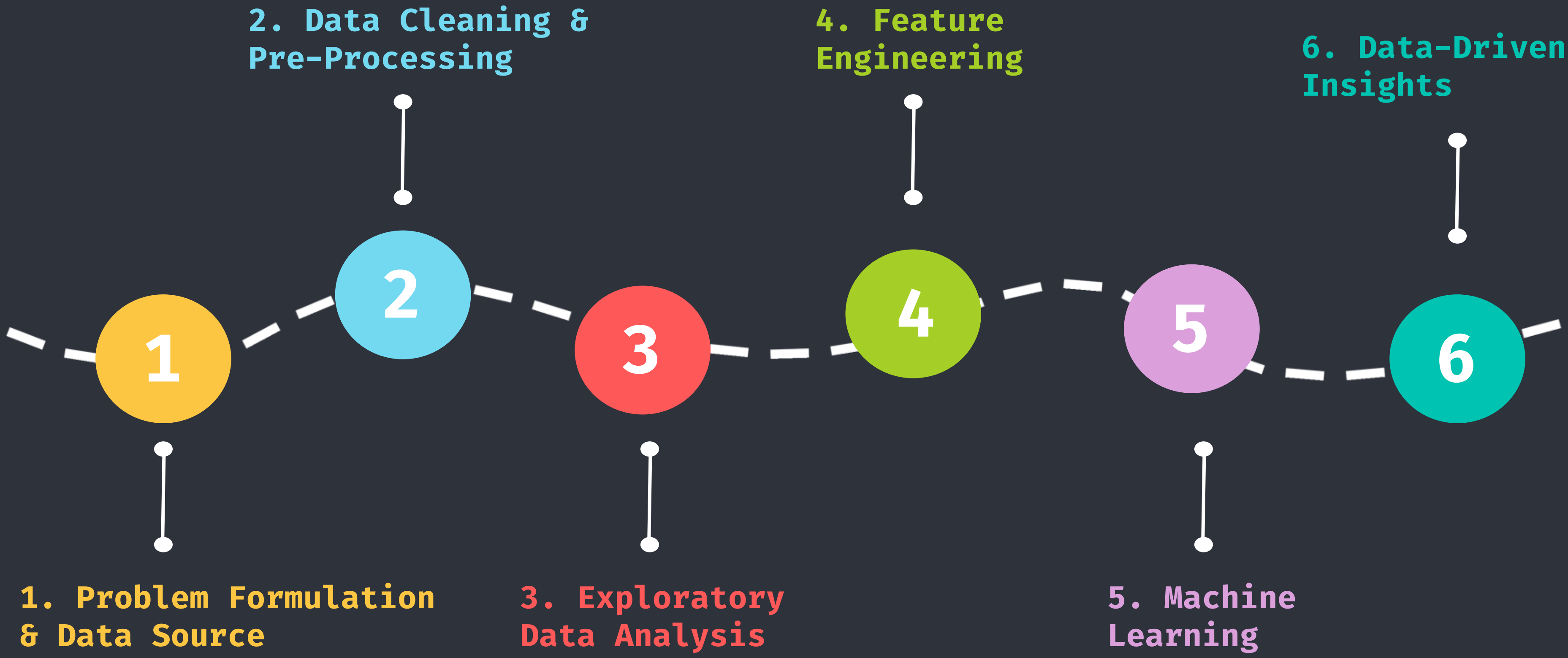
INCOME DETERMINANTS

EVER WONDERED WHAT INFLUENCES YOUR SALARY?

- - -

SC20 Group 6

RYAN NG . TEG SINGH TIWANA . AUGUSTINE LEE


2. Data Cleaning &
Pre-Processing
4. Feature
Engineering
6. Data-Driven
Insights
2
1
3
4
5
6
1. Problem Formulation
& Data Source
3. Exploratory
Data Analysis
5. Machine
Learning

{

PROBLEM FORMULATION
&
DATASET USED

}

MOTIVATION

- Regardless of fresh graduate or experienced hire, one key consideration when taking up a new job is:

  " How much salary should I expect? Am I being underpaid? "

- To answer this question, we need to understand what are the key factors affecting one's salary.

