w/ Linear Kernel

-SVM w/ RBF Kernel

-Gradient Boosting

For this presentation and for the sake of runtime each student took ownership of 3 of 9 required models. The following is our metrics and notes for each

Decision Tree

- Overview: Performs many ‘yes/no’ conditional statements to predict labels.
- Parameters:
  - max_depth = 5 - any number higher resulted in models that easily overfit, and number lower resulted in models that did not fit to training data at all
  - criterion = ‘entropy’
- Features used: 'sex_data', 'orientation_data', 'drinks_data',
- 'education_data', 'body_type_data', 'last_online_data',
- 'smokes_data', 'height', 'religion_data'

Metrics for Decision Tree

- Most models seem to perform around the 60% mark for all performance metrics
- Off the splits, the best models according to the metrics are:

  model 0 from split 0

  model 3 from split 1

  model 8 from split 2

Decision Tree Metrics for 10-fold on split 0

Fold	Accuracy	Precision	Recall	F1
0	0.634	0.636	0.634	0.634
1	0.585	0.586	0.585	0.586
2	0.609	0.609	0.609	0.609
3	0.617	0.617	0.617	0.617
4	0.611	0.611	0.611	0.611
5	0.586	0.587	0.586	0.586
6	0.594	0.596	0.594	0.595
7	0.574	0.579	0.574	0.573
8	0.596	0.597	0.596	0.596
9	0.597	0.599	0.597	0.598

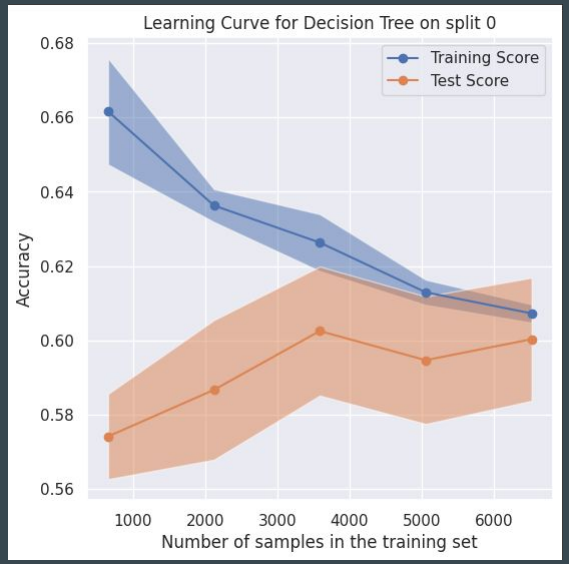
Decision Tree Metrics for 10-fold on split 1

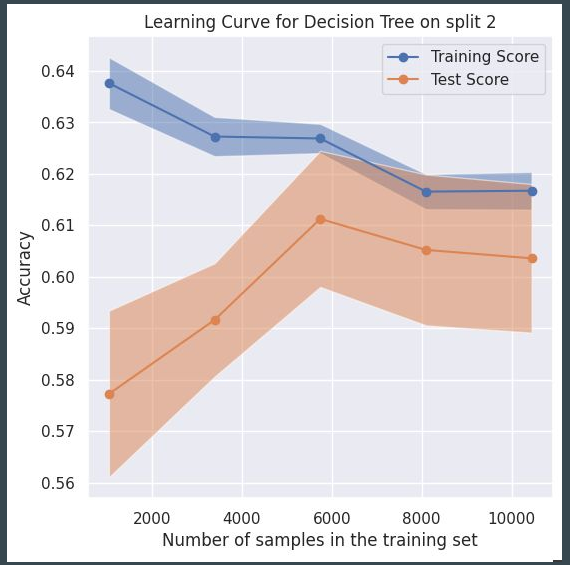
Fold	Accuracy	Precision	Recall	F1
0	0.612	0.612	0.612	0.612
1	0.584	0.583	0.584	0.582
2	0.58	0.582	0.58	0.58
3	0.632	0.632	0.632	0.632
4	0.599	0.599	0.599	0.599
5	0.629	0.628	0.629	0.628
6	0.614	0.614	0.614	0.609
7	0.594	0.6	0.594	0.593
8	0.583	0.584	0.583	0.583
9	0.622	0.621	0.622	0.621

Decision Tree Metrics for 10-fold on split 2

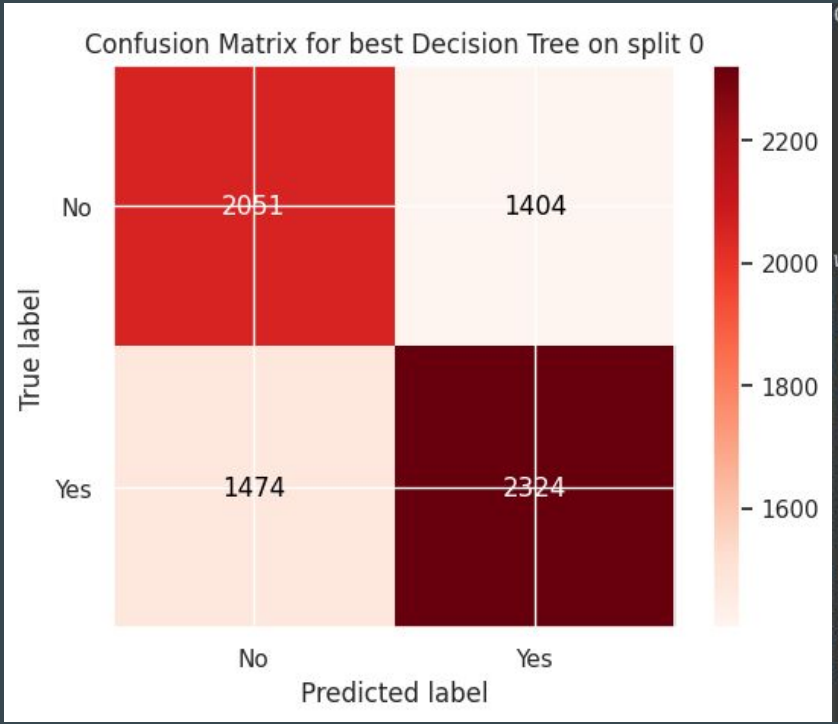
Fold	Accuracy	Precision	Recall	F1
0	0.585	0.587	0.585	0.585
1	0.593	0.592	0.593	0.592
2	0.585	0.586	0.585	0.585
3	0.602	0.602	0.602	0.602
4	0.599	0.598	0.599	0.598
5	0.625	0.624	0.625	0.624
6	0.609	0.608	0.609	0.607
7	0.595	0.594	0.595	0.594
8	0.627	0.626	0.627	0.626
9	0.616	0.616	0.616	0.615

Decision Tree Metrics for Splits





Decision Tree Best Model Confusion Matrix - Split 0 (0)



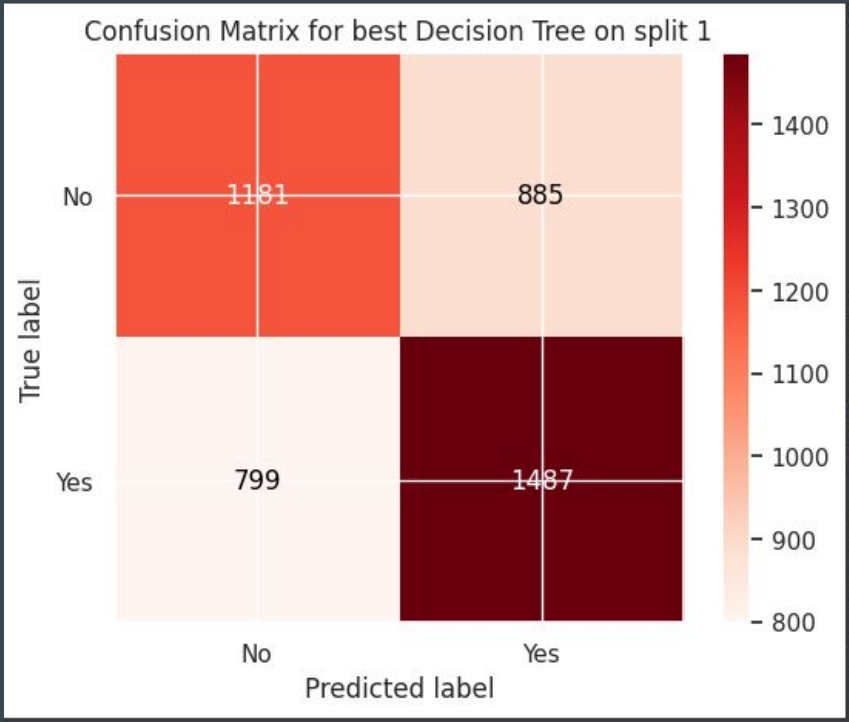
Classification report for best Decision Tree on unseen data on split 0

	precision	recall	f1-score	support
0	0.58	0.59	0.59	3455
1	0.62	0.61	0.62	3798
accuracy			0.60	7253
macro avg	0.60	0.60	0.60	7253
weighted avg	0.60	0.60	0.60	7253

Decision Tree Metrics for 10-fold on split 0

Fold	Accuracy	Precision	Recall	F1
0	0.634	0.636	0.634	0.634
1	0.585	0.586	0.585	0.586
2	0.609	0.609	0.609	0.609
3	0.617	0.617	0.617	0.617
4	0.611	0.611	0.611	0.611
5	0.586	0.587	0.586	0.586
6	0.594	0.596	0.594	0.595
7	0.574	0.579	0.574	0.573
8	0.596	0.597	0.596	0.596
9	0.597	0.599	0.597	0.598

Decision Tree Best Model Confusion Matrix - Split 1 (3)



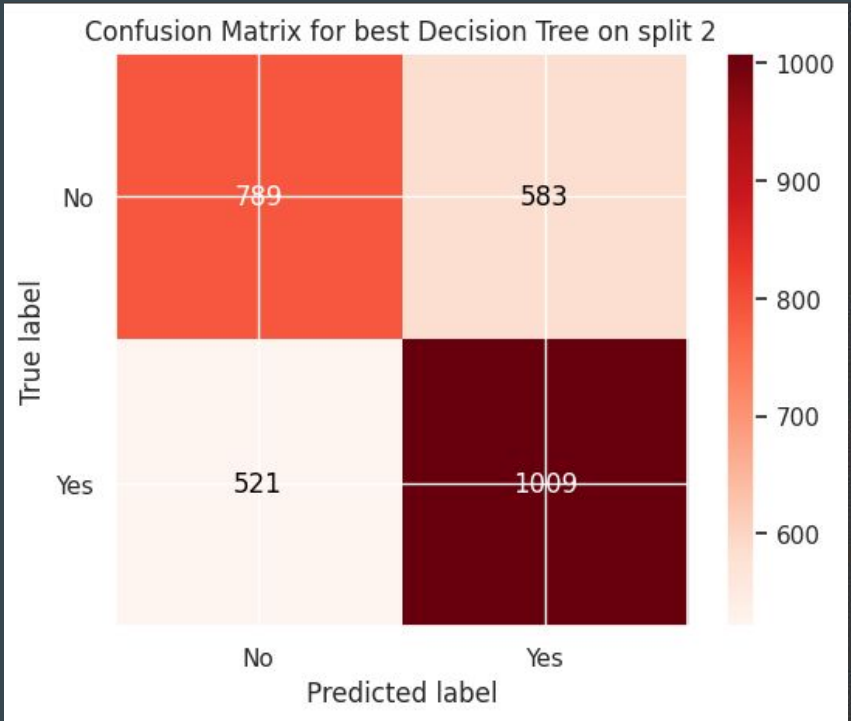
Classification report for best Decision Tree on unseen data on split 1

	precision	recall	f1-score	support
0	0.60	0.57	0.58	2066
1	0.63	0.65	0.64	2286
accuracy			0.61	4352
macro avg	0.61	0.61	0.61	4352
weighted avg	0.61	0.61	0.61	4352

Decision Tree Metrics for 10-fold on split 1

Fold	Accuracy	Precision	Recall	F1
0	0.612	0.612	0.612	0.612
1	0.584	0.583	0.584	0.582
2	0.58	0.582	0.58	0.58
3	0.632	0.632	0.632	0.632
4	0.599	0.599	0.599	0.599
5	0.629	0.628	0.629	0.628
6	0.614	0.614	0.614	0.609
7	0.594	0.6	0.594	0.593
8	0.583	0.584	0.583	0.583
9	0.622	0.621	0.622	0.621

