Modeling opinions and the gender labor force participation gap in the United States

Eric Karsten [*] Ezra Max [†] Deniz Turkcapar [‡]

March 12, 2020

Contribution Statement

The allocation of work among members of the group was as follows: Eric wrote the data cleaning code and fit the LASSO and Ridge models. Ezra fit the random forest and bagging models. Deniz fit the kNN model using different variations to see how kNN performs. All three coauthors contributed to writing this paper and producing the associated presentation.

[*] ekarsten@uchicago.edu
[†] emax@uchicago.edu
[‡] dturkcapar@uchicago.edu

Contents

1 Introduction

1.1 Our Question

The labor market looks different for men and women, both in labor activity and in remuneration. There remains a sizeable gender pay gap in the United States, although it has narrowed substantially over the last 40 years and continues to diminish. On the other hand, the gap between men and women's participation in the labor market has not diminished since the financial crisis of 2008-2009: women's labor force participation rate (LFPR) was 79.5% of men's LFPR in 2000, 82.3% of men's LFPR in 2010, and 82.1% of men's LFPR in 2019 (WBG). The stubbornness of the FLPR despite wage growth for female workers raises the question of which factors drive men and women's labor force participation, and how the dynamics of female labor force participation differ from those of male labor force participation. What non-financial factors play into women's decisions to work?

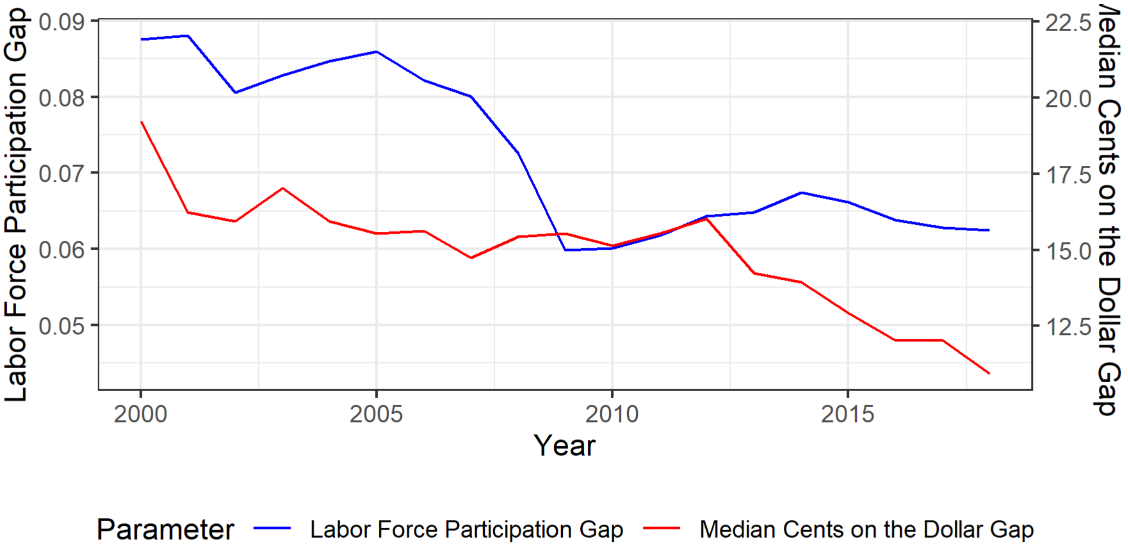

Figure 1: Labor force participation gap as well as wage gap between men and women over time.

Our project will attempt to build an individual-level model of labor force participation built on demographic, household, and opinion data from the 2018 CCES dataset. We will build classifiers modeling women's decision to participate in the labor market (and how they participate, e.g. full time or part time) on features including opinions on social issues, marital status, the presence of children, education level, and age. We build and tune classifiers using regression-adjacent techniques, ensemble techniques, and k-nearest neighbor techniques, and then use these models to (i) interrogate feature importance for a general model of labor force participation, (ii) interrogate how the impact and significance of features changes with gender (e.g., does having children affect men and women's decision to participate differently), (iii) try to build useful predictive models for labor participation (including robust models for different types of labor participation.) We hope that together this variety of machine-learning techniques will allow us new insight into labor participation, female labor participation, and the male-female labor participation gap.

1.2 Prior Work

We can broadly divide the relevant literature to date into two camps: predictive models for aggregate labor force participation built on regression analysis of population-level data, and analyses of how social and political structures affect women's labor participation that are carried out in small-scale experimental settings or through field research. We use this literature to inform features we might like to include in our analysis. We also use this as a starting point to specify a "state of the art" regression as an outside option to compare with our machine learning techniques.

1.2.1 Population-Level Predictive Models for Gender LFP Gap

In general, this literature builds causal linear models and then estimates their parameters. Grigoli, Koczan, and Topalova (2018) fit a linear regression for labor participation of different demographic groups across 23 countries since 1980. They find that secondary and tertiary education rates, public spending on childhood education and care, and the share of part-time employment in the job market have positive and statistically significant effects on female labor force participation. The paper also finds that the shape of women's age participation profile is consistent with a large share of women leaving the labor force to start their own family and, in the process, become the main caretakers of the family. We will use this paper to inform how we use the "age" feature we will include in our analysis.

Toossi (2009) fits a regression on both aggregate and individual features using time-series census data in the United States from individual data from 1970-2006 to predict labor force participation, finding statistically significant impact of marital status and education on women working. We will be using a similar dataset, but we will build on this work by using different modeling techniques and by merging the census data with additional opinion data at the geographic level.

Özerkek, Özbal, et al. (2017) fit a probit-regression model on individual data from Turkey and find that being married has a negative correlation with women participating in the labor force, while secondary education has a positive correlation. Taşseven, Altaş, and Turgut (2016) study aggregate FLFP rates across the OECD from 1980-2013 using a panel logit model and find a small and not statistically significant effect from the ratio of female to male enrollment in tertiary education, and a large and highly statistically significant effect from fertility rates on female labor force participation.

These findings square with the broad economic consensus established in Bertrand, Goldin, and Katz (2010) that maternity decisions are responsible for the vast majority of the gender wage gap that remains in the modern workplace. Furthermore, Albanesi and Olivetti (2016) use historical data to suggest that improved maternal health from 1930-1960 may have been responsible for gains in FLFP in the United States over the same period. We therefore will make sure to include maternity decisions in our analysis. Associated with maternal decisions is of course access to birth control and family planning. Goldin and Katz (2002) suggest a positive impact on women's labor participation due to the advent of oral birth control using a study of panel data.