PROBLEM FORMULATION

- What are the **most important factors** affecting one’s salary?

- Can we build a **classification model** to help in predicting an income range for a job-seeker?

DATASET

Annual Social and Economic Supplements,

Current Population Survey 2021

{

DATA CLEANING

}

1. Filtering Dataset

	PERIDNUM	PH_SEQ	P_SEQ	A_LINENO	PF_SEQ	PHF_SEQ	OED_TYP1	OED_TYP2	OED_TYP3	PERRP	...	I_DISVL1	I_DISVL2	I_SURV
0	823894601190205110110 1	1	1	1	1	1	0	0	0	40	...	0	0	
1	8238946011902051101102	1	2	2	1	1	0	0	0	42	...	0	0	
2	8238946011902051101103	1	3	3	1	1	0	0	0	50	...	0	0	
3	6092052593105071201101	2	1	1	1	1	0	0	0	40	...	0	0	
4	6092052593105071201102	2	2	2	1	1	0	0	0	42	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
163538	0105117401643341311101	90757	1	1	1	1	0	0	0	41	...	0	0	
163539	1107604140345131311101	90758	1	1	1	1	0	0	0	40	...	0	0	
163540	1107604140345131311102	90758	2	2	1	1	0	0	0	42	...	0	0	
163541	9516061708016151311101	90759	1	1	1	1	0	0	0	40	...	0	0	
163542	9516061708016151311102	90759	2	2	1	1	0	0	0	45	...	0	0	

163543 rows × 830 columns

- 830 Columns - Columns split into 10 sub-groups:
  - (1) Record Identifiers, (2) Weights, (3) Demographics, (4) Basic CPS Items, (5) Work Experience, (6) Income, (7) Poverty, (8) Health Insurance, (9) Supplemental Poverty Measure, (10) Migration

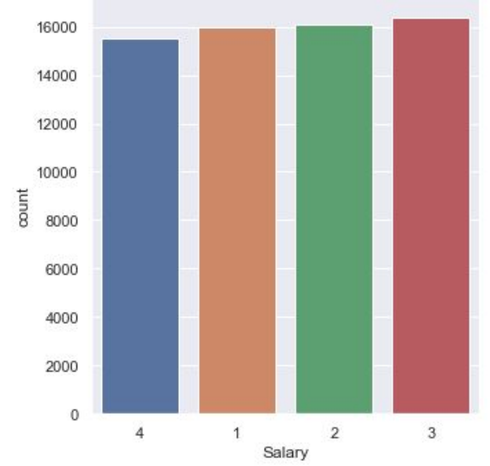
- Assumption made: Only features in Demographics, Basic CPS Items, Work Experience and Income are relevant to us.

- Also removed those who are not working & not receiving pay.

2. Check for Missing or Null Values

3. Numerical Encoding of Salary variable

- Split salary into 4 classes based on quartiles:
  Class 4: top 25%
  Class 3: 25-50% percentile (inclusive of 25)
  Class 2: 50-75% percentile (inclusive of 50)
  Class 1: 75-100% percentile (inclusive of 75)



- Target classes well balanced

4. Split into Train and Test Datasets

{

EXPLORATORY DATA ANALYSIS

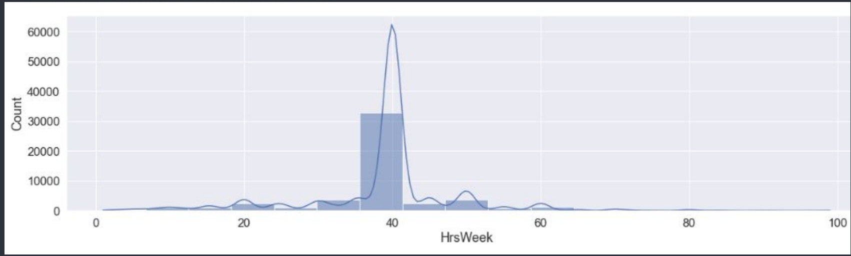
}

To better understand our data, we conducted:

