

# Predicting Divorce from Demographic Traits

**Helen Skinner**

# Contents

- Purpose and justification
- Data
- Model
- Results
- Uses, shortcomings and further work
- Acknowledgements

# Purpose and justification

**Purpose:** Is it possible to predict whether an individual has ever been divorced based on demographic traits?

## **Justification:**

- Commercial uses
- Intervention targeting and potential prevention
- General interest

# Data – Overview

## General Social Survey 2012



4,820 respondents

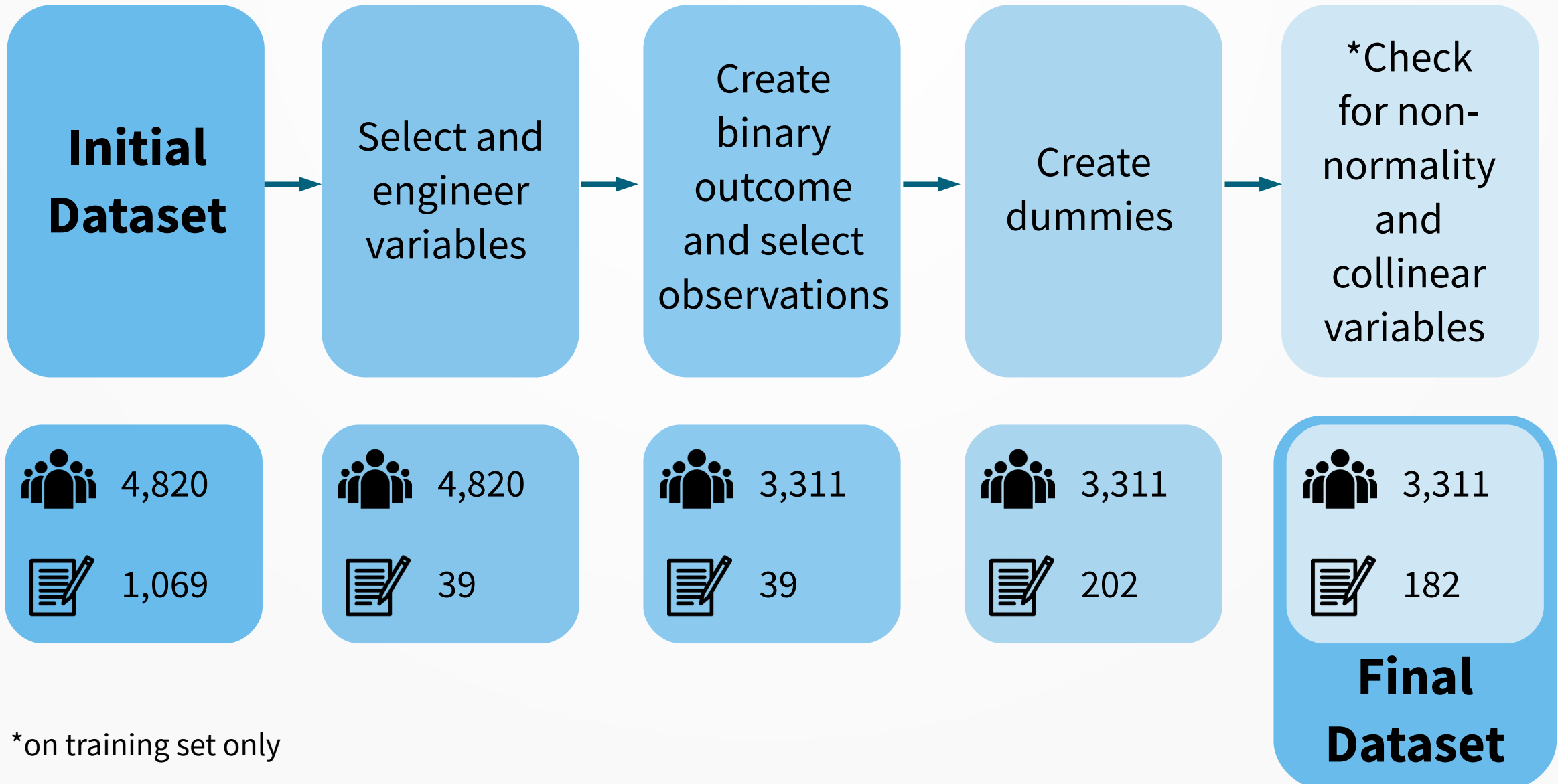


1,069 variables


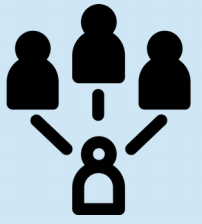
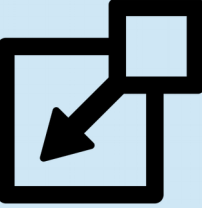
### **Significant data preprocessing required:**