1.2.2 Setting-Specific Evidence of Female LFP Shifters

In addition to population-level studies, economists has made valuable contributions to explaining the female labor force participation gap by looking at particular institutional dimensions of labor supply. In a seminal paper, Bertrand, Kamenica, and Pan (2015) find that the institution of marriage generally constrains a wife's earnings to below those of her husband and that even in marriages where the wife's earnings exceed those of her husband, the wife tends to do more of the household work.[1] This suggests the importance of the gendered institution of marriage in influencing female labor force decisions as a function of her marital partner's labor force decisions.

In the field of experimental economics, we see from Bursztyn, Fujiwara, and Pallais (2017) that unmarried female MBA students tend to be less ambitious compared to married MBA students in settings where their ambition might be visible to their male peers. This suggests a trade-off between success in marriage markets and success in labor markets (at least in this setting). In light of this, our analysis could be improved by including proxies for a husband's income and labor force decisions in our models of female labor force participation.

Additionally, Bursztyn, González, and Yanagizawa-Drott (2018) find that within the marriage norms in Saudi Arabia, many men privately do not mind if their wife works, but publicly oppose women's entry into the workforce. This suggests that institutional norms, and perceptions of public beliefs, can have a meaningful impact on whether women participate in the labor force. Our analysis is constrained to the United States, but we will incorporate similar public-opinion data in our analysis.

[1] Potentially, this is because wives want to offset the psychological frictions that arise when a female partner out earns her male partner.

1.3 Our Contribution

Building on the existing literature, this paper studies both conventional drivers of (female) labor force participation and proposed non-conventional drivers of female labor force participation inspired by the social sciences (e.g., opinions on feminism, abortion.) By including a wide set of features in our data and applying, tuning, and refining a variety of machine learning techniques, we hope to capture more of the complexity behind women's choice to participate in the labor market. In particular, our group aims to: (1) evaluate which data features are relevant in modeling labor force participation and the labor force participation gap (using techniques such as LASSO); (2) evaluate which classes of machine learning techniques have strong predictive power in this setting.

2 Data

For this project, we use data from the 2018 Cooperative Congressional Election Survey. This survey is a nationally representative sample of $\sim 36,000$ Americans which includes public opinion data, household and demographic data, and questions on employment status at the individual level. We construct our main variable of interest as the indicator variable for whether or not someone is employed, and we predict this based on a variety of features in the data including political opinions (about President Trump, abortion, feminists, etc.), demographics (ethnicity, rural vs. urban), and individuals' professional backgrounds (education, veteran status, etc.).

3 Empirical Strategy

We will use a variety of machine-learning techniques with the aim of predicting individual labor force participation. We will construct a set of variables which are female specific.

For example, the variable named "veteran_f" is an indicator for female veterans, and if we include the "veteran" variable in the same model, then we can attribute predictive power that comes from "veteran_f" as being a key driver of female labor force participation in a way that is different from men. Note that we only use these _f variables in models (such as lasso) where the interactions have to be pre-coded manually. In the case of the logit regression we run after feature selection, we don't include these variables, but rather we do the interaction within the formula call of the model.

After we fit our models using cross-validation to tune the relevant hyperparemeters, we will evaluate their performance on a holdout test dataset. We will use test dataset accuracy as a way of comparing the predictive power of our conventional regression models against models generated from ensemble and kNN techniques.

We will use the following techniques to address this question: Linear Regression as a baseline, LASSO Regression to evaluate important features; bagging and random forest to further evaluate feature selection; and k-nearest neighbors as a predictive model to add subtler analyses of labor participation (including a multi-class classifier) compete with the regression and ensemble techniques. We hope that by leveraging these different techniques we can identify both stronger predictive models and the key drivers of female labor force participation across models.

4 Results and Discussion

4.1 Regression-Adjacent Techniques

First, we will consider the classic workhorse of empirical social science research: the regression model. The main advantage that regression brings in our setting is that it returns coefficients which we can interpret sensibly as the magnitude of an effect that that dimension has on our outcome of interest. The main drawback however is that it performs poorly in settings where we don't have a specific model and where we have lots and lots of features that might describe a fairly complex societal problem.

It is clear that the female labor force participation gap is one of those settings. There are lots of things that have been identified as key drivers in the literature. Most prominent drivers include having children and access to education and family planning resources, but researchers have also identified more subtle opinion based drivers of this phenomenon. Because our dataset allows us to observe both of these at an individual level, we are able to make some meaningful and useful statements on this problem.

We begin by running LASSO and Ridge regression with λ values that we optimize using cross-validation on our training data. The goal of these models is to find the features which are most important to include in our regression model of the situation. The table of the top 30 features arranged according to importance in the Ridge model is reported in table 5 of the appendix. Because the tuned Lasso model includes many of the coefficients as controls, we also run lasso with lambda equal to .01 to weed out all but the most salient features to be used in future models.

Interpreting the output from this table, we see that the tuned lasso was the most accurate on the holdout data, able to classify 72.4% of this new data correctly, while the tuned ridge model was only able to classify 72.1% correctly. Now to summarize the most important coefficients. We notice that there are markedly different outcomes for people with children under the age of 18, veterans, as well as those of different education levels. We see in our models many of the effects well known to economists for a long time. The coefficient for having children under the age of 18 is 0.991 while the coefficient for having a child under the age of 18 and being female is -1.129, and the coefficient for being female is -0.823. This can be interpreted as follows: for men, having children makes the more likely to work while for women it has the opposite effect of pulling them out of the labor force. In fact, the relative magnitudes of the coefficients indicates that there is a stronger force of children pulling women out of the labor force than the unexplained difference term which we say might be something like discrimination or features that we can't observe in our dataset.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.336	0.804	-0.419	0.676
child18	1.746	0.048	36.288	0.000
veteran	-1.068	0.041	-26.076	0.000
hhinc_change	0.643	0.023	27.837	0.000
cit1	-1.603	0.799	-2.006	0.045
black	-0.073	0.073	-1.004	0.316
abortion	0.155	0.043	3.636	0.000
vote_trump	-0.534	0.046	-11.550	0.000
feminists_reasonable	-0.021	0.015	-1.420	0.155
female	-2.537	0.932	-2.723	0.006
child18:female	-1.210	0.059	-20.484	0.000
veteran:female	0.951	0.104	9.181	0.000
hhinc_change:female	-0.110	0.030	-3.629	0.000
cit1:female	1.792	0.924	1.939	0.052
black:female	0.271	0.089	3.060	0.002
abortion:female	0.229	0.057	3.994	0.000
vote_trump:female	0.202	0.063	3.222	0.001
feminists_reasonable:female	0.091	0.019	4.692	0.000

Table 1: Logit regression with most important coefficients from the Lasso and Ridge regressions. This allows Causal Interpretation.