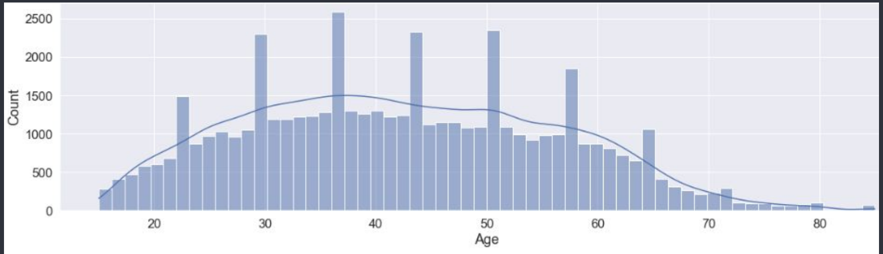
1. **Single-Variate Analysis** to understand our features
2. **Bi-Variate Analysis** to understand possible relationship of our features with salary
3. **Multi-Variate Analysis** to understand possible trends between features.

Single-Variate Analysis

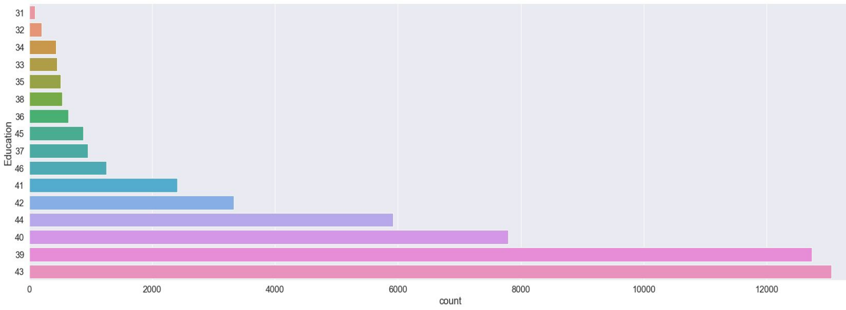
- Some interesting insights:



Most respondents work 40 hours weekly

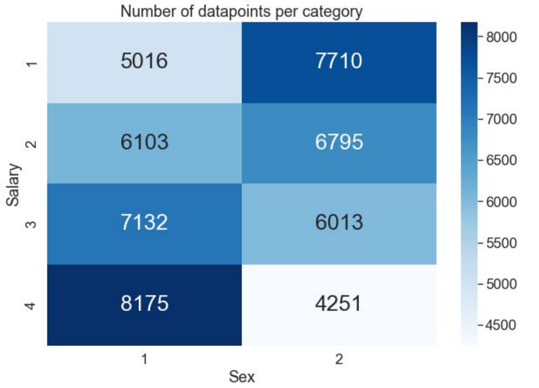


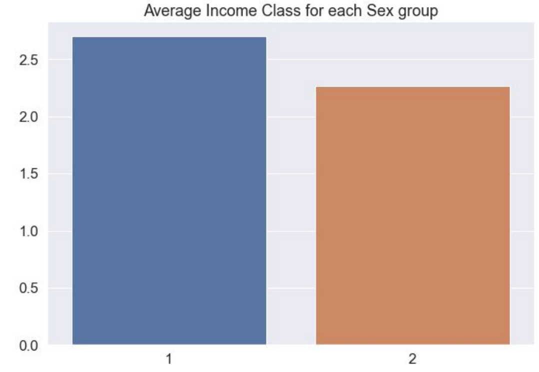
Most respondents aged between 20 to 64 years old