Decision Tree Best Model Confusion Matrix - Split 2 (8)



Classification report for best Decision Tree on unseen data on split 2

	precision	recall	f1-score	support
0	0.60	0.58	0.59	1372
1	0.63	0.66	0.65	1530
accuracy			0.62	2902
macro avg	0.62	0.62	0.62	2902
weighted avg	0.62	0.62	0.62	2902

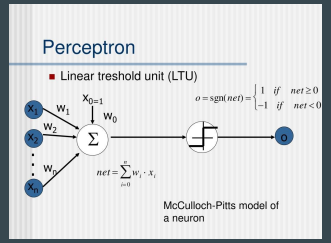
Decision Tree Metrics for 10-fold on split 2

Fold	Accuracy	Precision	Recall	F1
0	0.585	0.587	0.585	0.585
1	0.593	0.592	0.593	0.592
2	0.585	0.586	0.585	0.585
3	0.602	0.602	0.602	0.602
4	0.599	0.598	0.599	0.598
5	0.625	0.624	0.625	0.624
6	0.609	0.608	0.609	0.607
7	0.595	0.594	0.595	0.594
8	0.627	0.626	0.627	0.626
9	0.616	0.616	0.616	0.615

Decision Tree Model Selection

- From the best models from the splits, the best model seems to be the model trained on split 2 (80/20)
- Predictably, training accuracy on every fold as well as on the validation sets go down as more examples are added to the training set (decision tree sees more contradictory samples, resulting in a lot of noise, but also a better generalizing model on unseen data)

Perceptron



- Overview - A fairly simple linear classifier
- Parameters - setting fit_intercept to False resulted in a very slight overall improvement. Otherwise, defaults were about the best possible options
- Major improvement reducing feature list to just high correlation items
- Predicts - Poorly. The model does a particular;y horrible with class ‘doesn’t matter’
- Adjustment in dataset: Reducing features to just those with the highest correlation to the target class (>|0.04|) resulted in a significant improvement.

Model: Perceptron at split: split_0.5/0.5

	precision	recall	f1-score	support
It Matters	0.53	0.99	0.69	4559
Doesn't Matter	0.54	0.01	0.02	4131
accuracy			0.53	8690
macro avg	0.53	0.50	0.35	8690
weighted avg	0.53	0.53	0.37	8690

is a perceptron or svc, it likes less features

Perceptron attempting to classify with max features available (very bad)

Perceptron Metrics for Splits (all not great)

Chosen by best overall metrics though LC breadth is very wide

****

Model: Perceptron at split: split_0.5/0.5

	precision	recall	f1-score	support
Doesn't Matter	0.55	0.91	0.68	4559
It Matters	0.62	0.17	0.27	4131
accuracy			0.56	8690
macro avg	0.58	0.54	0.47	8690
weighted avg	0.58	0.56	0.48	8690

WTF

is a perceptron or svc, it likes less features

****

Model: Perceptron at split: split_0.7/0.3

	precision	recall	f1-score	support
Doesn't Matter	0.61	0.68	0.64	2735
It Matters	0.59	0.52	0.56	2479
accuracy			0.60	5214
macro avg	0.60	0.60	0.60	5214
weighted avg	0.60	0.60	0.60	5214

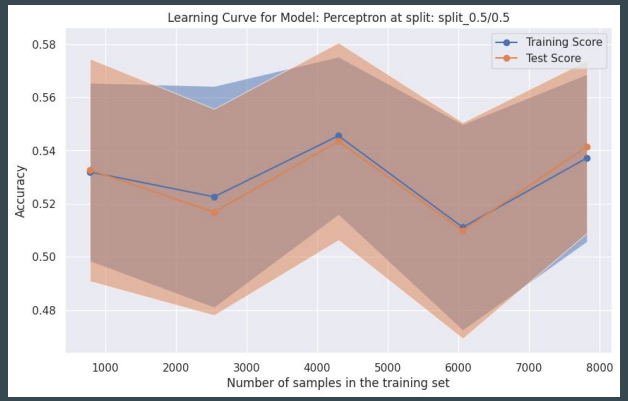
is a perceptron or svc, it likes less features

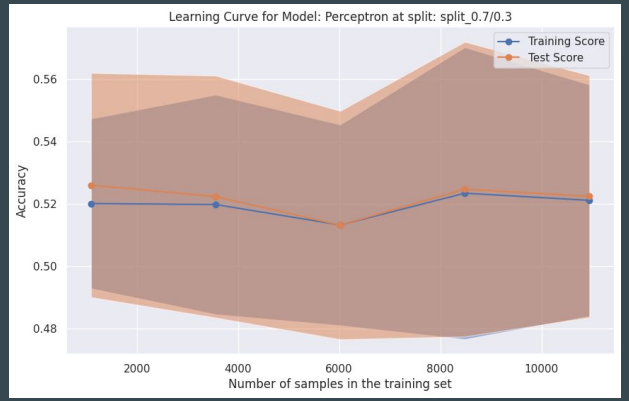
****

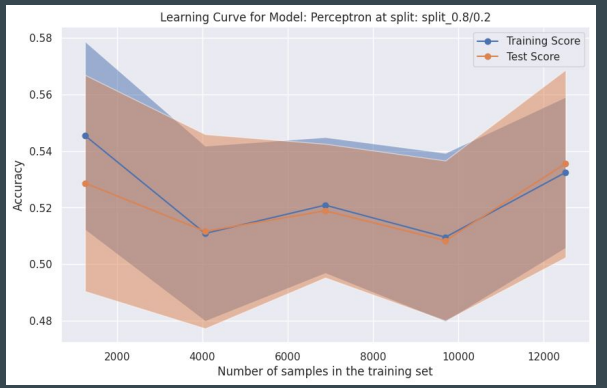
Model: Perceptron at split: split_0.8/0.2

	precision	recall	f1-score	support
Doesn't Matter	0.59	0.60	0.59	1846
It Matters	0.54	0.52	0.53	1630
accuracy			0.56	3476
macro avg	0.56	0.56	0.56	3476
weighted avg	0.56	0.56	0.56	3476

****





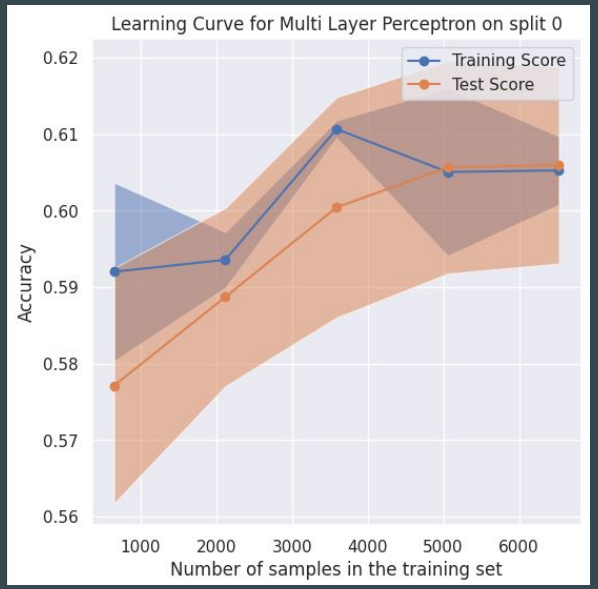


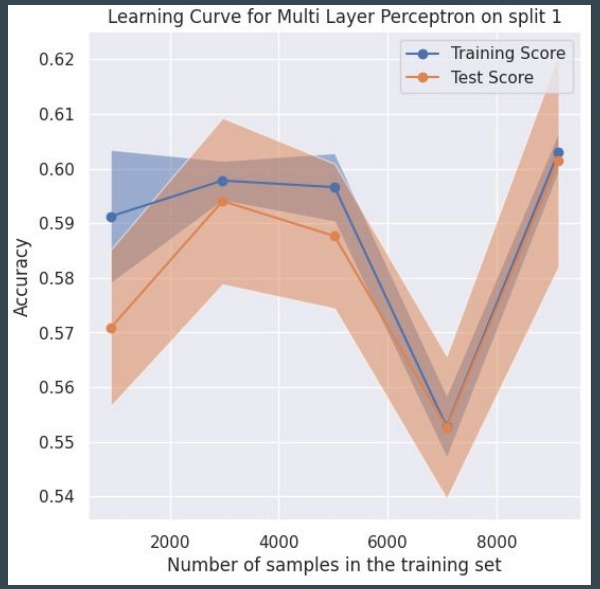
Multi-Layer Perceptron

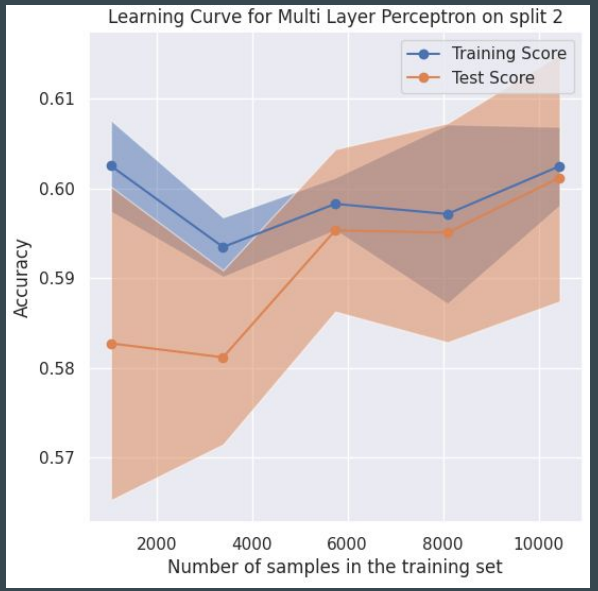
- Overview:
- Parameters: hidden_layer_sizes = (100,),
- shuffle = False,
- learning_rate = 'adaptive',
- Features used: 'sex_data', 'orientation_data', 'height', 'religion_data', 'religion_intensity'

- Predicts: Among one of the better models, performs significantly better than perceptron does but only when features are limited to the highest correlating features.

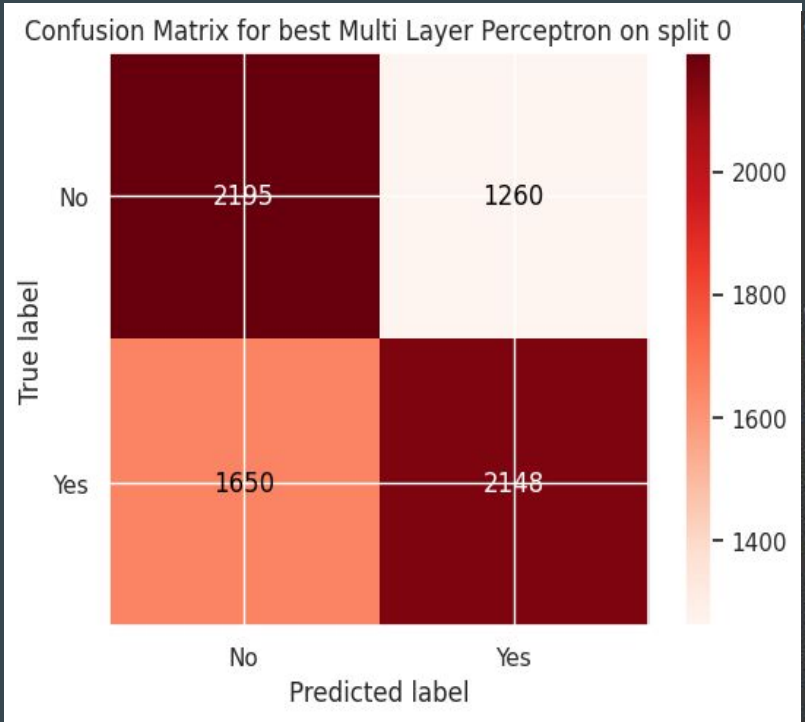
MLP Metrics for Splits







MLP Best Model Metrics on Split 0 (0)