The remainder of our results highlight the importance of race, living in a city, being educated, and interestingly being pro-choice on changing female labor force participation. The last one in particular highlights the findings of Bursztyn, González, and Yanagizawa-Drott (2018) about the importance of social norms on these decisions, and this could be some evidence of our model picking up on feminist social norms that keep women in the workplace and are associated with opinions such as being pro-choice.

Based on our feature selection in LASSO and Ridge, we come to the regression model in table 4.1, which we run as an unpenalized logit so that we can get interpretable standard errors. In particular, we are interested in the relative magnitude of opinion-based effects against more traditional predictors like education. We see in our regression output the traditional results about children increasing the labor force participation of men, but pushing women out of the labor force. We find the interesting effect that women who voted for Trump are more likely to be in the labor force than their male Trump-voting counterparts, but that both groups are less likely to be working full time. We see that women who are pro-choice are dramatically more likely to be employed and that women who think feminists are reasonable are also more likely to be employed. These effects are however broadly smaller in magnitude than traditional predictors like being a citizen. Interestingly, the most important coefficient in our regression with fewer coefficients (but interpretable standard errors) is the unexplained term for being female. This means that none of our predictors on their own can say more about female labor force participation than the unexplained portion.

4.2 Ensemble Techniques

Our goal in this section is to produce classifiers using random forest and bagging techniques. We first work to select the features we will use for both techniques, and then to tune the parameters for our models.

The techniques above allowed us to directly identify (by coefficient size) features' relative importance in our classifier. Random forests and bagging techniques will inherently put more weight on the most important features, but we have no convenient metric to judge features' relative importance in the resulting model like we can with regression. We perform a random forest and bagging based feature selection by looking at the marginal impact on validation-set accuracy scores of adding or removing a feature. This approach requires us to decide on a baseline feature set; we then train our models on that feature set, and then add or remove features one at a time and repeat, looking at the change in accuracy vs. our baseline.

For our purposes, this baseline set is the set of important features identified in the regression section. We will train four different models on these features: bagging with $n = 10$, bagging with $n = 50$, random forest with $n = 10$, and random forest with $n = 50$ (where n is the # of estimators in our model.) We first compute the validation-set accuracy scores for each model with our baseline feature set, and then remove features one-by-one, retrain our four models, and compare the resulting change in accuracy $S - S_k$ where S is the validation-set accuracy score for the baseline model and S_k the validation-set accuracy score for the model with feature k removed. Using this method, we can eliminate any features that are unimportant to our ensemble models (or indeed detract it, as in the case of overfitting.) We then take the trimmed feature list as our "baseline" set for the next stage of our analysis (evaluating the marginal impact of added features.)

The features which led to the smallest loss (or largest gain) in accuracy when excluded are summarized in table 5. Indeed, observe that no feature's exclusion improved the predictive power of all four of the classifiers we trained. The improvements in accuracy that we see are small and scattered; we conclude that there is no benefit, and likely some cost, in excluding them from our models moving forward. So our baseline for the next stage of feature selection remains unchanged. With this in mind, in table 6 below, we can see the impact on our models' accuracy scores with each additional feature. Note that ensemble techniques with higher n values are more prone to overfitting, and therefore tend to see a lower bump in accuracy for all features than lower n equivalents. With this mind, we find that birthyr, faminc new, educ, faminc new_f, approve trump, approve_trump_f, abortion, abortion f, vote trump, vote trump f all led to a notable improvement in our accuracy scores on the validation set across models, including for higher values of n. In adding these features to our model with an eye to the validation accuracy scores we hope to strike a reasonable balance between power of ensemble techniques in automatically filtering important features and the bias that can result.

We now turn our attention to tuning our models. In table 7 below, we assess the validation-set accuracy scores for random forest and bagging classifiers for $10 \leq n \leq 250$. We may observe that there is very little change in the validation scores over this range of values; we therefore pick, as a reasonably robust and cost-effective choice, $n = 100$ for our random forest and bagging models. These models each have an accuracy score of roughly 74.5%, which we can observe is a modest improvement on the results from regression-adjacent techniques above. Having thus optimized our models, we are now in a position to interrogate relative feature importance in models using ensemble techniques.

Table 2: Feature Importances

Feature	Importance (BC)
birthyr	0.413300
faminc_new	0.175765
educ	0.097484
hhinc_change	0.050834
faminc_new_f	0.049084
approve_trump	0.047750
hhinc_change_f	0.032122
abortion	0.024577
approve_trump_f	0.023365
child18	0.017693
veteran	0.015729
black	0.012110
abortion_f	0.010551
female	0.010533
child18_f	0.008050
black_f	0.005857
veteran_f	0.002876
cit1_f	0.001851
cit1	0.000469

Feature	Importance (RFC)
birthyr	0.470122
faminc_new	0.157027
educ	0.100099
faminc_new_f	0.053182
hhinc_change	0.050255
approve_trump	0.036265
child18	0.026642
hhinc_change_f	0.020519
approve_trump_f	0.018975
veteran	0.014795
abortion	0.014485
black	0.008360
female	0.006952
abortion_f	0.006007
child18_f	0.005828
black_f	0.004205
cit1_f	0.003821
veteran_f	0.002034
cit1	0.000426

These statistics allow us to examine the relative impact of all features and are summarized in table 2 below. The highlighted terms are those that did not appear in our feature list from the regression analysis above. As a first observation (and sanity check), non-gendered factors–like age, income (faminc_new), and education–are the most important in our model. This aligns with the standard results for most LFP models. In terms of opinion data, there is a strong association in our model between sociopolitical opinions and labor force participation–in particular, our results here support the findings in the regression analyses above that abortion and approval of President Trump (in our case, abortion and supporting Trump, in the regression case voting for Trump in 2016) can be linked to LFP.

participatinWe can also note that for all of the features listed as among the most important above, the female-linked version shows up as well–suggesting that some factors play either a more significant role or a different role entirely for women than for men. Finally, we find a significant impact as well from the unexplained female term, as in the regression models above, although in our case the relative importance is a good deal smaller (it was the most important factor in the regression analysis, and far from the most important in the RFC/bagging analysis.) It is not possible to rigorously determine from feature importance the contribution of each feature to our response–in this case, its impact on female labor force participation. But we can venture a reasonable guess that they align with the findings in the regression section above in terms of impact (positive or negative) on the likelihood of participation in the labor market. As such, the importance of *both* the child18 and the child18f terms in our explanatory model (which tells us that child18 (via child18f) has a different association with women's working than with men's, though we cannot identify that difference directly), paired with the fact that both terms appeared as coefficients in the regression model with opposite sign, leads us to see the results of our random forest and bagging techniques as strongly supportive of the hypothesis given above, that a child in the house makes men more likely to work and women less likely to work.