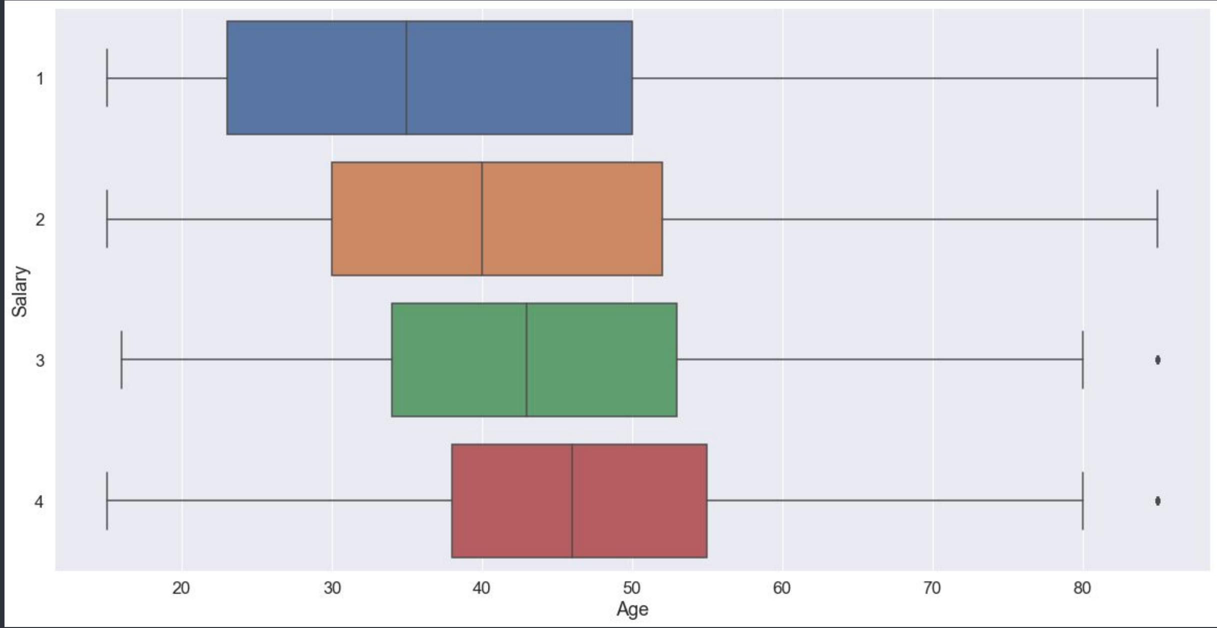
Most respondents are Bachelor Degree (43) holder or High School Graduate (39).