- Respondents did not answer every question
- Inconsistent coding for ‘inapplicable’, ‘don’t know’ and ‘no answer’
- Potential for label leakage

# Data – Preprocessing workflow



# Data – Select and engineer variables

<b>Dropping</b> 	<ul style="list-style-type: none"><li>• Low response rates</li><li>• Label leakage</li><li>• Manual selection required</li></ul>	e.g. Most opinion questions e.g. Dwelling type
<b>Grouping</b> 	<ul style="list-style-type: none"><li>• Reduce noise</li><li>• Reduce overfitting</li></ul>	e.g. Religion e.g. Occupation
<b>Imputing</b> 	<ul style="list-style-type: none"><li>• Potentially important but some missing data</li><li>• Sensible method available</li></ul>	e.g. Income using logical rules

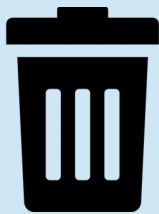
# Data – Create binary outcome and select observations

## Binarizing

01  
10

- Compare “has been divorced” vs “has never been divorced”
- Remove all single people as not applicable
- Previously divorced and now remarried counts as “has been divorced”
- Widowed counts as “has never been divorced”

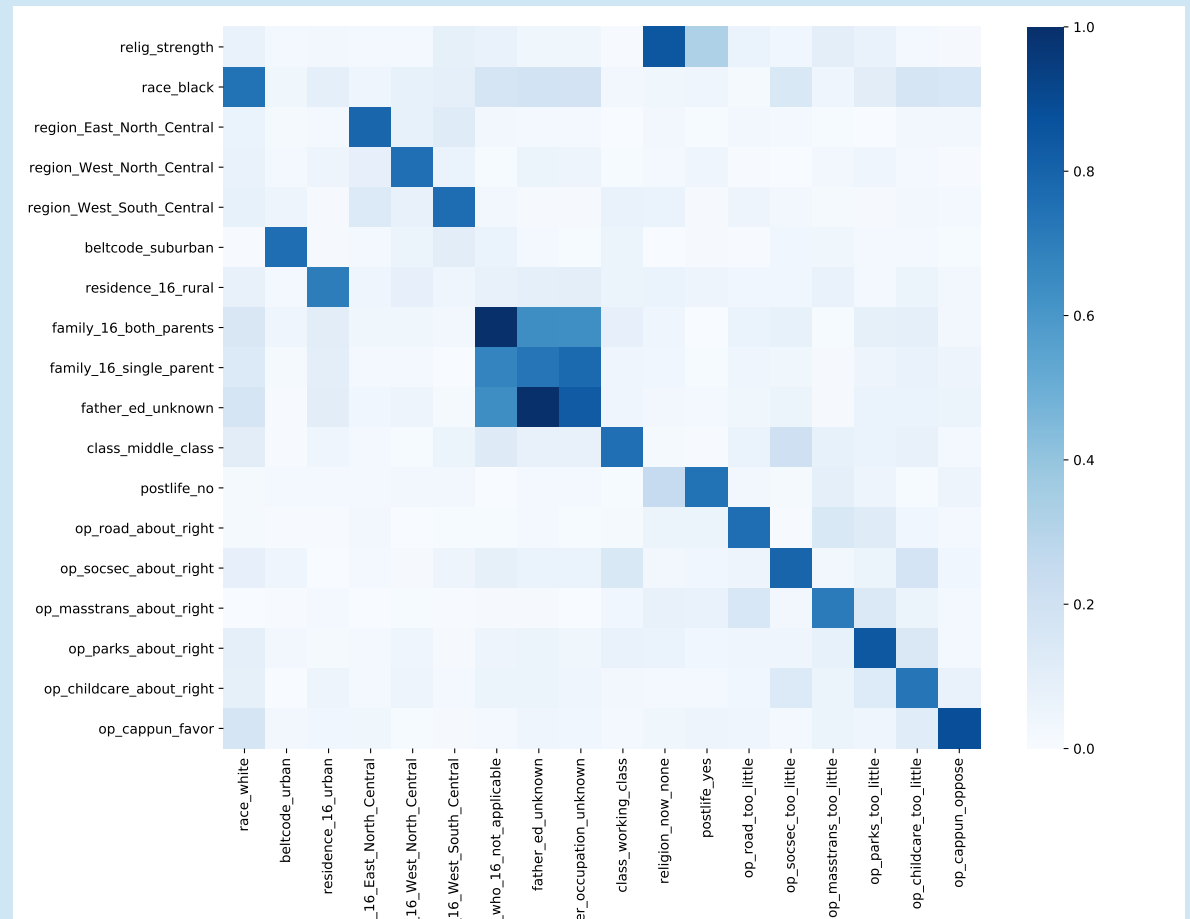
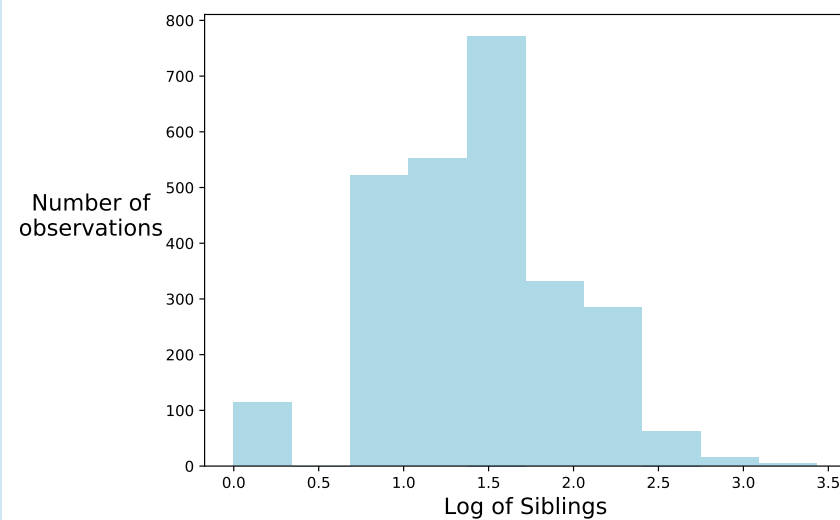
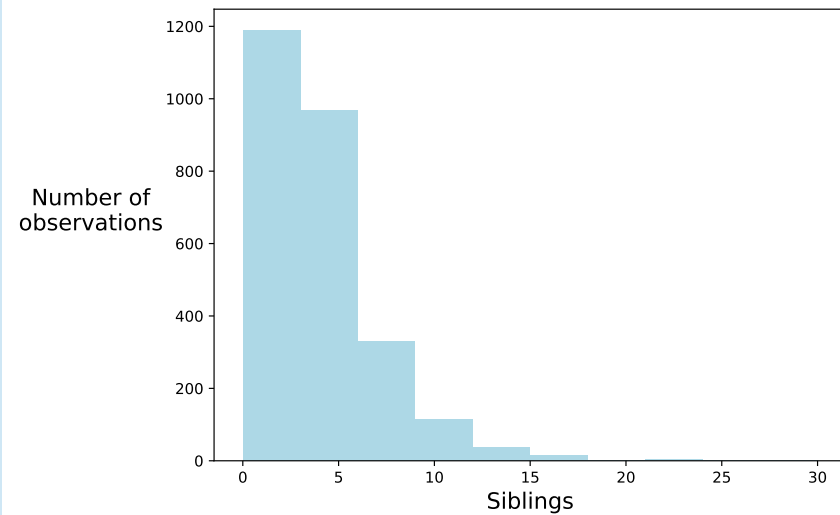
## Dropping