4.3 KNN

Our kNN analysis will mostly be an evaluation of whether this machine learning technique has a strong predictive power in this setting. If kNN can provide good results, then it suggests that the classes are quite separable; if KNN fails, then it would indicate that the metric vector we have chosen does not produce separable classes.

In our first kNN model, we include 9 different employment status options [2] for prediction. When we are performing kNN, there is a trade-off between using more points to get a more general model and losing nuance when we have too many observations being averaged to make the prediction. Therefore, we fit our model in table 3 for values of k ranging from 3 to 9. Having a small *k* could cause high variance and low bias, whereas having a very large *k* could cause low variance and high bias, sow we want to find the optimal *k* that balances this bias v. variance tradeoff. We see that our holdout set accuracy doesn't decline as *k* rises in this setting, and that we can achieve remarkably accurate predictions on such a high number of classes as observed in the table.

K	Fem. 9 Cl. Acc.	Fem. 2 Cl. Acc.	All 2 Cl. Acc.
3	0.512	0.711	0.740
4	0.527	0.709	0.739
5	0.537	0.716	0.748
6	0.541	0.720	0.751
7	0.547	0.727	0.759
8	0.554	0.724	0.757
9	0.556	0.728	0.761

Table 3: kNN Results

The 9 class shows shows the unique power of kNN in this setting to make more granular predictions than regression, but we also want to perform an apples-to-apples comparison between all our models to be able to determine which is most accurate in this setting.

[2] Legend: 1 = Full Time, 2 = Part-Time, 3 = Temporarily Laid Off, 4 = Unemployed, 5 = Retired, 6 = Permanently Disabled, 7 = Homemaker, 8 = Student, 9 = Other

Therefore, we perform the same two class classification problem as above in order to compare accuracy. Here the two classes are full-time employed and not in the labor force as a full-time worker. Again looking to table 3, we see that the 2 class accuracy is substantially higher than our 9 class accuracy. Additionally, it slightly edges out the accuracy of either the random forest models or the regression models. This suggests that kNN we much more effectively able to capture the non-linearity of the problem and fit the complex relationships that drive decisions based on culture and opinions in addition to economic factors.

5 Conclusion

First, we will compare our models' predictive accuracy. The accuracy of the best regression model is 72.4% for LASSO and Ridge is 72.1%. The best random forest model was 74.5% and the best bagging classifier had a test accuracy 74.6%. The best kNN model was 76.1% accurate on the 2 class equivalent model[3]. The 2 class model is most equivalent to what wad done in the regression and random forest models. Thus we see that kNN is dramatically more accurate in this setting. That said, it is not possible to extract why in particular kNN is performing better.

Feature selection revealed age, family income, education, veteran status, race, and citizenship to be significant non-gendered features in predicting labor participation. Our models also verified the disparity between child18 and child18_f. Both factors were significant using both the regression and ensemble techniques models, and yet–according to our regression results–had opposite significance: men with a child under 18 in the house were more likely to work; women with a child under 18 in the house were less likely to work. This result is generally consistent with the finding of Bertrand, Goldin, and Katz (2010) that maternity

[3] A prior presentation of these results had a much higher accuracy for the kNN model. This was due to a bug in the code that led to the categories being used as predictors. We remedied this bug, but it naturally changed the results that are presented in this paper. This footnote aims to clear up any confusion regarding the change in results and show our commitment to producing reproducible and accurate research.

decisions play an important role in wage and labor gaps. Finally, we found that gender-adjacent opinion data–such as a negative opinion of President Trump or a positive opinion of abortion–were associated with female labor force participation. To our knowledge this finding is novel in the existing literature and hints at the social and societal factors (beyond basic economic principles like wages, education, etc.) associated with women's participation or non-participation in the workforce.

More generally, we feel that our project was useful in bringing to bear a variety of ML methods not usually applied to this class of problems in economics. The improved accuracy of kNN may be valuable for purposes where we want to predict labor force participation as opinions and demographics change over time. By combining regression and ensemble techniques we were able to get a more nuanced portrait of the features at play in female labor force participation and the interaction between cultural factors and structural factors. Our most important conclusion remains the measured magnitudes and directions of different factors influencing the female labor force participation gap that we were able to measure after feature selection in the regression portion of our paper.

References

Albanesi, Stefania, and Claudia Olivetti. 2016. "Gender roles and medical progress." *Journal of Political Economy* 124 (3): 650–695.

Bertrand, Marianne, Claudia Goldin, and Lawrence F Katz. 2010. "Dynamics of the gender gap for young professionals in the financial and corporate sectors." *American economic journal: applied economics* 2 (3): 228–55.

Bertrand, Marianne, Emir Kamenica, and Jessica Pan. 2015. "Gender identity and relative income within households." *The Quarterly Journal of Economics* 130 (2): 571–614.

Bursztyn, Leonardo, Thomas Fujiwara, and Amanda Pallais. 2017. "'Acting Wife': Marriage Market Incentives and Labor Market Investments." *American Economic Review* 107 (11): 3288–3319.

Bursztyn, Leonardo, Alessandra L González, and David Yanagizawa-Drott. 2018. *Misperceived social norms: Female labor force participation in Saudi Arabia.* Technical report. National Bureau of Economic Research.

Goldin, Claudia, and Lawrence F Katz. 2002. "The power of the pill: Oral contraceptives and women's career and marriage decisions." *Journal of political Economy* 110 (4): 730–770.

Grigoli, Francesco, Zsoka Koczan, and Petia Topalova. 2018. *A Cohort-Based Analysis of Labor Force Participation for Advanced Economies.* International Monetary Fund.

Özerkek, Yasemin, Yasemin Özbal, et al. 2017. "The Effects of Education and Marital Status on Women's Labor Force Participation: A Regional Analysis of Turkey." *Ekonomi-tek-International Economics Journal* 6 (3): 15–38.

Taşseven, Özlem, Dilek Altaş, and ÜN Turgut. 2016. “The determinants of female labor force participation for OECD countries.” *Uluslararası Ekonomik Araştırmalar Dergisi* 2 (2): 27–38.

Toossi, Mitra. 2009. “Employment outlook: 2008-18-labor force projections to 2018: older workers staying more active.” *Monthly Lab. Rev.* 132:30.

WBG. *World Bank Data: Ratio of female to male labor force participation rate (%) (modeled ILO estimate).* `https://data.worldbank.org/indicator/SL.TLF.CACT.FM.ZS`. Accessed: 2020-01-23.