Bi-Variate Analysis



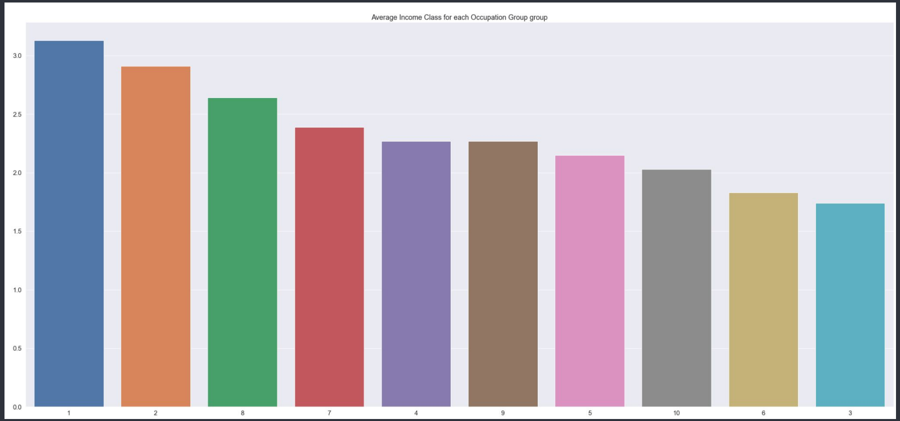


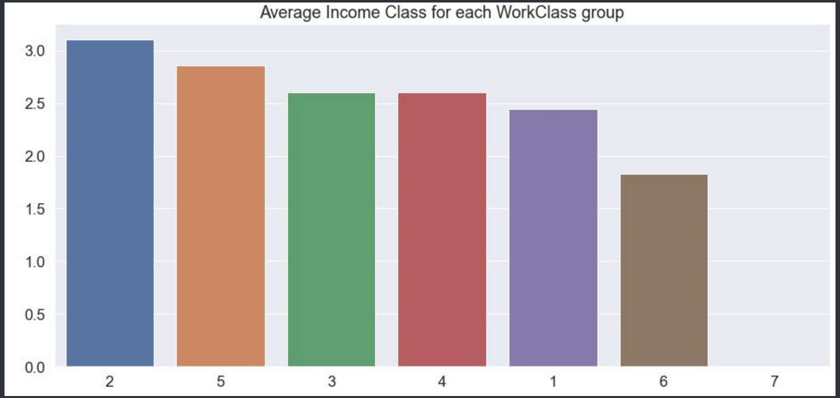
Males generally earn more than females



People who are older tend to earn more

Bi-Variate Analysis

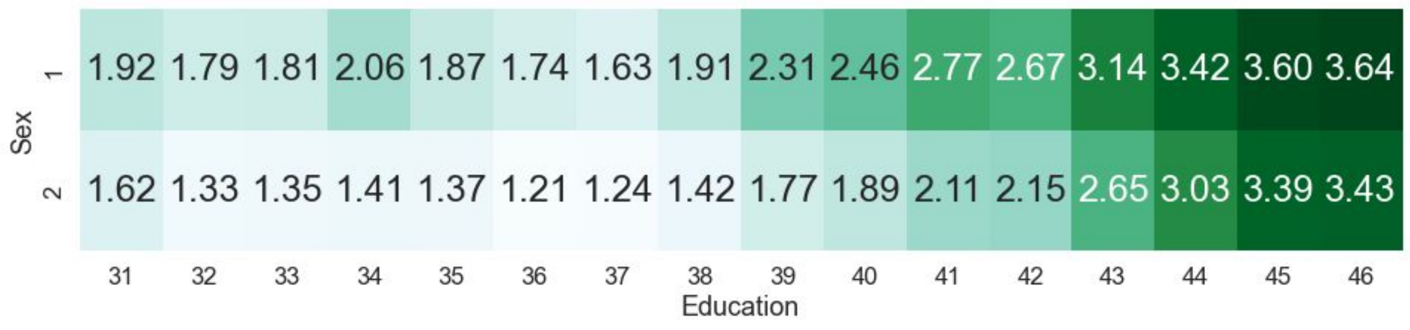




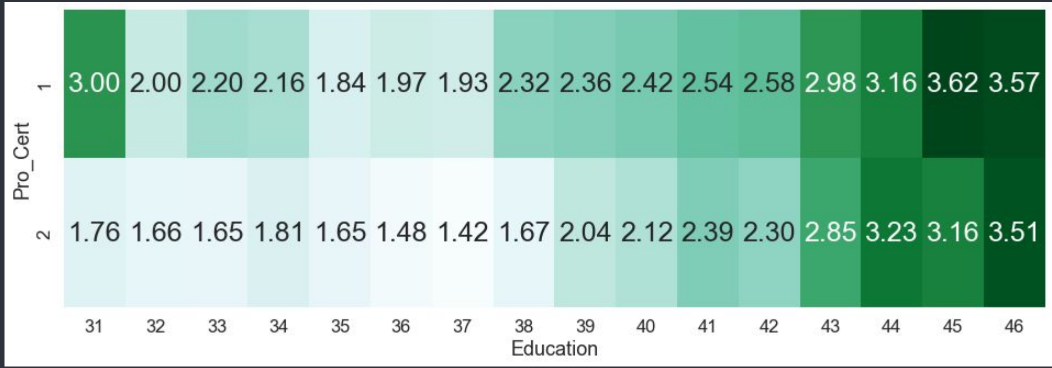
Occupation Group	Average Income Class
Management, Business & Financial Occupations (1)	3.13
Professional & Related Occupations (2)	2.91
Installation, Maintenance & Repair Occupations (8)	2.64
Construction and Extraction Occupations (7)	2.39
Sales & Related Occupations (4)	2.27
Production Occupations (9)	2.27
Office & Administrative Support (5)	2.15
Transportation & Material Moving Occupations (10)	2.03
Farming, Fishing & Forestry Occupations	1.83
Service Occupations (3)	1.74