- Remove single people as not applicable
- Remove observations with missing data

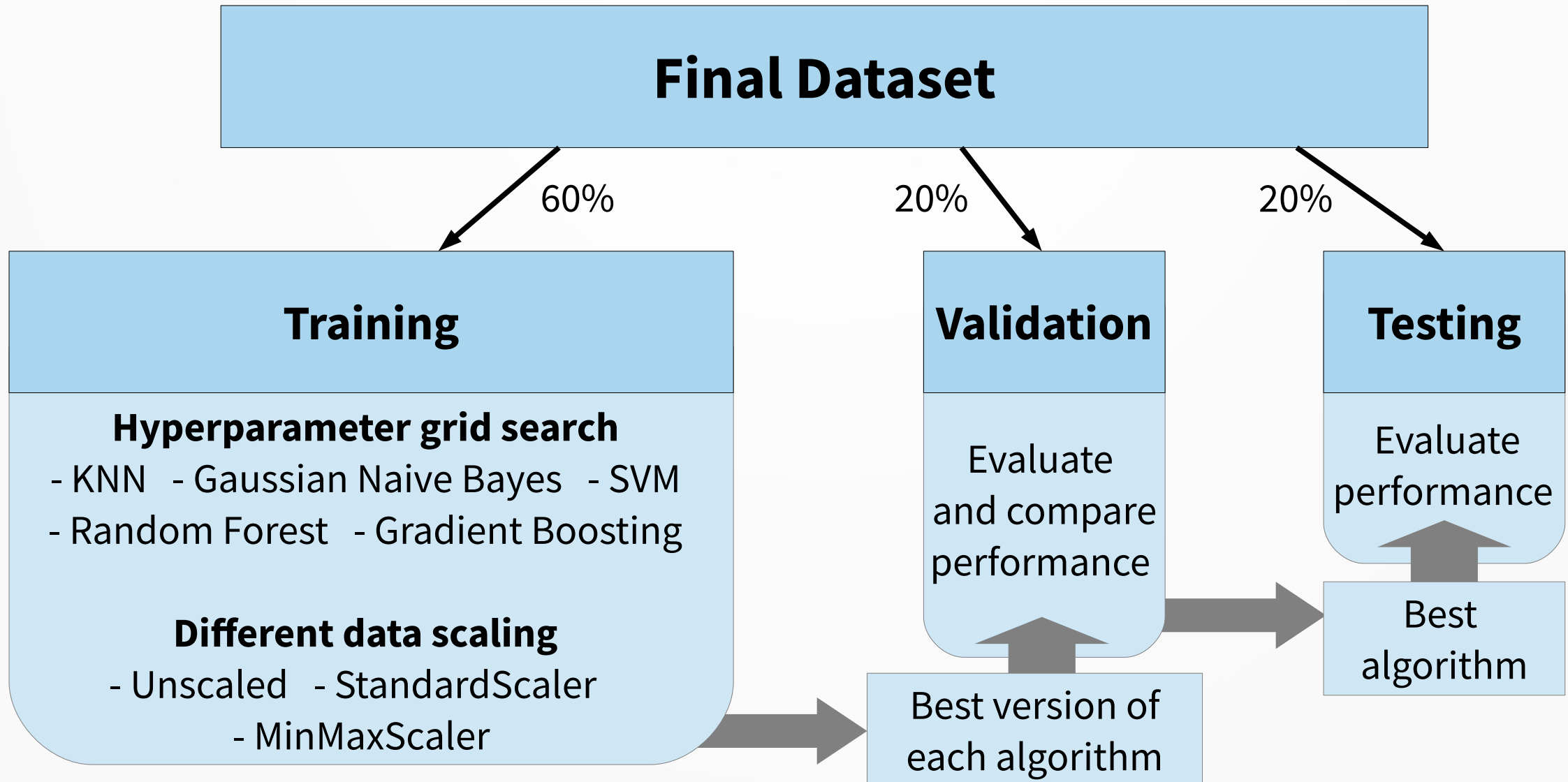


# Data – Non-normality and collinearity





# Modeling – Selection pathway



# Modeling – Training strategy

Algorithm	Tuned hyperparameters
<b>KNN</b>	<ul style="list-style-type: none"><li>• n_neighbors</li><li>• weights</li></ul>
<b>Gaussian Naive Bayes</b>	<ul style="list-style-type: none"><li>• var_smoothing</li></ul>
<b>Random Forest</b>	<ul style="list-style-type: none"><li>• n_estimators</li><li>• criterion</li><li>• max_depth</li><li>• max_features</li></ul>
<b>Gradient Boosting</b>	<ul style="list-style-type: none"><li>• loss</li><li>• learning_rate</li><li>• max_depth</li><li>• max_features</li></ul>
<b>SVM</b>	<ul style="list-style-type: none"><li>• C</li><li>• gamma</li></ul>

- Aim to find best version of each algorithm
- Tune each algorithm to find
  - Best hyperparameters
  - Best data scaling
- Use f1 score to assess
- All algorithms trained on:
  - Unscaled data
  - StandardScaler
  - MinMaxScaler

# Modeling – Training results

Algorithm	Best hyperparameters	Best data scaling
<b>KNN</b>	<ul style="list-style-type: none"><li>• n_neighbors = 18</li><li>• weights = distance</li></ul>	Unscaled
<b>Gaussian Naive Bayes</b>	<ul style="list-style-type: none"><li>• var_smoothing = 1e-09</li></ul>	StandardScaler
<b>Random Forest</b>	<ul style="list-style-type: none"><li>• n_estimators = 100</li><li>• criterion = entropy</li><li>• max_depth = 4</li><li>• max_features = None</li></ul>	No difference
<b>Gradient Boosting</b>	<ul style="list-style-type: none"><li>• loss = exponential</li><li>• learning_rate = 0.1</li><li>• max_depth = 4</li><li>• max_features = None</li></ul>	No difference
<b>SVM</b>	<ul style="list-style-type: none"><li>• C = 10</li><li>• gamma = 0.0005</li></ul>	Unscaled