Figures Appendix

Extra Regression Tables

Variable	Lasso $\lambda = 0.00035$	Lasso $\lambda = 0.01$	Ridge $\lambda = 0.01947$
Accuracy	0.724	0.711	0.721
immigrant_noncitizen_f	-0.84	0	-1.067
child18	0.991	0.52	0.741
child18_f	-1.129	-0.601	-0.724
veteran	-0.574	-0.249	-0.538
veteran_f	0.636	0	0.524
hhinc_change	0.439	0.288	0.36
cit1	-0.4	0	-0.312
immigrant_noncitizen	0.112	0	0.308
black_f	0.407	0.152	0.293
votereg	0.311	0	0.276
investor	0.209	0.217	0.23
internethome	-0.264	0	-0.226
female_f	0	0	-0.186
city	0.16	0.116	0.175
female	-0.823	-0.246	-0.173
cit1_f	0	0	-0.157
catholic_f	0.199	0	0.156
educ	0.161	0.124	0.156
faminc_new	0.193	0.156	0.151
mil_fam_f	-0.127	0	-0.144
jewish	-0.177	0	-0.143
city_f	0.199	0	0.135
investor_f	0.195	0	0.13
first_gen_f	-0.124	0	-0.128
black	0.051	0.045	0.112
vote_trump_f	0.19	0	0.091
daily_prayer_f	-0.091	0	-0.09
mil_fam	0.07	0	0.084
abortion	0.048	0.001	0.081
suburb	0.033	0	0.076

Table 4: Results from LASSO and Ridge Regression. Features are ordered by magnitude of Ridge coefficient.

Extra Ensemble Methods Tables

Table 5: Marginal impact from removing features (Here, Adj. Score is $\alpha - \alpha_0$, where α_0 is 0-1 loss for control (baseline feature set) and α is 0-1 loss for model.)

Variable	Adj. Testing Score RFC (n=10)	Adj. Testing Score RFC (n=50)	Adj. Testing Score BC (n=10)	Adj. Testing Score BC (n=50)
black	-0.000550	0.000137	-0.000687	0.000137
child18_f	-0.000275	0.000000	-0.000137	0.000000
veteran_f	-0.000275	0.000000	0.000412	-0.000275
control	0.000000	0.000000	0.000000	0.000000
female	0.000000	0.000275	0.000550	0.000000
black_f	0.000000	0.000687	0.001375	-0.000275
cit1_f	0.000137	0.000000	0.000962	0.000000
hhinc_change_f	0.000137	0.000275	0.000275	0.000000
cit1	0.000137	0.000275	0.000412	0.000275
veteran	0.010172	0.010172	0.010584	0.011684

Table 6: Marginal impact ($\alpha - \alpha_0$) from adding features

Variable	Adj. Testing Score RFC (n=10)	Adj. Testing Score RFC (n=50)	Adj. Testing Score BC (n=10)	Adj. Testing Score BC (n=50)
birthyr	-0.068316	-0.066254	-0.069278	-0.067491
faminc new	-0.033540	-0.037388	-0.032715	-0.038076
educ	-0.033540	-0.033265	-0.031753	-0.033677
faminc new f	-0.019931	-0.021718	-0.020619	-0.020481
approve trump	-0.016495	-0.018832	-0.019381	-0.018419
approve trump f	-0.015533	-0.017320	-0.016357	-0.016357
abortion	-0.014845	-0.016082	-0.015533	-0.015808
abortion f	-0.014433	-0.015945	-0.016220	-0.017045
vote trump	-0.013196	-0.015533	-0.011684	-0.015533
vote trump f	-0.013471	-0.015395	-0.014296	-0.015395

Table 7: Testing and training α scores for # of estimators n

n	Training score RFC	Testing score RFC	Training score BC	Testing score BC
10	0.045756	0.260893	0.044438	0.258969
20	0.034898	0.256082	0.035279	0.256357
30	0.032088	0.254158	0.032261	0.253883
40	0.031117	0.258694	0.031325	0.255945
50	0.030492	0.252234	0.030596	0.256082
60	0.030353	0.253746	0.030215	0.253471
70	0.030180	0.252921	0.030111	0.253471
80	0.030180	0.252784	0.030076	0.256632
90	0.030007	0.252509	0.030041	0.253058
100	0.030076	0.255120	0.030041	0.254845
110	0.030007	0.254708	0.030076	0.257595
120	0.030007	0.254570	0.030007	0.255808
130	0.030007	0.257320	0.030007	0.253058
140	0.030007	0.254021	0.030007	0.256220
150	0.030007	0.253333	0.030007	0.254158
160	0.030007	0.254158	0.030007	0.256220
170	0.030007	0.252646	0.030007	0.255533
180	0.030007	0.254021	0.030007	0.252096
190	0.030007	0.254570	0.030007	0.254158
200	0.030007	0.253608	0.030007	0.253196
210	0.030007	0.252646	0.030007	0.254158
220	0.030007	0.254021	0.030007	0.253883
230	0.030007	0.253196	0.030007	0.254983
240	0.030007	0.254845	0.030007	0.253883