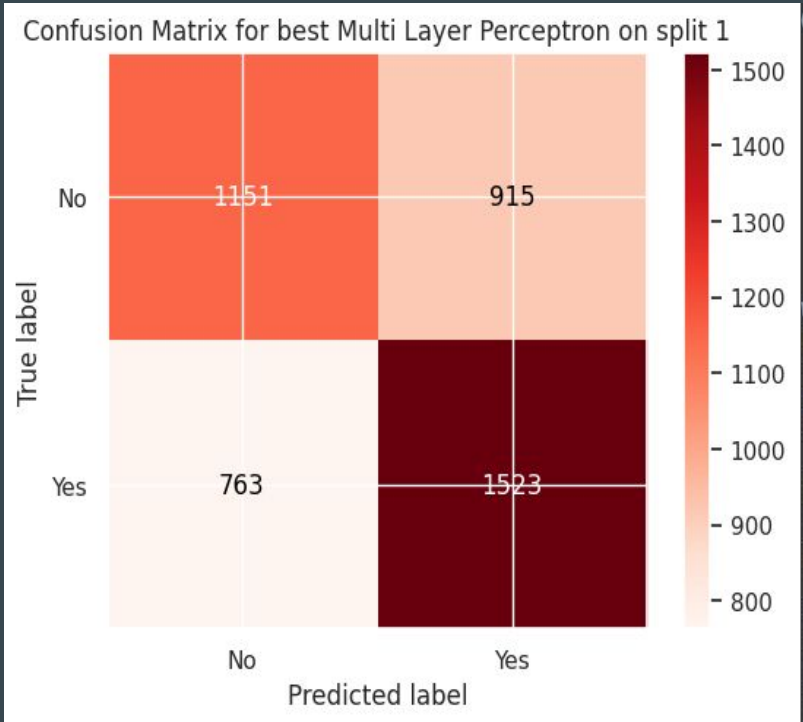
Classification report for best Multi Layer Perceptron on unseen data on split 0

	precision	recall	f1-score	support
0	0.57	0.64	0.60	3455
1	0.63	0.57	0.60	3798
accuracy			0.60	7253
macro avg	0.60	0.60	0.60	7253
weighted avg	0.60	0.60	0.60	7253

Multi Layer Perceptron Metrics for 10-fold on split 0

Fold	Accuracy	Precision	Recall	F1
0	0.628	0.633	0.628	0.628
1	0.596	0.595	0.596	0.594
2	0.605	0.604	0.605	0.602
3	0.625	0.624	0.625	0.622
4	0.606	0.608	0.606	0.596
5	0.611	0.611	0.611	0.607
6	0.603	0.605	0.603	0.603
7	0.606	0.605	0.606	0.603
8	0.604	0.603	0.604	0.603
9	0.586	0.586	0.586	0.586

MLP Best Model Metrics on Split 1 (4)



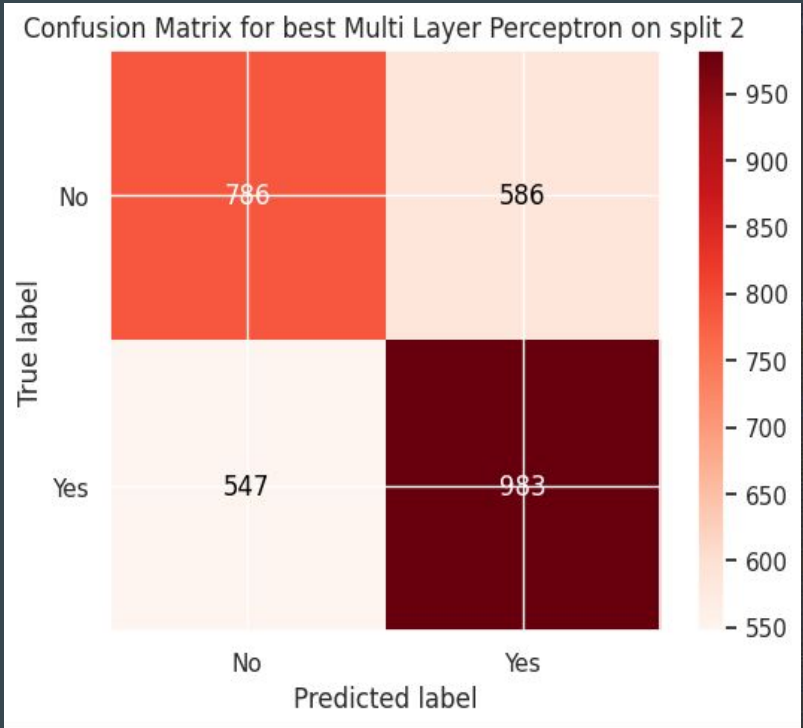
Classification report for best Multi Layer Perceptron on unseen data on split 1

	precision	recall	f1-score	support
0	0.60	0.56	0.58	2066
1	0.62	0.67	0.64	2286
accuracy			0.61	4352
macro avg	0.61	0.61	0.61	4352
weighted avg	0.61	0.61	0.61	4352

Multi Layer Perceptron Metrics for 10-fold on split 1

Fold	Accuracy	Precision	Recall	F1
0	0.598	0.599	0.598	0.592
1	0.584	0.583	0.584	0.582
2	0.582	0.581	0.582	0.577
3	0.606	0.606	0.606	0.603
4	0.624	0.623	0.624	0.621
5	0.636	0.636	0.636	0.635
6	0.598	0.598	0.598	0.593
7	0.605	0.605	0.605	0.605
8	0.567	0.566	0.567	0.563
9	0.612	0.612	0.612	0.612

MLP Best Model Metrics on Split 2 (8)



Classification report for best Multi Layer Perceptron on unseen data on split 2

	precision	recall	f1-score	support
0	0.59	0.57	0.58	1372
1	0.63	0.64	0.63	1530
accuracy			0.61	2902
macro avg	0.61	0.61	0.61	2902
weighted avg	0.61	0.61	0.61	2902

Multi Layer Perceptron Metrics for 10-fold on split 2

Fold	Accuracy	Precision	Recall	F1
0	0.603	0.602	0.603	0.602
1	0.596	0.596	0.596	0.596
2	0.599	0.601	0.599	0.6
3	0.607	0.607	0.607	0.607
4	0.563	0.576	0.563	0.555
5	0.619	0.619	0.619	0.619
6	0.606	0.605	0.606	0.604
7	0.592	0.591	0.592	0.59
8	0.619	0.618	0.619	0.618
9	0.614	0.615	0.614	0.614

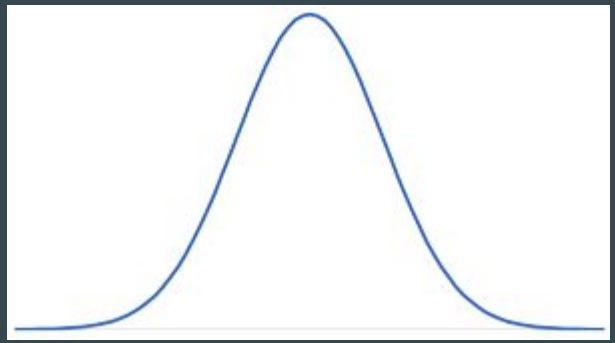
Multi Layer Perceptron Model Selection

- Models trained on split 1 and 2 perform similarly, with negligible difference in precision for class 1 and 0.
- The difference between the two shows up mostly in generalization error. The model trained on split 2 has lower generalization error owing to the fact that all the folds performed better on the validation set.
- -> I believe that the MLP trained on fold 8 of split 2 is the best model

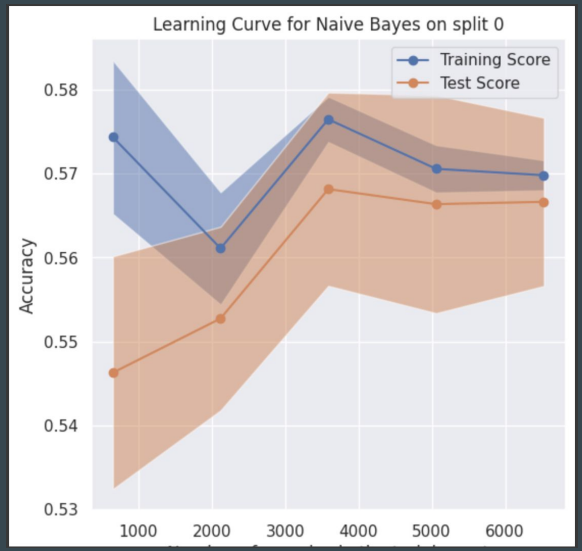
Gaussian Naive Bayes (GNB)

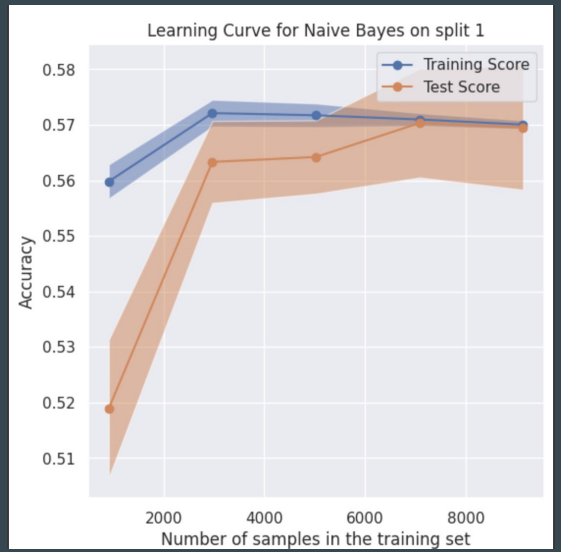
- Classifies data using bayesian statistics. Using prior assumptions, the classifier conditionally predicts outputs. It is most valuable in the analysis of large datasets with high dimensionality.

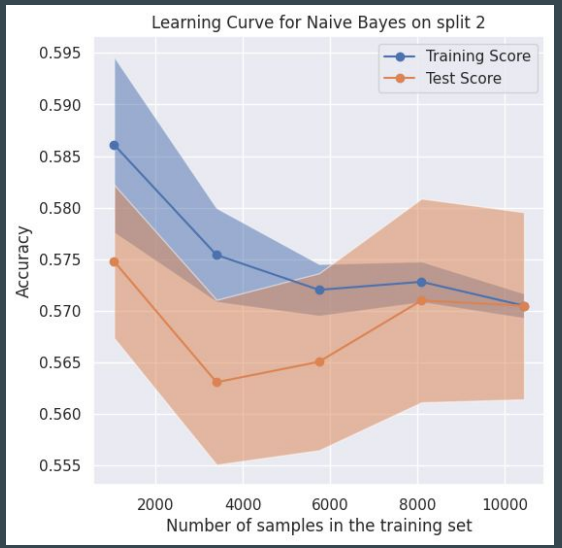
$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$



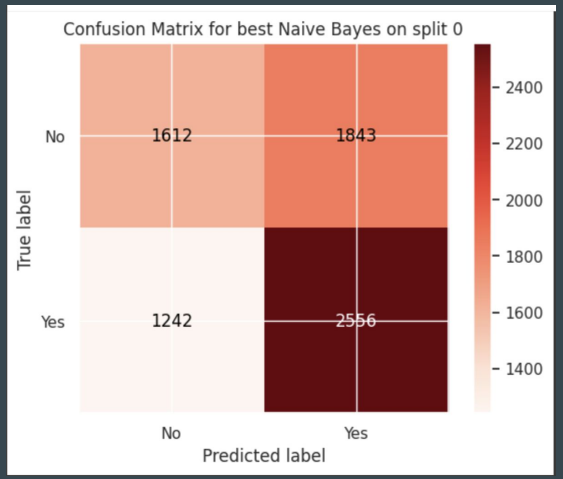
GNB Metrics for Splits

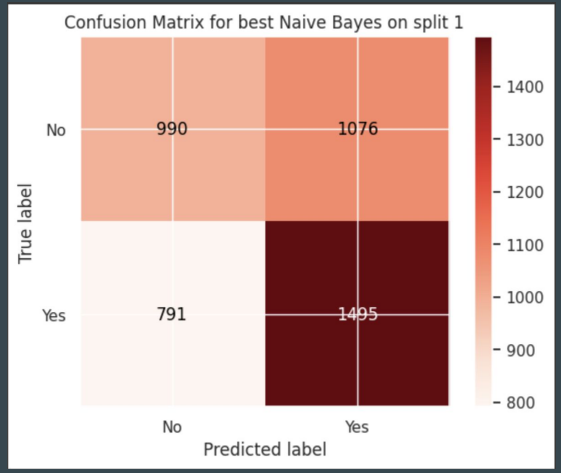


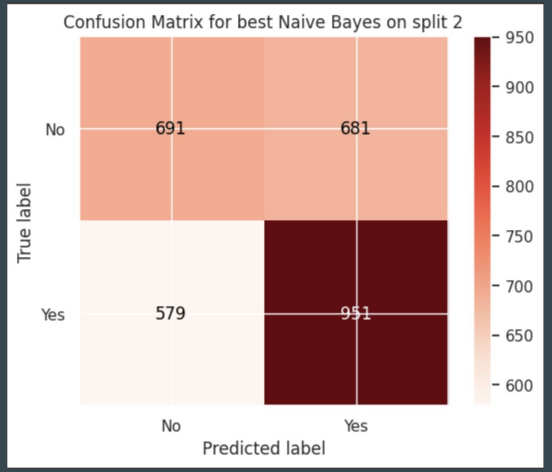




GNB Best Model Metrics







Classification report for best Naive Bayes on unseen data on split 0 50/50

	precision	recall	f1-score	support
0	0.56	0.47	0.51	3455
1	0.58	0.67	0.62	3798
accuracy			0.57	7253
macro avg	0.57	0.57	0.57	7253
weighted avg	0.57	0.57	0.57	7253