To get the best performance from each algorithm:

- Use data scaled in this way
  - Use these hyperparameters

# Modeling – Validation results

KNN

		Actual	
		0	1
Predicted	0	249	142
	1	116	155

f1: 0.546 Accuracy: 0.610

G. Naive  
Bayes

		Actual	
		0	1
Predicted	0	24	10
	1	341	287

f1: 0.621 Accuracy: 0.470

Random  
Forest

		Actual	
		0	1
Predicted	0	187	69
	1	178	228

f1: 0.649 Accuracy: 0.627

Gradient  
Boosting

		Actual	
		0	1
Predicted	0	268	140
	1	97	157

f1: 0.570 Accuracy: 0.642

SVM

		Actual	
		0	1
Predicted	0	267	132
	1	98	165

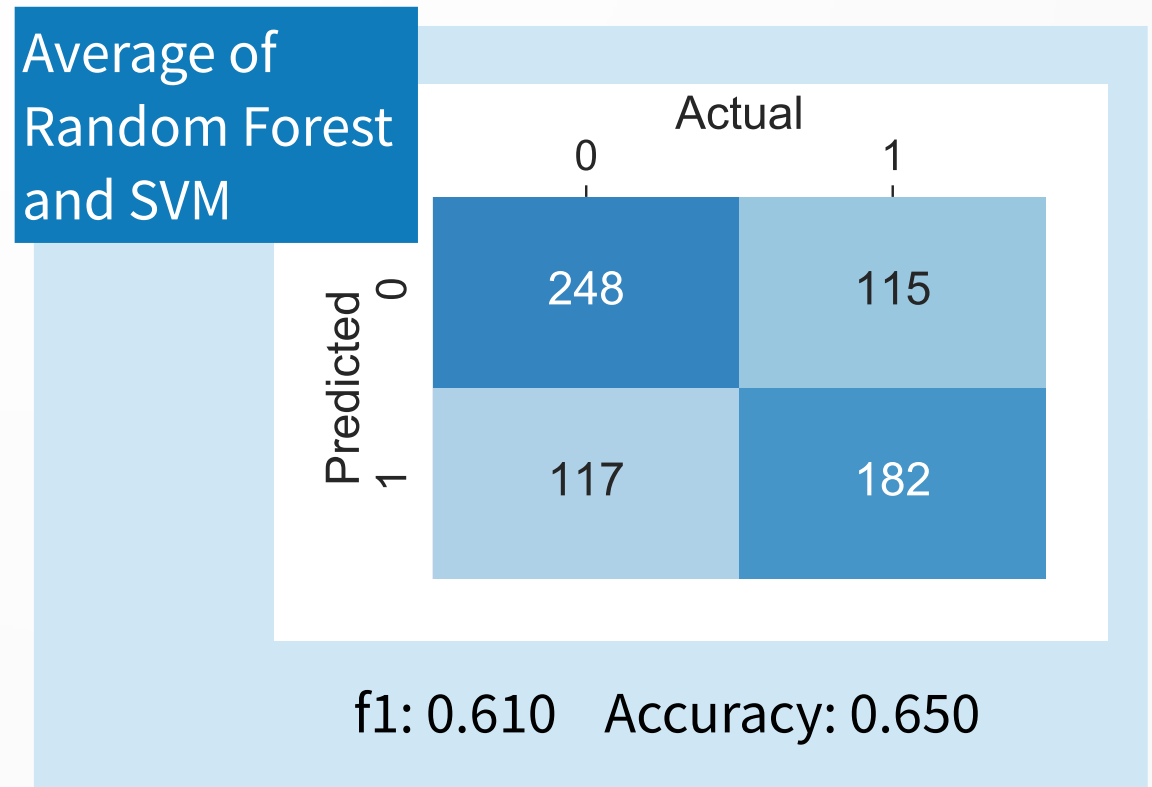
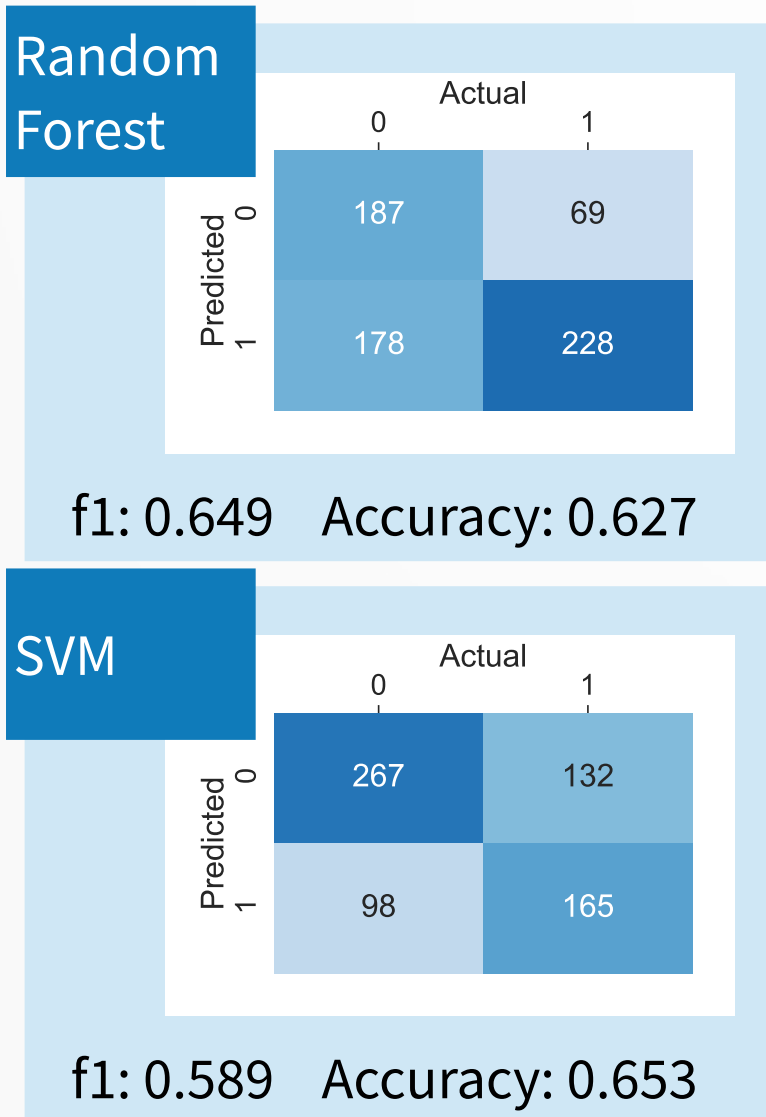
f1: 0.589 Accuracy: 0.653