Code Appendix

Data Cleaning Code

```r
# Cleaning CCES data
# File Started on Feb 28 by Eric Karsten ekarsten@uchicago.edu

# Libraries
library(tidyverse)
library(readstata13)

# Data Directory Setup
root <- getwd()
while(basename(root) != "macss_project") {
  root = dirname(root)
}
source(file.path(root, "data.R"))


files <- list.files(file.path(ddir, "CCES RAW"), pattern = ".dta", full.names = T)

df <- files %>%
  read.dta13() %>%
  as_tibble()

df <- df %>%
  select(c("birthyr", "gender", "employ", "educ", "votereg", "race", "hispanic",
           "internethome", "marstat", "pid3", relig_important = pew_religimp,
           "pew_churatd", "pew_prayer", "religpew", "child18",
           "faminc_new", "ownhome", "urbancity", current_mil = milstat_1,
           mil_fam = milstat_2, veteran = milstat_3, "immstat", "cit1",
           "investor", national_economy = CC18_301, hhinc_change = CC18_
   302,
           approve_trump = CC18_308a, approve_congress = CC18_308b,
           approve_scotus = CC18_308c, vote_2016 = CC18_317, abortion =
   CC18_321a,
          racial_sentiment = CC18_422e, racial_sentiment2 = CC18_422h,
          racial_sentiment3 = CC18_422b,
           women_complain = CC18_422c, feminists_reasonable = CC18_422d))

df <- df %>%
  mutate(female = gender - 1,
         black = if_else(race == 2, 1, 0),
         asian = if_else(race == 4, 1, 0),
         hispanic = 2 - hispanic,
         internethome = if_else(internethome <= 2, 1, 0),
         democrat = if_else(pid3 == 1, 1, 0),
         republican = if_else(pid3 == 2, 1, 0),
         relig_important = 5 - relig_important,
         weekly_church = if_else(pew_churatd <= 2, 1, 0),
         daily_prayer = if_else(pew_prayer <= 2, 1, 0),
         protestant = if_else(religpew == 1, 1, 0),
         catholic = if_else(religpew == 2, 1, 0),
         jewish = if_else(religpew == 5, 1, 0),
         child18 = 2 - child18,
         ownhome = if_else(ownhome == 1, 1, 0),
         city = if_else(urbancity == 1, 1, 0),
         suburb = if_else(urbancity == 2, 1, 0),
         town = if_else(urbancity == 3, 1, 0),
         current_mil = 2 - current_mil,
         mil_fam = 2 - mil_fam,
         veteran = 2 - veteran,
         immigrant_citizen = if_else(immstat == 1, 1, 0),
         immigrant_noncitizen = if_else(immstat == 2, 1, 0),
         first_gen = if_else(immstat == 3, 1, 0),
         cit1 = 2 - cit1,
         investor = 2 - investor,
         national_economy = 6 - national_economy,
         hhinc_change = 6 - hhinc_change,
         approve_trump =  5 - approve_trump,
         approve_congress = 5 - approve_congress,
         approve_scotus = 5 - approve_scotus,
         vote_trump = if_else(vote_2016 == 1, 1, 0),
         vote_clinton = if_else(vote_2016 == 2, 1, 0),
         abortion = 2 - abortion,
         give_blacks_no_govt_help = 6 - racial_sentiment,
         blacks_lazy = 6 - racial_sentiment2,
         racism_is_isolated = 6 - racial_sentiment3,
         women_complain = 6 - women_complain,
         feminists_reasonable = 6 - feminists_reasonable,
         votereg = if_else(votereg == 1, 1, 0)) %>%
  select(-immstat, -gender, -vote_2016, -racial_sentiment, -racial_sentiment2,
         -racial_sentiment3, - race, -pid3, -pew_churatd, -pew_prayer, -religpew,
         -urbancity) %>%
  filter(faminc_new <= 16)

f_vars <- colnames(df)[7:length(df)]

```

```r
df_f = df[f_vars] %>% mutate_all(function(c) { if_else(df$female == 1, c, 0) })

colnames(df_f) = str_c(f_vars, "_f")

df = bind_cols(df, df_f) %>% drop_na()



write_csv(df, path = file.path(ddir, "cces_clean.csv"))
```

Regression Code

```r
# Lasso and Ridge regression
# File Started on Feb 29 by Eric Karsten ekarsten@uchicago.edu

# Libraries
library(glmnet)
library(rsample)
library(tidyverse)

# Data Directory Setup
root <- getwd()
while(basename(root) != "macss_project") {
  root = dirname(root)
}
source(file.path(root, "data.R"))

# data
df <- read_csv(file.path(ddir, "cces_clean.csv")) %>%
  mutate(employ = if_else(employ == 1, 1, 0)) %>%
  filter(current_mil == 0)
```

```


set.seed(1776)

# Split data
split = initial_split(df, .8)
train = training(split)
test = testing(split)

y = train$employ
X = as.matrix(train[,-2])

## Lasso
las <- cv.glmnet(X, y, family = "binomial", nfold = 10, alpha = 1)

# Fit the optimal model and inspect
lassomod <- glmnet(X, y, family = "binomial", lambda = las$lambda.min, alpha = 1)
lassomod2 <- glmnet(X, y, family = "binomial", lambda = .01, alpha = 1)


## Ridge
rid <- cv.glmnet(X, y, family = "binomial", nfold = 10, alpha = 0)

# Fit the optimal model and inspect
ridmod <- glmnet(X, y, family = "binomial", lambda = rid$lambda.min, alpha = 0)

accuracy <- function(m) {
  pred = predict.glmnet(m, as.matrix(test[,-2]))
  pred = if_else(pred < .5, 0, 1)
  mean(pred == test$employ)
}

process_mod <- function(m) {
  a <- as.matrix(m$beta)
  t <- tibble(Variable = rownames(a), Col2 = a[,1])
  colnames(t)[2] <- paste0(if_else(m$call$alpha == 1, "Lasso ", "Ridge "), "lambda = ", round(m$lambda, 5))
  return(t)
}

output <- list(lassomod, lassomod2, ridmod) %>%
  map(process_mod) %>%
  reduce(full_join) %>%
  arrange_at(4, function(x) { -abs(x)}) %>%
  head(30) %>%
  mutate_at(2:4, function(x) { round(x,3)})

output = rbind(c("Accuracy", accuracy(lassomod), accuracy(lassomod2), accuracy(ridmod)), output)

mod = glm(employ ~ (child18 + veteran + hhinc_change + cit1 + black + abortion + vote_trump + feminists_reasonable)*female, data = df, family = "binomial")


xtable::xtable(summary(mod)$coefficients, digits = 3)
print(xtable::xtable(output), include.rownames=FALSE, digits = 3)
```

Random Forest and Bagging Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
from sklearn import ensemble
from sklearn import linear_model
from sklearn import model_selection
from sklearn import svm
from sklearn import metrics
from scipy import optimize

np.random.seed(33002)

df=pd.read_csv(’/Users/emax/Downloads/cces_clean.csv’)

#Collapse employment into full employment (0) and other (1)
df[’employ’]=df.apply(lambda row: int(row[’employ’]==1), axis=1)

#Drop current military per BLS approach
df=df[df.current_mil==0]

columns=list(df.columns)

good_cols=[’child18’,
           ’veteran’,
           ’hhinc_change’,
           ’cit1’,
           ’black’,
           ’female’,
           ’child18_f’,
           ’veteran_f’,
           ’hhinc_change_f’,
           'cit1_f',
           'black_f']

#Split dataset
msk = np.random.rand(len(df)) < 0.8
df_train=df[msk]
df_test=df[~msk]


score_train_rfc10=[]
score_test_rfc10=[]

score_train_rfc50=[]
score_test_rfc50=[]

score_train_btc_10=[]
score_test_btc_10=[]

score_train_btc_50=[]
score_test_btc_50=[]

col_taken=['control']

#Set up testing and training sets
x_train=df_train[good_cols]
y_train=np.array(df_train[['employ']]).reshape(len(x_train),)

x_test=df_test[good_cols]
y_test=np.array(df_test[['employ']]).reshape(len(x_test),)