Work Class	Average Income Class
Government - Federal (2)	3.1
Self-Employed - Incorporated (5)	2.86
Government - State (3)	2.6
Government - Local (4)	2.6
Private (1)	2.44
Self-Employed - Unincorporated (9)	1.83

Multi-Variate Analysis

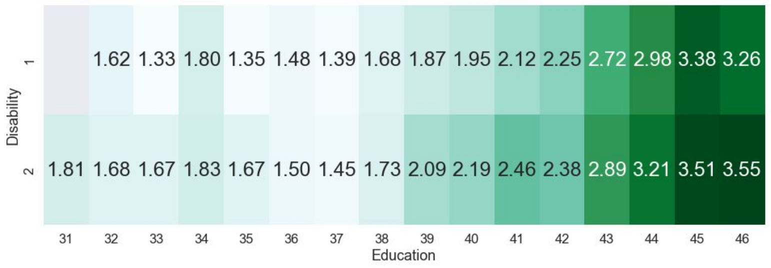


For the same education level, males (1) generally earn more

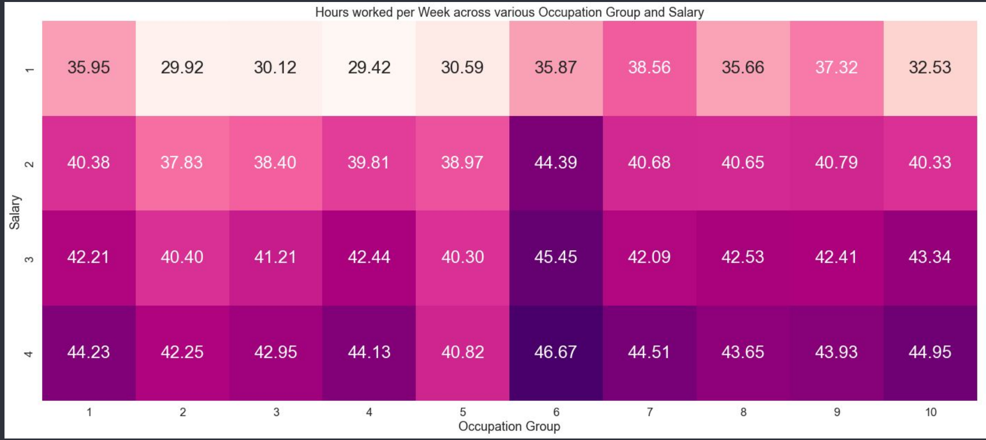


For the same education level, those with professional certificates (1) tend to earn more

Multi-Variate Analysis



For the same education level, those with disability (1) generally earn less



Jobs relating to Farming, Fishing and Forestry (6) work the longest hours to earn the same income class

{

FEATURE ENGINEERING

}

1. MinMax Scaling of Numerical Variables

- Aim: To improve performance that are more sensitive to scaling: SVM, KNN & MLP

Before Scaling

	Age	Last Week Working Hrs	HrsWeek
0	23	60	45
1	39	60	40
2	19	15	20
3	50	0	30
4	33	0	50
...	...	...	...
51190	33	40	40
51191	61	30	30
51192	71	8	8
51193	32	80	40
51194	35	60	60

51195 rows × 3 columns

After Scaling

	Age	Last Week Working Hrs	HrsWeek
0	0.114286	0.606061	0.448980
1	0.342857	0.606061	0.397959
2	0.057143	0.151515	0.193878
3	0.500000	0.000000	0.295918
4	0.257143	0.000000	0.500000
...	...	...	...
51190	0.257143	0.404040	0.397959
51191	0.657143	0.303030	0.295918
51192	0.800000	0.080808	0.071429
51193	0.242857	0.808081	0.397959
51194	0.285714	0.606061	0.602041