Guessing

		Actual	
		0	1
Predicted	0	365	297
	1	0	0

f1: N/A Accuracy: 0.551

# Modeling – Validation results

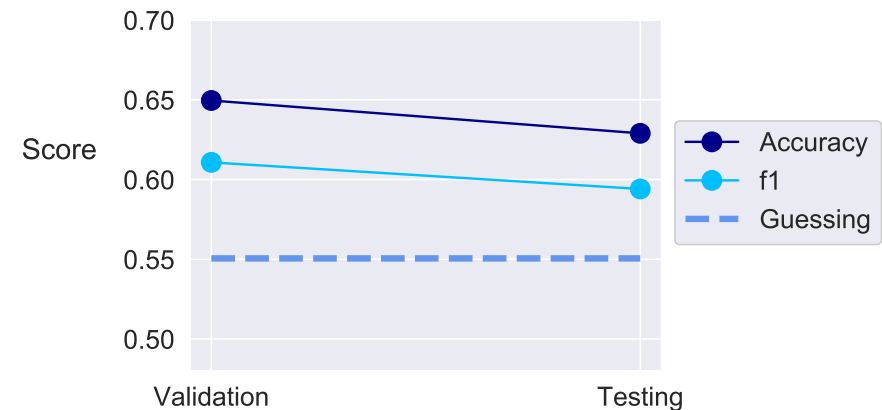


# Modeling – Testing results

Average of  
Random Forest  
and SVM

		Actual	
		0	1
Predicted	0	237	118
	1	128	180

f1: 0.594 Accuracy: 0.629



- Small reduction in accuracy on test set compared to validation
- Still more accurate than guessing all not divorced



# Practical uses of the model

- Advertising or actuarial
  - Counseling or legal services via social media to at risk groups
  - Insurance implications
- Intervention
  - Support or help for at risk groups
  - Charity or governmental
- General interest
  - Individuals may be interested to know personal probability
  - Either for decision-making or not



# Weak points of the model

- Accuracy
  - Approximately 8 percentage points better than guessing
- Feature importances
  - Difficult to extract due to use of SVM
- Scaling
  - Run time of 11.2 seconds for training set with 2,648 observations and testing set with 663 observations
  - Estimated run time of over 24 hours for datasets over ~5 million