Classification report for best Naive Bayes on unseen data on split 1 70/30

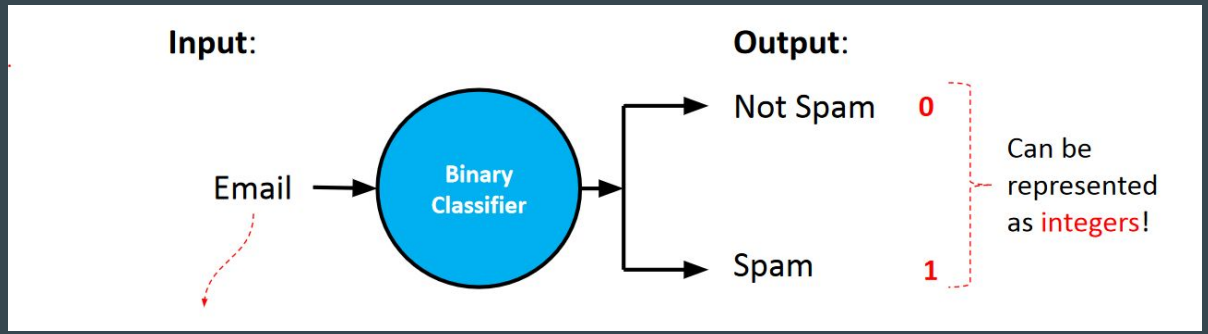
	precision	recall	f1-score	support
0	0.56	0.48	0.51	2066
1	0.58	0.65	0.62	2286
accuracy			0.57	4352
macro avg	0.57	0.57	0.57	4352
weighted avg	0.57	0.57	0.57	4352

Classification report for best Naive Bayes on unseen data on split 2 80/20

	precision	recall	f1-score	support
0	0.54	0.50	0.52	1372
1	0.58	0.62	0.60	1530
accuracy			0.57	2902
macro avg	0.56	0.56	0.56	2902
weighted avg	0.56	0.57	0.56	2902

Logistic Regression

- Overview
- Parameters - While minor, 'Solver' had the most impact (see next slide)
  - no change with # iterations >50.
  - Adjusting regularization strength C had very minor effects, default was sufficient.
- Predicts - Slightly improved performance over Perceptron in both classes.



Logistic Regression - Comparing 'solvers'

=> 'lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga'

-Each of the above is an algorithm used in minimizing the cost function

- Compared all solvers, not much of a difference but the newton methods did technically better.

- The Scikitlearn docs note "'newton-cholesky' is a good choice for n_samples >> n_features", which is the case for this dataset.

	Model Name	Accuracy	Precision	Recall	F1-score	RMSE
0	lbfgs	0.615362	0.613393	0.612705	0.612832	0.620192
1	liblinear	0.616226	0.61428	0.613625	0.613755	0.619495
2	newton-cholesky	0.616513	0.614565	0.613896	0.614027	0.619263
3	newton-cg	0.616513	0.614565	0.613896	0.614027	0.619263
4	sag	0.610088	0.608895	0.60655	0.606002	0.624429
5	saga	0.608554	0.607324	0.605012	0.604451	0.625657

Metrics of the 6 solver options cross validated with the same settings.

https://stackoverflow.com/questions/38640109/logistic-regression-python-solvers-definitions (a great write-up about this)

LogReg Metrics for Splits

Chose 80/20. Better average overall for Class 'It Matters'
Learning Curve appears to be less variable than 70/30

****

Model: LogisticRegression at split: split_0.5/0.5

	precision	recall	f1-score	support
Doesn't Matter	0.61	0.67	0.64	4559
It Matters	0.59	0.53	0.56	4131
accuracy			0.60	8690
macro avg	0.60	0.60	0.60	8690
weighted avg	0.60	0.60	0.60	8690

****

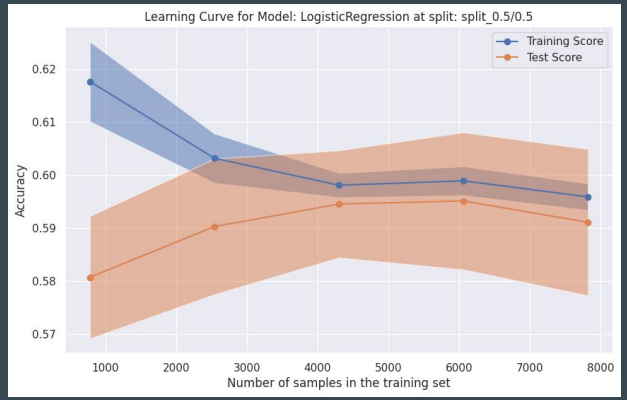
Model: LogisticRegression at split: split_0.7/0.3

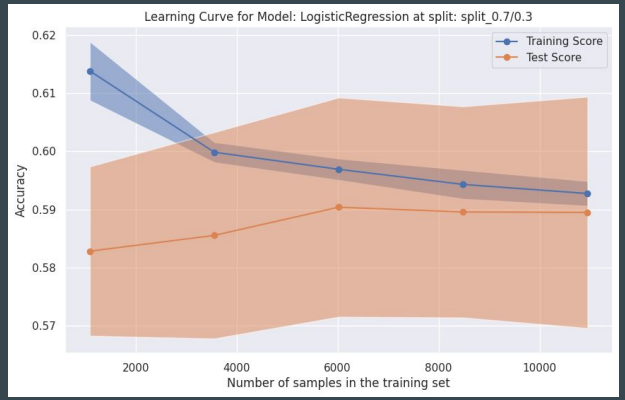
	precision	recall	f1-score	support
Doesn't Matter	0.62	0.70	0.66	2735
It Matters	0.61	0.53	0.57	2479
accuracy			0.62	5214
macro avg	0.62	0.61	0.61	5214
weighted avg	0.62	0.62	0.61	5214

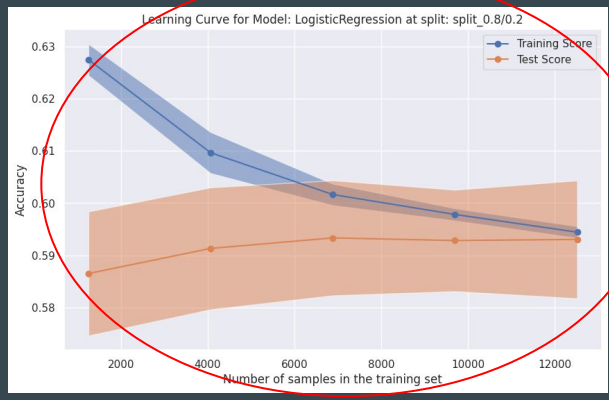
****

Model: LogisticRegression at split: split_0.8/0.2

	precision	recall	f1-score	support
Doesn't Matter	0.63	0.66	0.65	1846
It Matters	0.59	0.57	0.58	1630
accuracy			0.62	3476
macro avg	0.61	0.61	0.61	3476
weighted avg	0.62	0.62	0.62	3476

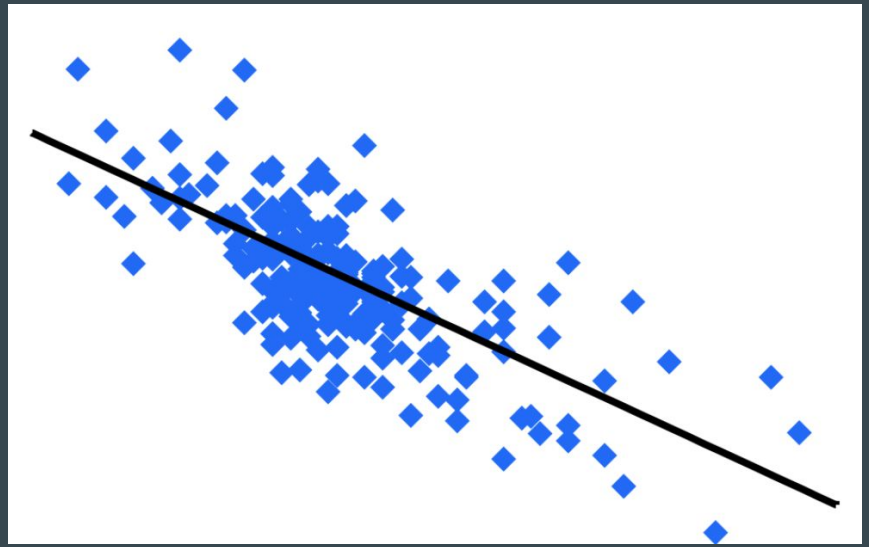




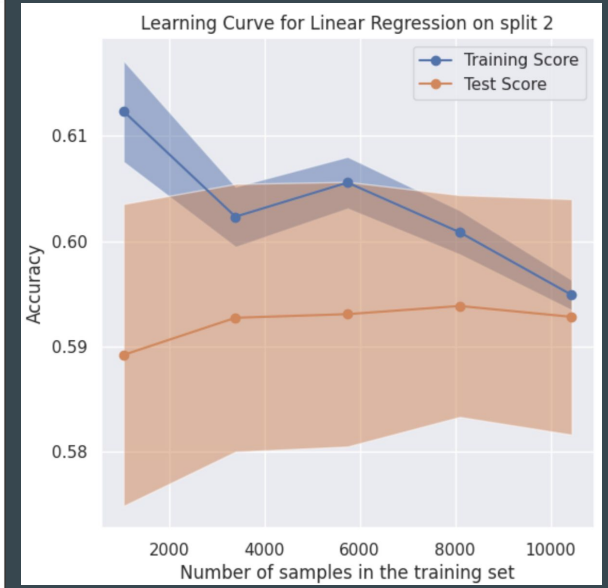


Linear Regression - Ridge Regression

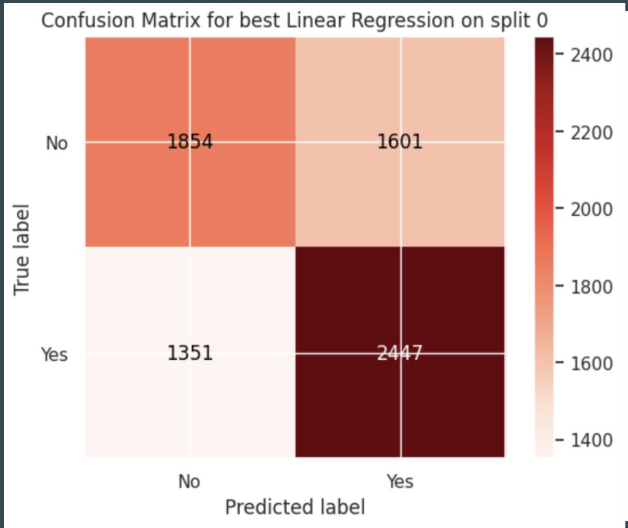
- Linear regression models are designed for continuous output prediction.
- In our case, where we have two classifications, a linear regression model will not predict the outputs we need.
- Used Ridge Regression