#Train control
```

```python
rdmf=ensemble.RandomForestClassifier(n_estimators=10).fit(x_train,y_train)

score_train_rfc10.append(float(metrics.zero_one_loss(y_train, rdmf.predict(x_train))))
score_test_rfc10.append(float(metrics.zero_one_loss(y_test, rdmf.predict(x_test))))

rdmf=ensemble.RandomForestClassifier(n_estimators=50).fit(x_train,y_train)

score_train_rfc50.append(float(metrics.zero_one_loss(y_train, rdmf.predict(x_train))))
score_test_rfc50.append(float(metrics.zero_one_loss(y_test, rdmf.predict(x_test))))

btc=ensemble.BaggingClassifier(n_estimators=10).fit(x_train,y_train)

score_train_btc_10.append(float(metrics.zero_one_loss(y_train, btc.predict(x_train))))
score_test_btc_10.append(float(metrics.zero_one_loss(y_test, btc.predict(x_test))))

btc=ensemble.BaggingClassifier(n_estimators=50).fit(x_train,y_train)

score_train_btc_50.append(float(metrics.zero_one_loss(y_train, btc.predict(x_train))))
score_test_btc_50.append(float(metrics.zero_one_loss(y_test, btc.predict(x_test))))


for i in range(len(good_cols)):
    '''
```

```
 Remove columns one by one and compute resulting scores
 '''
 x_train=df_train[good_cols[:i]+good_cols[i+1:]]
 y_train=np.array(df_train[['employ']]).reshape(len(x_train),)

 x_test=df_test[good_cols[:i]+good_cols[i+1:]]
 y_test=np.array(df_test[['employ']]).reshape(len(x_test),)

 rdmf=ensemble.RandomForestClassifier(n_estimators=10).fit(x_train,
y_train)

 score_train_rfc10.append(float(metrics.zero_one_loss(y_train, rdmf.
predict(x_train))))
 score_test_rfc10.append(float(metrics.zero_one_loss(y_test, rdmf.
predict(x_test))))

 rdmf=ensemble.RandomForestClassifier(n_estimators=50).fit(x_train,
y_train)

 score_train_rfc50.append(float(metrics.zero_one_loss(y_train, rdmf.
predict(x_train))))
 score_test_rfc50.append(float(metrics.zero_one_loss(y_test, rdmf.
predict(x_test))))

 btc=ensemble.BaggingClassifier(n_estimators=10).fit(x_train,y_train)

 score_train_btc_10.append(float(metrics.zero_one_loss(y_train, btc.
predict(x_train))))
 score_test_btc_10.append(float(metrics.zero_one_loss(y_test, btc.
predict(x_test))))

```

```
    btc=ensemble.BaggingClassifier(n_estimators=50).fit(x_train,y_train)

    score_train_btc_50.append(float(metrics.zero_one_loss(y_train, btc.predict(x_train))))
    score_test_btc_50.append(float(metrics.zero_one_loss(y_test, btc.predict(x_test))))

    col_taken.append(good_cols[i])


taken_col_df=pd.DataFrame({'Variable':col_taken,
            'Adj. Testing Score RFC (n=10)':score_test_rfc10,
              'Adj. Testing Score RFC (n=50)': score_test_rfc50,
              'Adj. Testing Score BC (n=10)': score_test_btc_10,
              'Adj. Testing Score BC (n=50)': score_test_btc_50
             })

score_train_rfc10=[]
score_test_rfc10=[]

score_train_rfc50=[]
score_test_rfc50=[]

score_train_btc_10=[]
score_test_btc_10=[]

score_train_btc_50=[]
score_test_btc_50=[]

col_added=['control']

```

```
#Set up testing and training sets
x_train=df_train[good_cols]
y_train=np.array(df_train[['employ']]).reshape(len(x_train),)

x_test=df_test[good_cols]
y_test=np.array(df_test[['employ']]).reshape(len(x_test),)

#Train control
rdmf=ensemble.RandomForestClassifier(n_estimators=10).fit(x_train,y_train)

score_train_rfc10.append(float(metrics.zero_one_loss(y_train, rdmf.predict(x_train))))
score_test_rfc10.append(float(metrics.zero_one_loss(y_test, rdmf.predict(x_test))))

rdmf=ensemble.RandomForestClassifier(n_estimators=50).fit(x_train,y_train)

score_train_rfc50.append(float(metrics.zero_one_loss(y_train, rdmf.predict(x_train))))
score_test_rfc50.append(float(metrics.zero_one_loss(y_test, rdmf.predict(x_test))))

btc=ensemble.BaggingClassifier(n_estimators=10).fit(x_train,y_train)

score_train_btc_10.append(float(metrics.zero_one_loss(y_train, btc.predict(x_train))))
score_test_btc_10.append(float(metrics.zero_one_loss(y_test, btc.predict(x_test))))

btc=ensemble.BaggingClassifier(n_estimators=50).fit(x_train,y_train)

```

```
score_train_btc_50.append(float(metrics.zero_one_loss(y_train, btc.predict(x_train))))
score_test_btc_50.append(float(metrics.zero_one_loss(y_test, btc.predict(x_test))))

for col in [col for col in columns if col not in good_cols]:
    ’’’
    Add columns one by one and report resulting scores.
    ’’’
    x_train=df_train[good_cols+[col]]
    y_train=np.array(df_train[[’employ’]]).reshape(len(x_train),)

    x_test=df_test[good_cols+[col]]
    y_test=np.array(df_test[[’employ’]]).reshape(len(x_test),)

    rdmf=ensemble.RandomForestClassifier(n_estimators=10).fit(x_train, y_train)

    score_train_rfc10.append(float(metrics.zero_one_loss(y_train, rdmf.predict(x_train))))
    score_test_rfc10.append(float(metrics.zero_one_loss(y_test, rdmf.predict(x_test))))

    rdmf=ensemble.RandomForestClassifier(n_estimators=50).fit(x_train, y_train)

    score_train_rfc50.append(float(metrics.zero_one_loss(y_train, rdmf.predict(x_train))))
    score_test_rfc50.append(float(metrics.zero_one_loss(y_test, rdmf.predict(x_test))))

    btc=ensemble.BaggingClassifier(n_estimators=10).fit(x_train,y_train)

    score_train_btc_10.append(float(metrics.zero_one_loss(y_train, btc.predict(x_train))))
    score_test_btc_10.append(float(metrics.zero_one_loss(y_test, btc.predict(x_test))))

    btc=ensemble.BaggingClassifier(n_estimators=50).fit(x_train,y_train)

    score_train_btc_50.append(float(metrics.zero_one_loss(y_train, btc.predict(x_train))))
    score_test_btc_50.append(float(metrics.zero_one_loss(y_test, btc.predict(x_test))))

    col_added.append(col)

added_col_df=pd.DataFrame({'Variable':col_added,
            'Adj. Testing Score RFC (n=10)':score_test_rfc10,
              'Adj. Testing Score RFC (n=50)': score_test_rfc50,
              'Adj. Testing Score BC (n=10)': score_test_btc_10,
              'Adj. Testing Score BC (n=50)': score_test_btc_50
             })

for col in [col for col in list(taken_col_df.columns) if col[:4]=="Adj."]:
    #Compute change in accuracy
    taken_col_df[col]=taken_col_df[col]-taken_col_df.iloc[0][col]

for col in [col for col in list(added_col_df.columns) if col[:4]=="Adj."]:
    #Compute change in accuracy
    added_col_df[col]=added_col_df[col]-added_col_df.iloc[0][col]

```