# Further work

## **Test on alternative data**

- Other years of GSS available
- Would require significant data preprocessing

## **Try using PCA**

- Decrease computing time
- Increase difficulty in extracting feature importances

## **Different data**

- More observations
- Additional variables about marital information e.g. age married

## **Investigate feature importances**

- Straightforward for Random Forest
- Not so straightforward for SVM

## **Further feature engineering**

- Lower collinearity threshold
- Categorize occupations differently

## **Different model**

- Try non-binary classification
- Similar but different predictions e.g. whether someone has children

# Acknowledgements

- Technical advice and support gratefully received from
  - Jenny Yu
  - Tom Nickson
  - Technical Coaching and peer group via Slack
- Icons
  - Icons made by [OCHA](#) and [Freepik](#) from [www.flaticon.com](http://www.flaticon.com)

# **Appendix**

## **Additional slides for information purposes**

# Modeling – Validation results

KNN

		Actual	
		0	1
Predicted	0	68%	48%
	1	32%	52%

f1: 0.546 Accuracy: 0.610

G. Naive  
Bayes

		Actual	
		0	1
Predicted	0	7%	3%
	1	93%	97%

f1: 0.621 Accuracy: 0.470

Random  
Forest

		Actual	
		0	1
Predicted	0	51%	23%
	1	49%	77%

f1: 0.649 Accuracy: 0.627

Gradient  
Boosting

		Actual	
		0	1
Predicted	0	73%	47%
	1	27%	53%

f1: 0.570 Accuracy: 0.642

SVM

		Actual	
		0	1
Predicted	0	73%	44%
	1	27%	56%

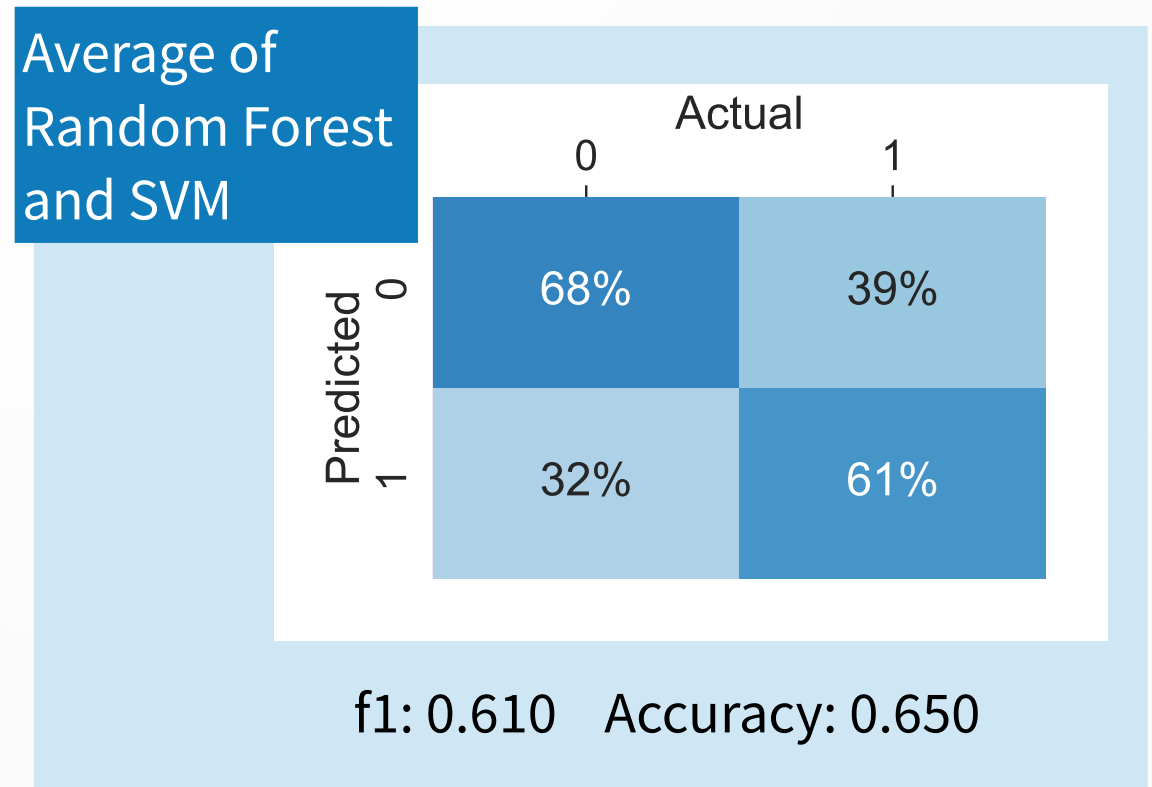
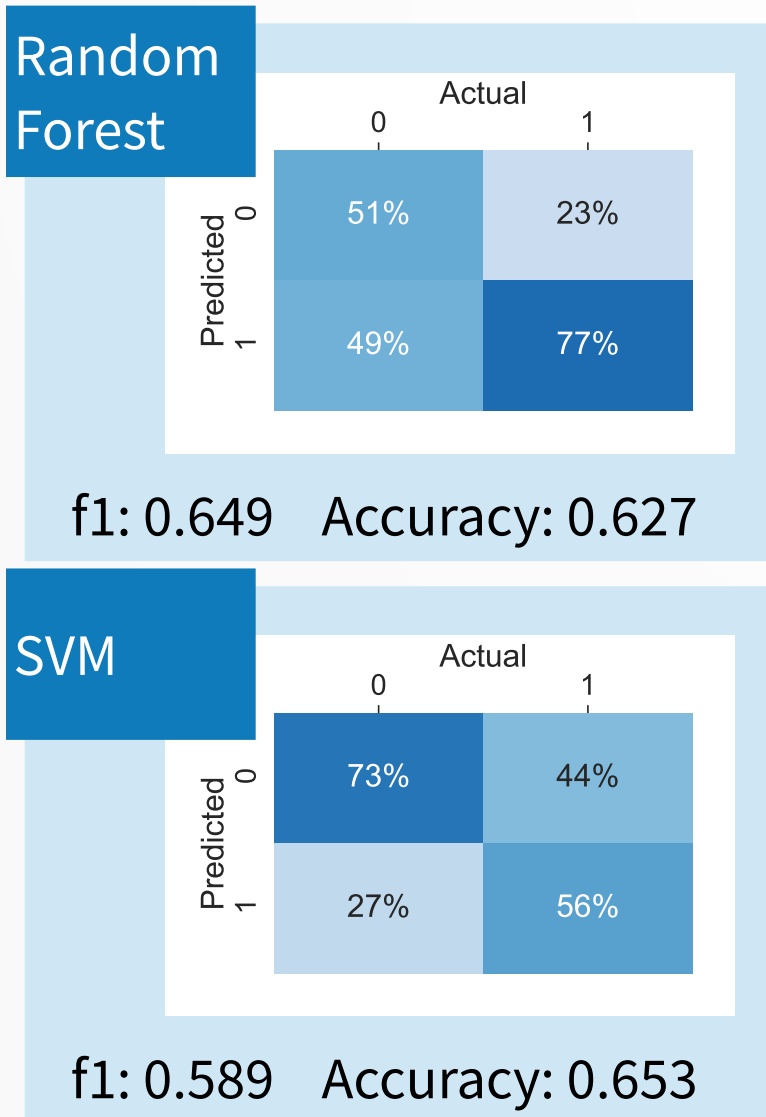
f1: 0.589 Accuracy: 0.653