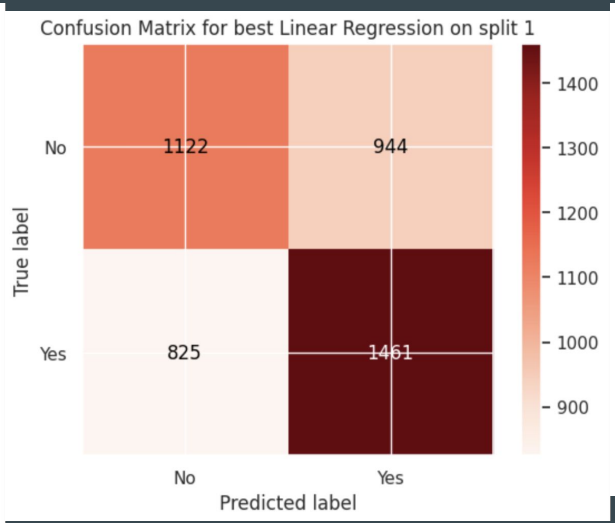


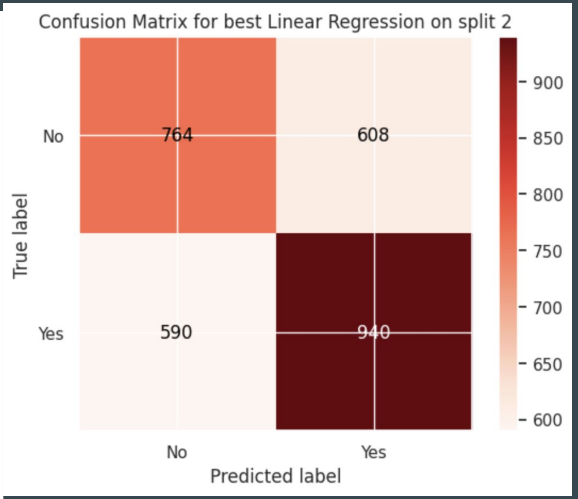
Linear Regression Metrics for Splits



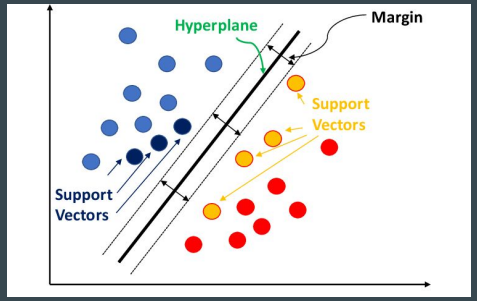
Linear Regression Best Performers







Support Vector Machine (Linear Kernel)



- Overview:
- Parameters - Studied different regularization values
- Used LinearSVC over SVC(kernel='linear') for large improvement in metrics and execution time. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
  - *'Similar to SVC with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.'*
- Features - Performed better with condensed set of features
- Predicts - Worse than logistic reg but improved over perceptron. Worse than DT

C = .45, f=small, LinSVC	0.605006	0.60266	0.600908	0.600747	0.628486
C = 1.0, f=small, LinSVC	0.60817	0.606855	0.601124	0.599059	0.625963
C = .25, f=large, LinSVC	0.537687	0.553738	0.509532	0.395681	0.679936
C = .25, f=Small, LinSVC	0.610184	0.607949	0.606179	0.606075	0.624352

Comparison of Acc/Prec/Recall/F1/RMSE for differing values of C and feature set sizes (small/large)

SVM-Linear Metrics for Splits

Chose 80/20. Better average overall for Class 'It Matters'
Learning Curve appears to be less variable than 70/30 & 50/50

****

Model: LinearSVC at split: split_0.5/0.5

	precision	recall	f1-score	support
Doesn't Matter	0.60	0.69	0.64	4559
It Matters	0.59	0.49	0.54	4131
accuracy			0.60	8690
macro avg	0.60	0.59	0.59	8690
weighted avg	0.60	0.60	0.59	8690

****

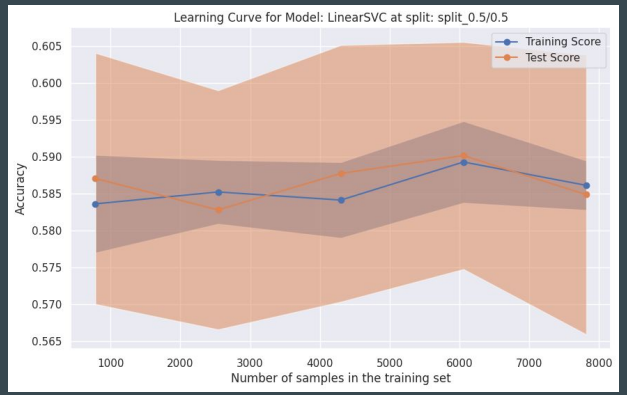
Model: LinearSVC at split: split_0.7/0.3

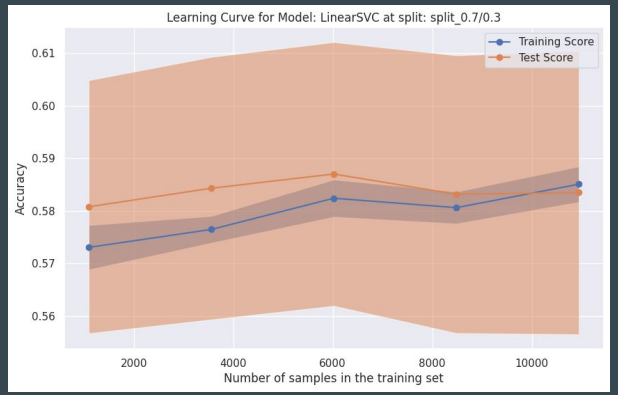
	precision	recall	f1-score	support
Doesn't Matter	0.61	0.70	0.65	2735
It Matters	0.61	0.51	0.55	2479
accuracy			0.61	5214
macro avg	0.61	0.61	0.60	5214
weighted avg	0.61	0.61	0.61	5214

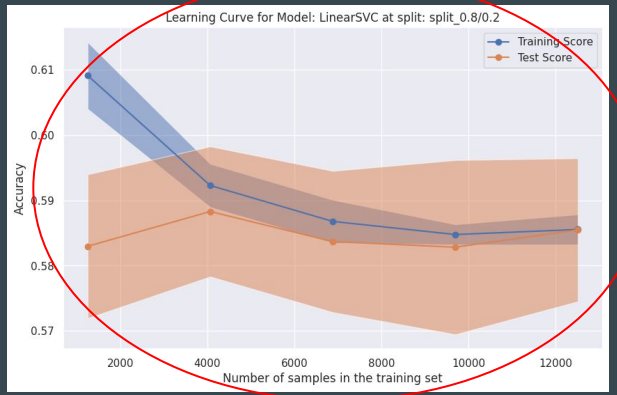
****

Model: LinearSVC at split: split_0.8/0.2

	precision	recall	f1-score	support
Doesn't Matter	0.63	0.64	0.63	1846
It Matters	0.58	0.57	0.58	1630
accuracy			0.61	3476
macro avg	0.61	0.61	0.61	3476
weighted avg	0.61	0.61	0.61	3476





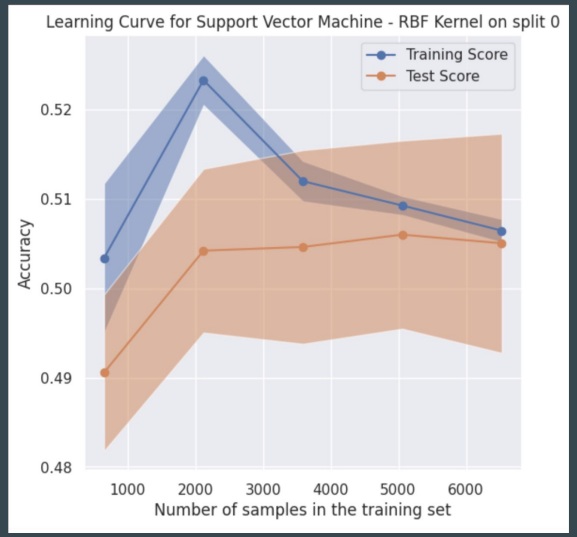


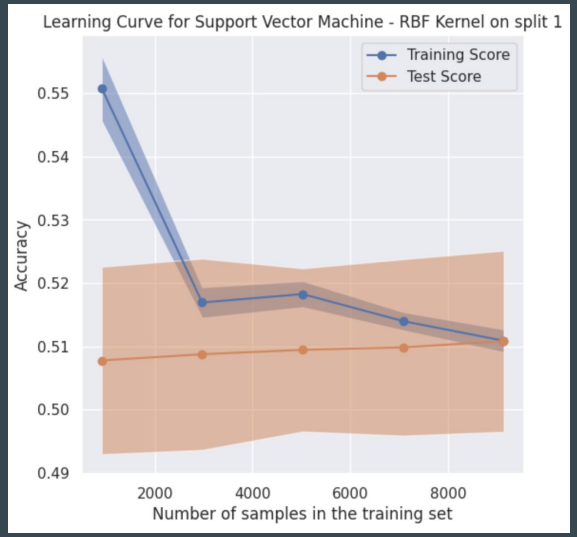
Support Vector Machine - RBF Kernel

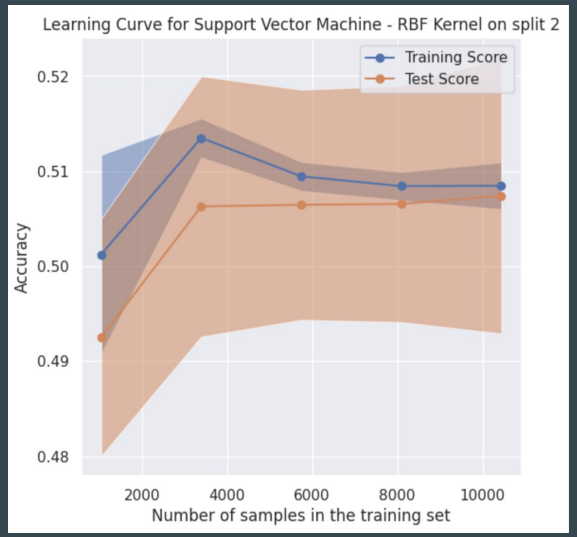
- Overview - Supervised learning model used for classification.
- Radial Basis Function Kernel is a stationary kernel and is sometimes referred to as the ‘squared exponential kernel’

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

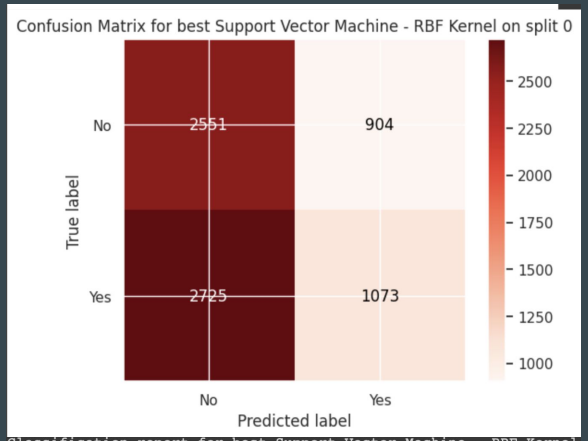
SVM - RBF Metrics for Splits





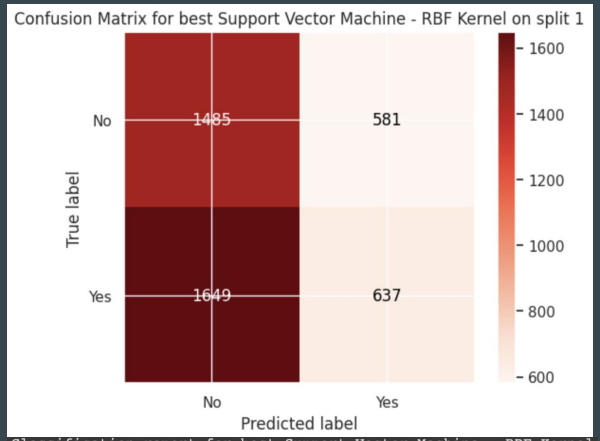


SVM - RBF Kernel Best Performers



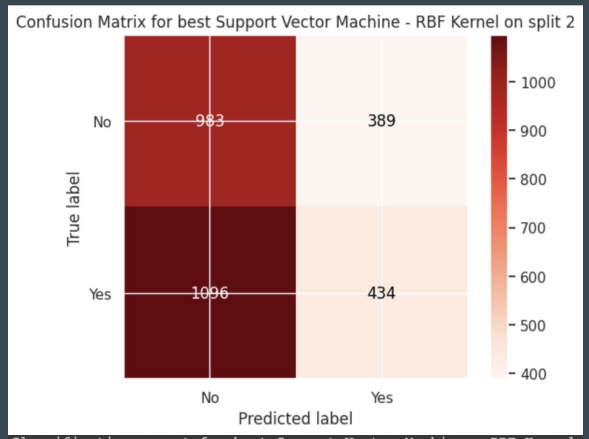
Classification report for best Support Vector Machine - RBF Kernel

	precision	recall	f1-score	support
0	0.48	0.74	0.58	3455
1	0.54	0.28	0.37	3798
accuracy			0.50	7253
macro avg	0.51	0.51	0.48	7253
weighted avg	0.51	0.50	0.47	7253