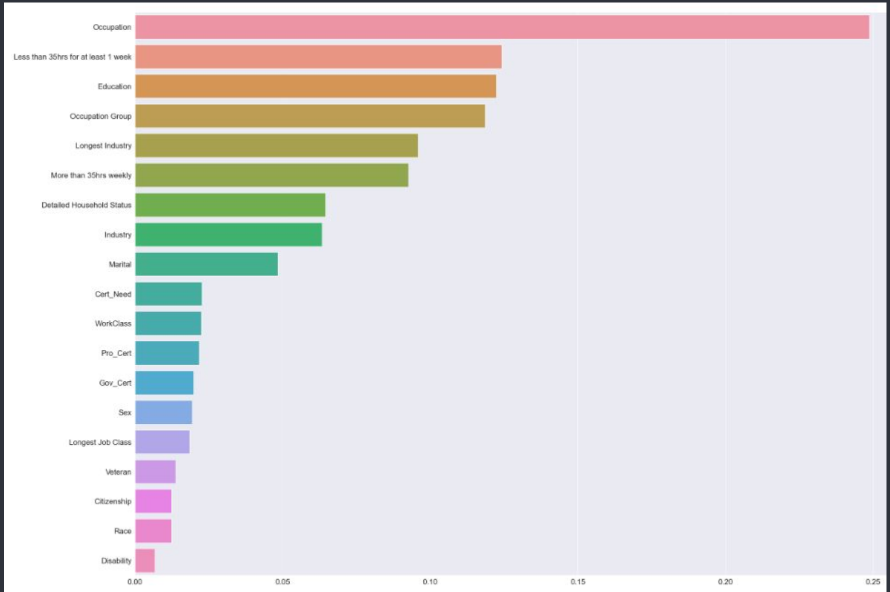
51195 rows × 3 columns

2. Mutual Information

**** Only done on train data to avoid information leakage**

- Aim: To understand categorical features' dependence with response variable.

- Features importance:




```
Occupation : 0.24866995116471946
Less than 35hrs for at least 1 week : 0.12424925709839307
Education : 0.12226432297143175
Occupation Group : 0.11867100557438492
Longest Industry : 0.09586694680866259
More than 35hrs weekly : 0.09258318088114459
Detailed Household Status : 0.06453934691995089
Industry : 0.06328110854554314
Marital : 0.04849125955930278
Cert_Need : 0.022669510478366295
WorkClass : 0.02245016757791296
Pro_Cert : 0.021712109572769478
Gov_Cert : 0.019880597732325977
Sex : 0.019349025035164225
Longest Job Class : 0.018499135976110637
Veteran : 0.013760484296965636
Citizenship : 0.0124466283204554
Race : 0.012352572246313809
Disability : 0.006655613588452347
```

3. Chi-Squared Test for Independence

**** Only done on train data to avoid information leakage**

- Aim: To corroborate MI's findings by also understanding categorical features' dependence with response variable.

- P-Score of features:

```
Disability : 0.4537402007277509
Veteran : 0.22151875612092423
Race : 1.2228736395956913e-06
Longest Job Class : 3.427751770507403e-13
Industry : 4.984059473715531e-33
Pro_Cert : 4.0002841568913574e-44
Citizenship : 9.7997660792459e-62
WorkClass : 3.9272609846339284e-65
Sex : 1.1820928452009293e-68
Longest Industry : 3.873866133204988e-134
Education : 0.0
Marital : 0.0
Gov_Cert : 0.0
Cert_Need : 0.0
Less than 35hrs for at least 1 week : 0.0
Detailed Household Status : 0.0
More than 35hrs weekly : 0.0
Occupation Group : 0.0
Occupation : 0.0
```

Findings from MI & Chi-2

**** Only done on train data to avoid information leakage**

- From both MI and Chi-2, disability status has the least dependence with salary.