Guessing

		Actual	
		0	1
Predicted	0	100%	100%
	1	0%	0%

f1: N/A Accuracy: 0.551

# Modeling – Validation results

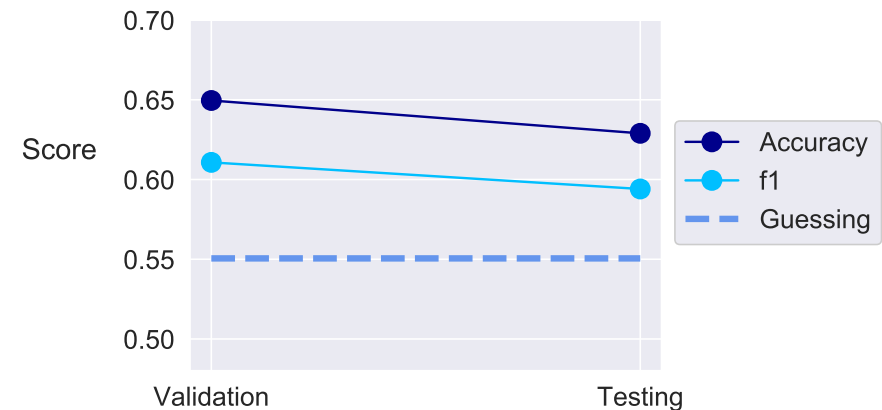


# Modeling – Testing results

Average of  
Random Forest  
and SVM

		Actual	
		0	1
Predicted	0	65%	40%
	1	35%	61%

f1: 0.594 Accuracy: 0.629



- Small reduction in accuracy on test set compared to validation
- Still more accurate than guessing all not divorced