Classification report for best Support Vector Machine - RBF Kernel

	precision	recall	f1-score	support
0	0.47	0.72	0.57	2066
1	0.52	0.28	0.36	2286
accuracy			0.49	4352
macro avg	0.50	0.50	0.47	4352
weighted avg	0.50	0.49	0.46	4352



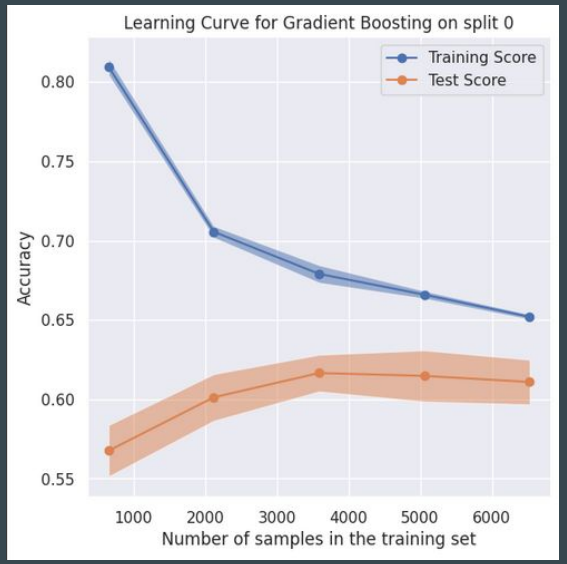
Classification report for best Support Vector Machine - RBF Kernel

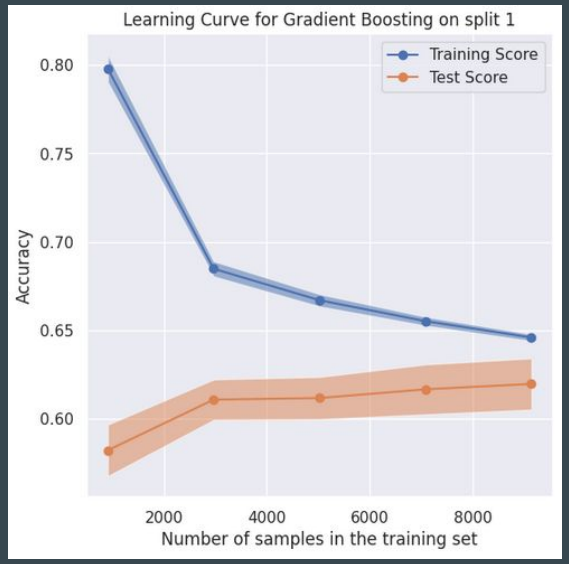
	precision	recall	f1-score	support
0	0.47	0.72	0.57	1372
1	0.53	0.28	0.37	1530
accuracy			0.49	2902
macro avg	0.50	0.50	0.47	2902
weighted avg	0.50	0.49	0.46	2902

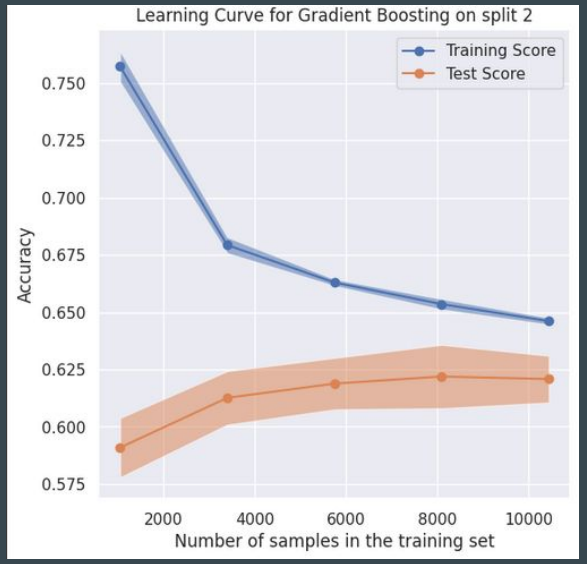
Gradient Boosting

- Overview
- Parameters
  - loss = log loss - works well on probabilistic labels
  - learning rate = 0.1 - learning rate is sufficiently small for model to not overshoot when performing backpropagation
  - n_estimators = 100 - bumping up to 1000 changed nothing about performance
  - subsample = 1.0 - setting to values between (0,1) didn't meaningfully affect performance either
  - criterion = 'friedman_mse' - 'squared error' seems to just shift metrics between the folds around without meaningfully changing the results
  - min_samples_split: 2 - has a negligible effect on prediction results
- Features used: 'sex_data', 'orientation_data', 'height', 'religion_data', 'religion_intensity'

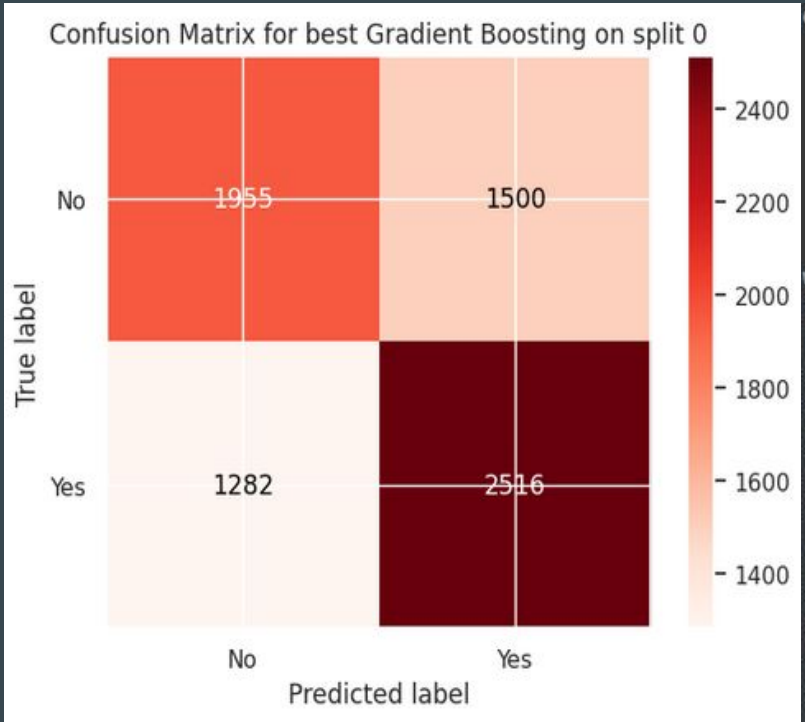
Gradient Boosting Learning Curves - Split 0







Gradient Boosting Best Model Confusion Matrix - Split 0 (0)



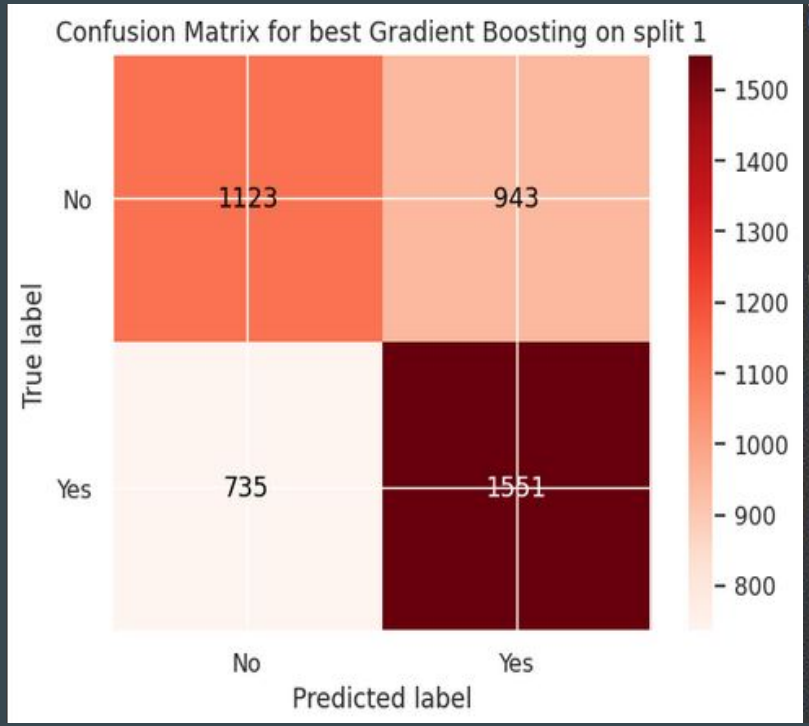
Classification report for best Gradient Boosting on unseen data on split 0

	precision	recall	f1-score	support
0	0.60	0.57	0.58	3455
1	0.63	0.66	0.64	3798
accuracy			0.62	7253
macro avg	0.62	0.61	0.61	7253
weighted avg	0.62	0.62	0.62	7253

Gradient Boosting Metrics for 10-fold on split 0

Fold	Accuracy	Precision	Recall	F1
0	0.627	0.626	0.627	0.627
1	0.601	0.6	0.601	0.599
2	0.605	0.604	0.605	0.602
3	0.634	0.634	0.634	0.632
4	0.63	0.63	0.63	0.628
5	0.597	0.596	0.597	0.596
6	0.597	0.596	0.597	0.596
7	0.601	0.601	0.601	0.601
8	0.618	0.617	0.618	0.616
9	0.6	0.599	0.6	0.599

Gradient Boosting Best Model Confusion Matrix - Split 1 (5)



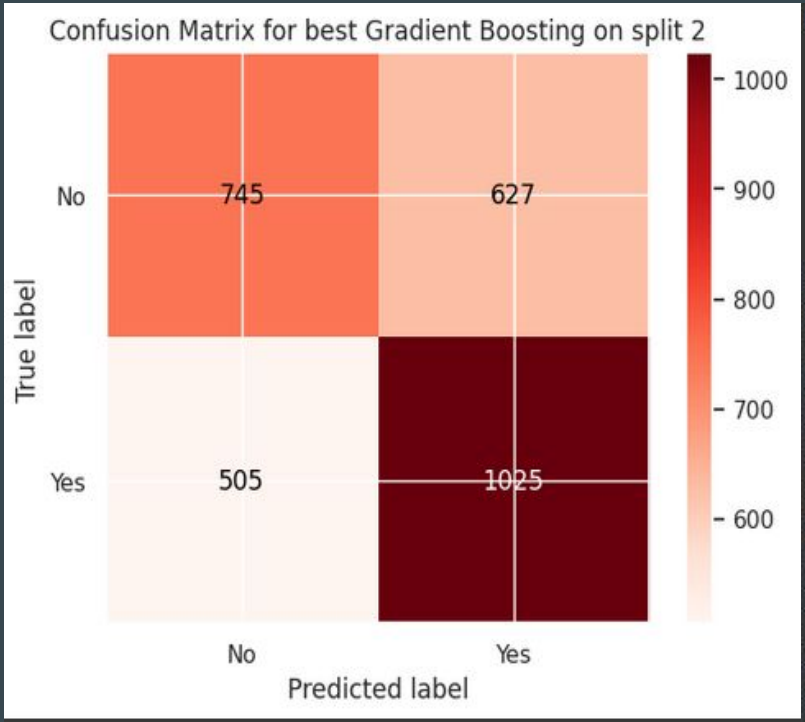
Classification report for best Gradient Boosting on unseen data on split 1

	precision	recall	f1-score	support
0	0.60	0.54	0.57	2066
1	0.62	0.68	0.65	2286
accuracy			0.61	4352
macro avg	0.61	0.61	0.61	4352
weighted avg	0.61	0.61	0.61	4352