```
taken_col_df.to_latex(’table_adj_taken’, index=False)
added_col_df.to_latex(’table_adj_added’, index=False)



#By inspection, removed features that improved accuracy when
#removed and added features that improved accuracy when
#added to feature list
good_cols=[’child18’,’veteran’,’hhinc_change’,’cit1’,’black’,
           ’female’,’child18_f’,’veteran_f’,’hhinc_change_f’,
           ’cit1_f’, ’black_f’, ’birthyr’, ’faminc_new’, ’educ’,
           ’faminc_new_f’,’approve_trump’, ’approve_trump_f’,
           ’abortion’, ’abortion_f’]


#Tune based on training and testing accuracy scores
x_train=df_train[good_cols]
y_train=np.array(df_train[[’employ’]]).reshape(len(x_train),)

x_test=df_test[good_cols]
y_test=np.array(df_test[[’employ’]]).reshape(len(x_test),)

def get_RFC_scores(k):
    rdmf=ensemble.RandomForestClassifier(n_estimators=k).fit(x_train, y_train)
    return float(metrics.zero_one_loss(y_train, rdmf.predict(x_train))), float(metrics.zero_one_loss(y_test, rdmf.predict(x_test)))
def get_BC_scores(k):
    bc=ensemble.RandomForestClassifier(n_estimators=k).fit(x_train,y_train)
    return float(metrics.zero_one_loss(y_train, bc.predict(x_train))),
   float(metrics.zero_one_loss(y_test, bc.predict(x_test)))

rf_scores=[get_RFC_scores(i*10) for i in range(1,25)]
bc_scores=[get_RFC_scoreS(i*10) for i in range(1,25)]

tuning_df=pd.DataFrame(
    {'Training score (RFC)':[scores[0] for scores in rf_scores],
     'Testing score (RFC)':[scores[1] for scores in rf_scores],
     'Training score (BC)': [scores[0] for scores in bc_scores],
     'Testing score (BC)': [scores[1] for scores in bc_scores]
    })
tuning_df.to_latex(tuning_df, index=False)


#Get feature importances for tuned models
bc=ensemble.BaggingClassifier(n_estimators=100)
bc.fit(x_train,y_train)

bc_feature_importances = np.mean([
    tree.feature_importances_ for tree in bc.estimators_
], axis=0)

rfc=ensemble.RandomForestClassifier(n_estimators=100)
rfc.fit(x_train,y_train)

importances_df=pd.DataFrame({'Importance (RFC)': rfc.feature_importances_,
    'Importance (BC)': bc_feature_importances},
                      index = good_cols).sort_values('Importance (RFC)')
importances_df.to_latex('importances_df', index=False)
```

kNN Code

```r
library(tidyverse)
library(class)
library(rsample)
library(caret)
library(xtable)

# Data Directory Setup
root <- getwd()
while(basename(root) != "macss_project") {
  root = dirname(root)
}
source(file.path(root, "data.R"))

set.seed(1234)

# data
df <- read_csv(file.path(ddir, "cces_clean.csv")) %>%
  mutate(employ2 = if_else(employ == 1, 1, 0)) %>%
  filter(current_mil == 0)

# Split data
split = initial_split(df, .8)
train = training(split)
test = testing(split)

y = train$employ
y2 = train$employ2
X = as.matrix(train %>% select(-employ, -employ2))

y_test = test$employ
```

```r
y2_test = test$employ2
X_test = as.matrix(test %>% select(-employ, -employ2))

yf <- y[train$female == 1]
y2f <- y2[train$female == 1]
Xf = as.matrix(train %>% select(-employ, -employ2) %>% filter(female == 1))

yf_test <- y_test[test$female == 1]
y2f_test <- y2_test[test$female == 1]
Xf_test = as.matrix(test %>% select(-employ, -employ2) %>% filter(female == 1))

make_fem_data_knn_9class <- function(K) {
  # Run kNN Analysis with k = 9
  pred_data_fem_9 <- knn(train = Xf, test = Xf_test, cl = yf, k = K)

  ##create confusion matrix for 2 classes
  conf_matrix_fem_9 <- table(pred_data_fem_9, yf_test)
  matrix_res_fem_9 <- confusionMatrix(conf_matrix_fem_9)
  matrix_res_fem_9[3]$overall[1]
}

make_fem_data_knn <- function(K) {
  # Run kNN Analysis with k = 9
  pred_data_fem_2 <- knn(train = Xf, test = Xf_test, cl = y2f, k = K)

  ##create confusion matrix for 2 classes
  conf_matrix_fem_2 <- table(pred_data_fem_2, y2f_test)
  matrix_res_fem_2 <- confusionMatrix(conf_matrix_fem_2)
  matrix_res_fem_2[3]$overall[1]
}

make_full_data_kkn <- function(K) {
  # Run kNN Analysis with k = 9
  pred_data_all <- knn(train = X, test = X_test, cl = y2, k = K)

  ##create confusion matrix for 2 classes
  conf_matrix_all <- table(pred_data_all, y2_test)
  matrix_res_all <- confusionMatrix(conf_matrix_all)
  matrix_res_all[3]$overall[1]
}

# Female data accuracy for k= 3 to 9 for 9 classes
fem_9class_acc <- map(3:9, make_fem_data_knn_9class)

# Female data accuracy for k= 3 to 9 for 2 classes
fem_2class_acc <- map(3:9, make_fem_data_knn)

# Full data accuracy for k= 3 to 9
all_2class_acc <- map(3:9, make_full_data_kkn)


my_table <- tibble(
  K = as.integer(3:9),
  "Fem. 9 Cl. Acc." = unlist(fem_9class_acc),
  "Fem. 2 Cl. Acc." = unlist(fem_2class_acc),
  "All 2 Cl. Acc." = unlist(all_2class_acc)) %>%
  mutate(K = as.integer(K))

my_table %>% xtable(digits=3) %>% print(include.rownames = F)
```