Gradient Boosting Metrics for 10-fold on split 1

Fold	Accuracy	Precision	Recall	F1
0	0.615	0.614	0.615	0.613
1	0.601	0.601	0.601	0.598
2	0.6	0.6	0.6	0.598
3	0.609	0.608	0.609	0.608
4	0.627	0.627	0.627	0.622
5	0.649	0.649	0.649	0.649
6	0.636	0.637	0.636	0.633
7	0.62	0.619	0.62	0.619
8	0.613	0.612	0.613	0.612
9	0.626	0.625	0.626	0.624

Gradient Boosting Best Model Confusion Matrix - Split 2 (2)



Classification report for best Gradient Boosting on unseen data on split 2

	precision	recall	f1-score	support
0	0.60	0.54	0.57	1372
1	0.62	0.67	0.64	1530
accuracy			0.61	2902
macro avg	0.61	0.61	0.61	2902
weighted avg	0.61	0.61	0.61	2902

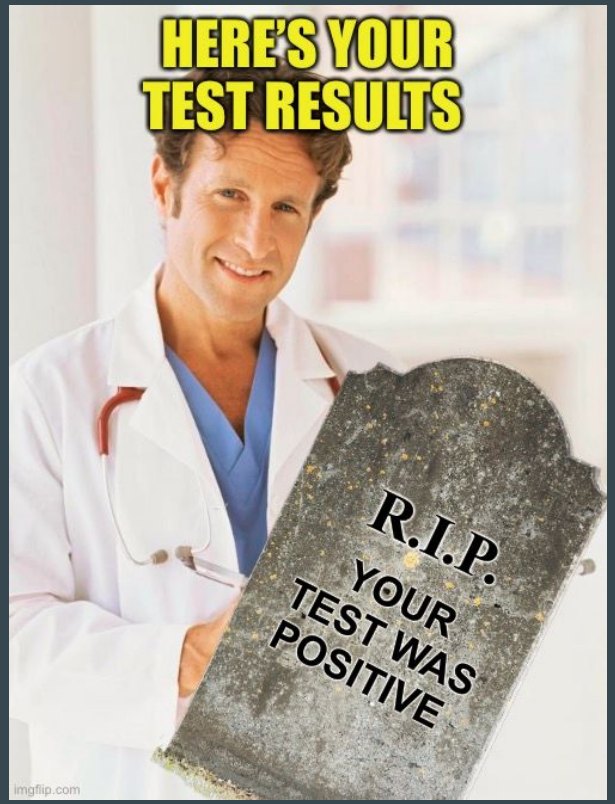
Gradient Boosting Metrics for 10-fold on split 2

Fold	Accuracy	Precision	Recall	F1
0	0.616	0.615	0.616	0.614
1	0.617	0.616	0.617	0.614
2	0.639	0.639	0.639	0.637
3	0.63	0.629	0.63	0.627
4	0.618	0.617	0.618	0.617
5	0.622	0.622	0.622	0.62
6	0.631	0.631	0.631	0.628
7	0.595	0.594	0.595	0.593
8	0.627	0.626	0.627	0.625
9	0.62	0.619	0.62	0.618

Gradient Boosting Model Selection

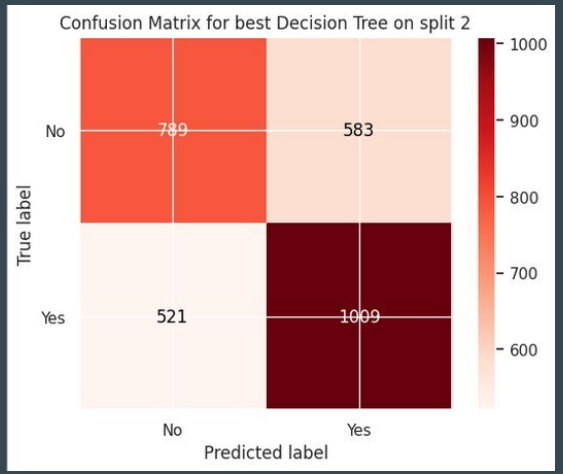
- Interestingly, the model trained on split 0 was the most performant out of the three splits, though this might just be due to a lucky split, and the difference was minimal (only around 1% difference in metrics)
- Performance on the third split is the most consistent, with metrics in the same range as the other two splits, therefore it seems to have the lowest generalization error
- -> Choose model trained on fold 2 of split 2

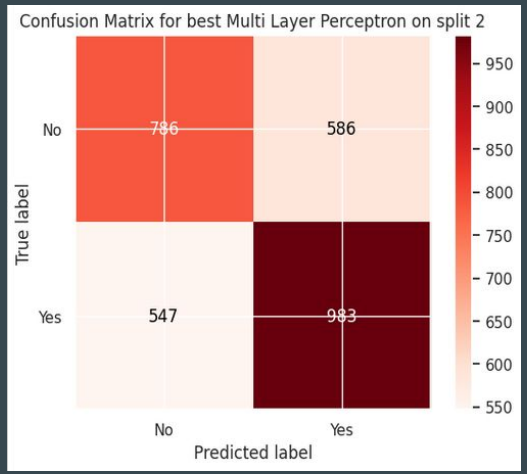
Section 3 - The ~~good~~ stuff

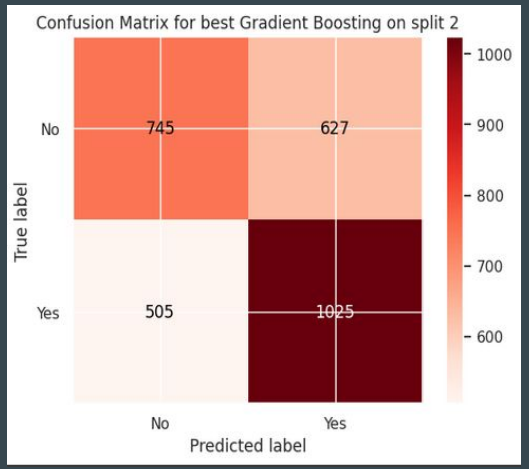


Best performers - Dtree, MLP, Gradient Boosting

	Model name	Accuracy	Precision	Recall	F1
0	Decision Tree (80/20 split)	62	62	62	62
1	Multi-layer Perceptron (80/20 split)	61	61	61	61
2	Gradient Boosting (80/20 split)	61	61	61	61

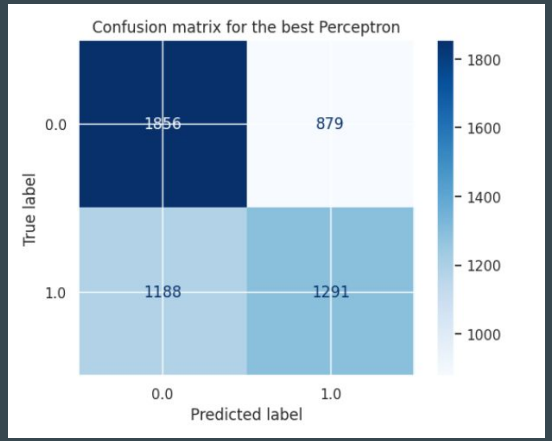


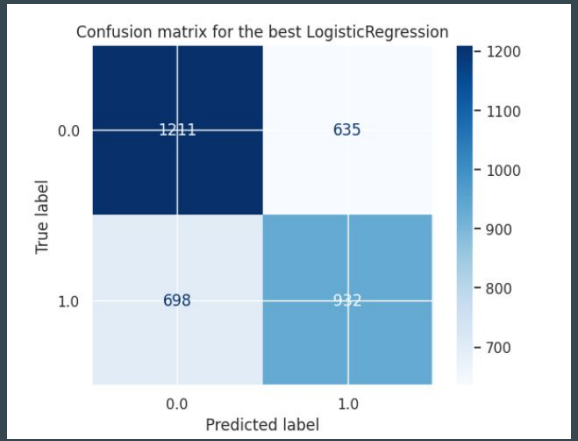


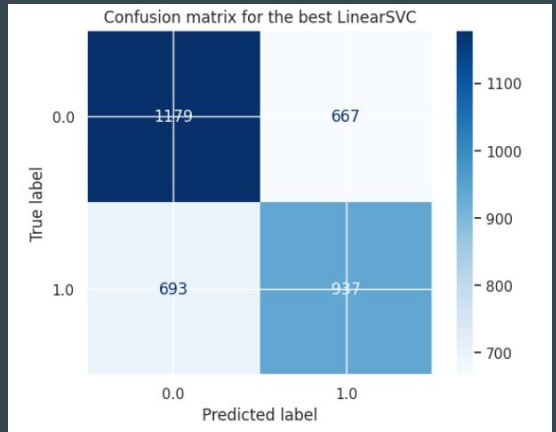


Best performers - Perceptron, Logistic Regression, Linear SVM

	Model Name	Accuracy	Precision	Recall	F1-score	RMSE
0	Perceptron, split=split_0.7/0.3	0.603567	0.602327	0.599693	0.598857	0.629629
1	LogisticRegression, split=split_0.8/0.2	0.616513	0.614565	0.613896	0.614027	0.619263
2	LinearSVC, split=split_0.8/0.2	0.608746	0.606986	0.606762	0.606840	0.625503

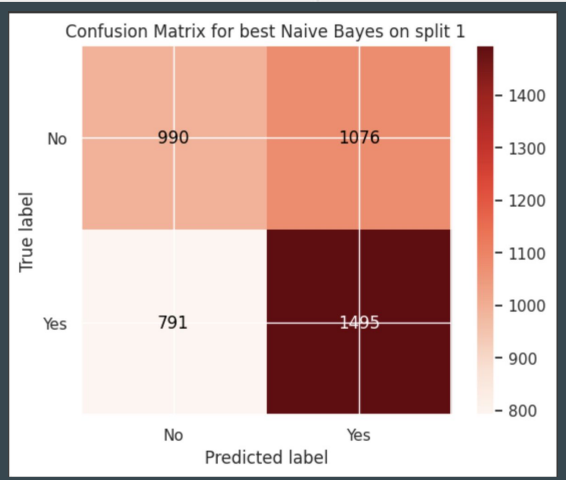


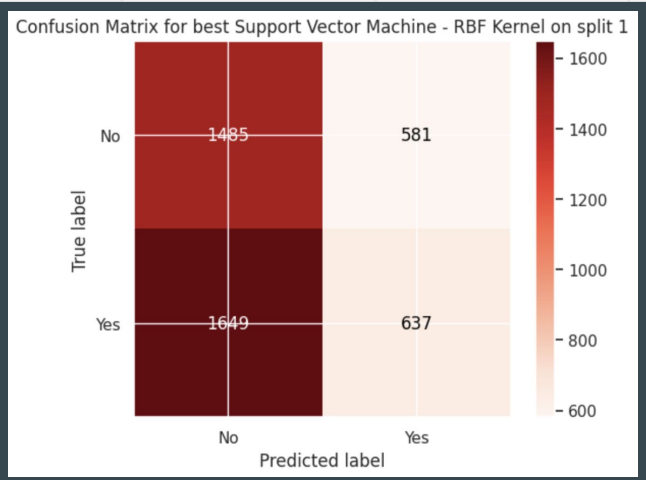




Best performers - Linreg, GNB, rbf SVM

Splits	Accuracy	Precision	Recall	F1
GNB (70/30)	0.57	0.57	0.57	0.57
Linear Regression (80/20)	0.05	-	-	-
SVM RBF (70/30)	0.5	0.5	0.49	0.46





BEST OVERALL MODEL

Because most of our better performing models performed roughly the same (61-62% for all scores), we have opted to choose the most interpretable model for our choice of ‘best overall model’

-> Decision tree trained on fold 8 of split 2

Conclusions

Did the different sampling help or made things worse?

- From our observations it seems that the samples that provided a greater training size set (80/20) resulted in more generalizable models.

What model generalizes better?

- Decision tree

What would you do differently?

- Pick a better problem and a cleaner dataset to start with. Responses were highly variable, all fields had missing data, or custom options. The data is also antiquated in some ways (eg internet and therefore dating site use in 2012 likely skewed towards higher income, and education more than in 2023)

Well at least we can predict something...

- Sex is trivial predict, every model does very well
- Note that Sex is HEAVILY correlated with Height, without it predicting is less easy
- Chose to not to pivot to this as it wouldn't really accomplish anything of interest

Correlation chart highlighting Sex vs non-normalized Height

```
numeric.corr()
```

	sign_intensity	age	status_data	sex_data	o
sign_intensity	1.000000	0.021602	0.001815	0.145914	
age	0.021602	1.000000	-0.058474	0.059859	
status_data	0.001815	-0.058474	1.000000	0.018308	
sex_data	0.145914	0.059859	0.018308	1.000000	
orientation_data	0.077667	-0.032686	0.078092	-0.051338	
body_type_data	0.050065	-0.000470	0.068933	0.269166	
diet_data	0.025280	-0.018713	0.071215	0.008281	
drinks_data	-0.011177	-0.119073	-0.002425	-0.001053	
drugs_data	0.044252	-0.156981	0.075922	-0.079778	
education_data	-0.074440	0.231623	-0.029488	0.063437	
height_data	-0.110963	-0.037120	-0.007334	-0.656208	
sign_data	0.007643	-0.005894	-0.003050	-0.008136	

Models trained for Sex, with and without height

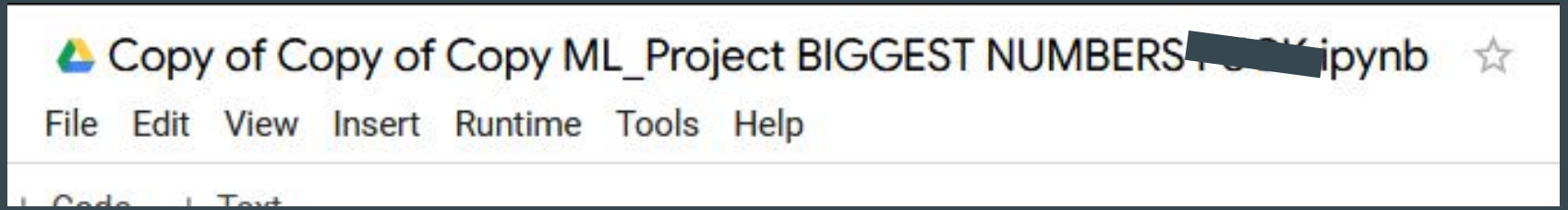
	Model Name	Accuracy	Precision	Recall	F1-score	RMSE
0	Perceptron	0.825853	0.815366	0.814107	0.814722	0.417309
1	DecisionTreeClassifier	0.840161	0.834116	0.825869	0.829458	0.399799
2	RandomForestClassifier	0.860184	0.854259	0.847179	0.850375	0.373920
3	GaussianNB	0.808113	0.796615	0.796701	0.796658	0.438049

	Model Name	Accuracy	Precision	Recall	F1-score	RMSE
DecisionTreeClassifier	0.721519	0.745396	0.654569	0.655679	0.527713
GaussianNB	0.623130	0.594565	0.589707	0.590875	0.613897
LogisticRegression	0.634925	0.599489	0.570256	0.562166	0.604214
GradientBoostingClassifier	0.765535	0.764075	0.726361	0.735404	0.484216

Section 4...let's try something stupid

Gradient boosting - But in the laziest possible way

- Noticed adding more random features to gradient boosting improved metrics a bit
- Therefore: Maximum # of features == more numbers.
- Threw all features into standard `OrdinalEncoder()` instead of intelligently hand coded values (noted that correlations remained ~ the same after this translation)
- Feature set of all 22 possible including essay length.



PREPARE FOR THE BIGGEST NUMBERS YOU'VE EVER SEEN

****

Model: GradientBoostingClassifier at split: split_0.5/0.5

	precision	recall	f1-score	support
It Matters	0.64	0.68	0.66	4559
Doesn't Matter	0.62	0.57	0.59	4131
accuracy			0.63	8690
macro avg	0.63	0.63	0.63	8690
weighted avg	0.63	0.63	0.63	8690

****

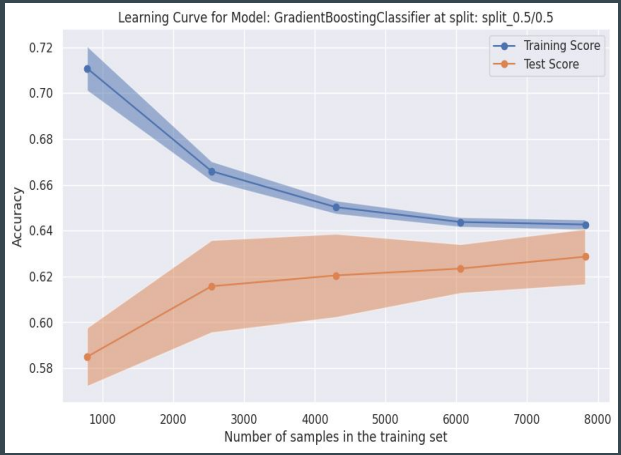
Model: GradientBoostingClassifier at split: split_0.7/0.3

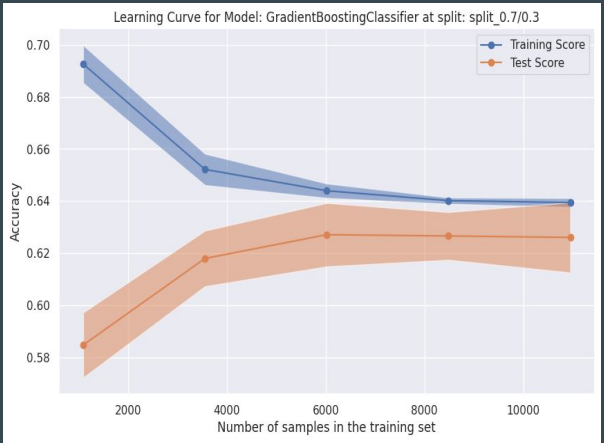
	precision	recall	f1-score	support
It Matters	0.65	0.72	0.68	2735
Doesn't Matter	0.64	0.57	0.60	2479
accuracy			0.65	5214
macro avg	0.65	0.64	0.64	5214
weighted avg	0.65	0.65	0.64	5214

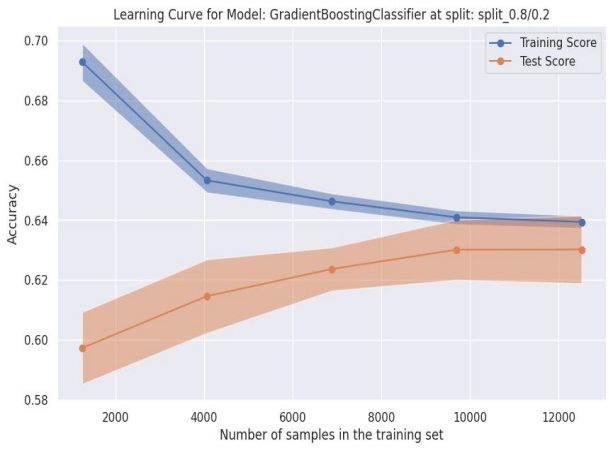
****

Model: GradientBoostingClassifier at split: split_0.8/0.2

	precision	recall	f1-score	support
It Matters	0.65	0.71	0.68	1846
Doesn't Matter	0.64	0.57	0.60	1630
accuracy			0.65	3476
macro avg	0.64	0.64	0.64	3476
weighted avg	0.65	0.65	0.64	3476

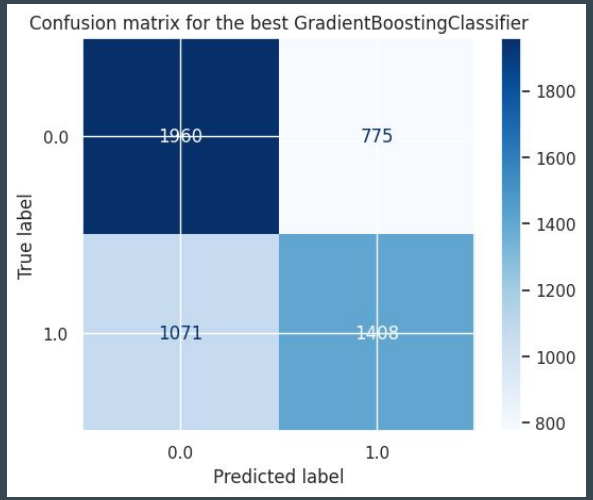


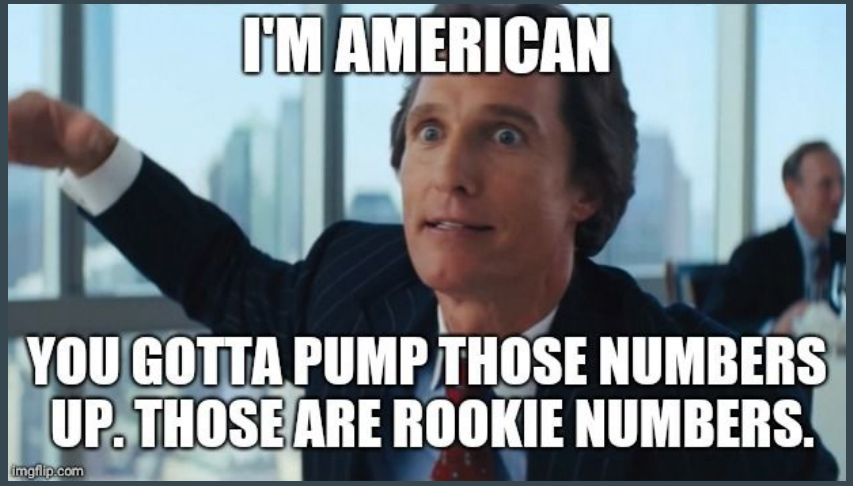




BIG 65s! AWWWWW YEA

	Model Name	Accuracy	Precision	Recall	F1-score	RMSE
0	Perceptron, split=split_0.7/0.3	0.603567	0.602327	0.599693	0.598857	0.629629
1	DecisionTreeClassifier, split=split_0.7/0.3	0.602608	0.602967	0.603196	0.602485	0.630390
2	GaussianNB, split=split_0.7/0.3	0.595896	0.596060	0.596288	0.595712	0.635692
3	LogisticRegression, split=split_0.8/0.2	0.609033	0.606893	0.605957	0.606040	0.625273
4	GradientBoostingClassifier, split=split_0.7/0.3	0.645953	0.645818	0.642304	0.641940	0.595018





Tried seed other than 1234

<- Click to play sad trombone.mp3

In reality, it's about as bad at predicting 'doesn't matter' as any other attempt, but is more fit to the other class.


```
*******************************************
Model: GradientBoostingClassifier at split: split_0.5/0.5
              precision    recall  f1-score   support

    It Matters       0.62      0.71      0.66      4583
Doesn't Matter       0.62      0.52      0.57      4107

      accuracy                           0.62      8690
     macro avg       0.62      0.62      0.61      8690
  weighted avg       0.62      0.62      0.62      8690

*******************************************
Model: GradientBoostingClassifier at split: split_0.7/0.3
              precision    recall  f1-score   support

    It Matters       0.63      0.69      0.66      2720
Doesn't Matter       0.62      0.55      0.58      2494

      accuracy                           0.62      5214
     macro avg       0.62      0.62      0.62      5214
  weighted avg       0.62      0.62      0.62      5214

*******************************************
Model: GradientBoostingClassifier at split: split_0.8/0.2
              precision    recall  f1-score   support

    It Matters       0.62      0.68      0.65      1817
Doesn't Matter       0.61      0.55      0.58      1659

      accuracy                           0.62      3476
     macro avg       0.62      0.62      0.61      3476
  weighted avg       0.62      0.62      0.62      3476

*******************************************
```

Questions? Concerns for sanity?

END