- Disability status and veteran status P Scores significantly higher than others → least dependent with salary.

- However, features' P scores all below 0.05 → significant dependence with salary.

- We decided to remove disability status and veteran status but we recognise that this dimension reduction may not improve our model performance.

MACHINE LEARNING MODELS

What are the most important factors affecting one's salary?

Can we build a <u>classification model</u> to help in predicting an income range for a job-seeker?

How much salary should I expect? Am I being underpaid?

We attempted the following:

1. **Support Vector Machines(SVMs)**
2. **Logistic Regression**
3. **K-Nearest Neighbor(KNN)**
4. **Decision Trees**
5. **Adaptive Boosting (AdaBoost)**
6. **Gradient Boosting(CatBoost, XGBoost & GradientBoostedMachine(GBM))**

After train-test split,

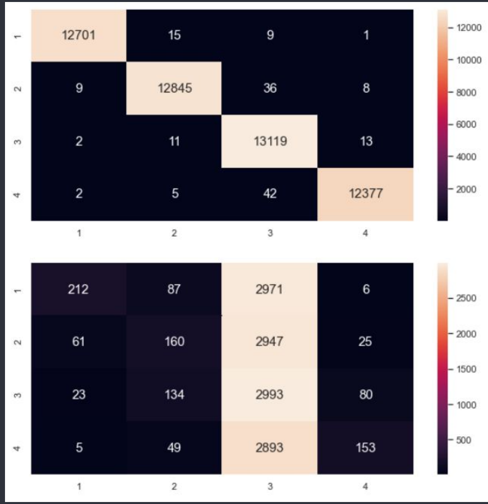
1) Regular Dataset
2) Feature-engineered Dataset

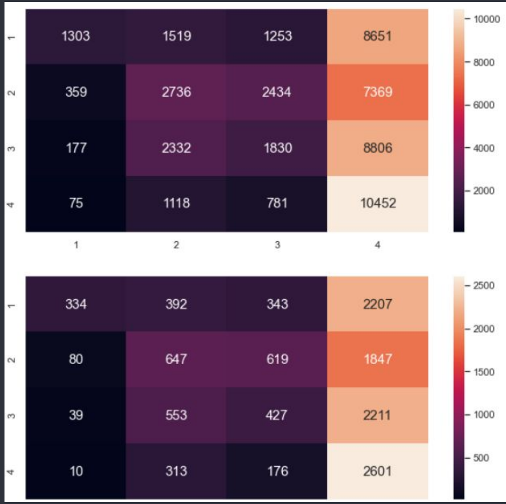
1. Support Vector Machines (SVMs)

1) Initial tests with **regular** dataset & default parameters on various kernels

Train

Test





Radial Basis Function | Polynomial | Sigmoid

F1 (RBF Kernel): 0.174888

F1 (Polynomial Kernel): 0.259292

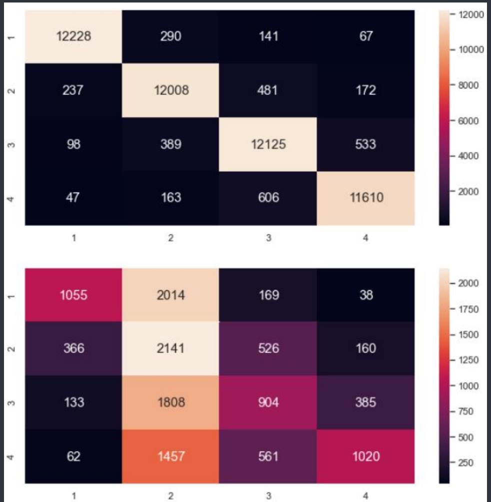
F1 (Sigmoid Kernel): 0.280103

1. Support Vector Machines (SVMs)

2) Initial tests with **feature engineered** dataset & default parameters on various kernels

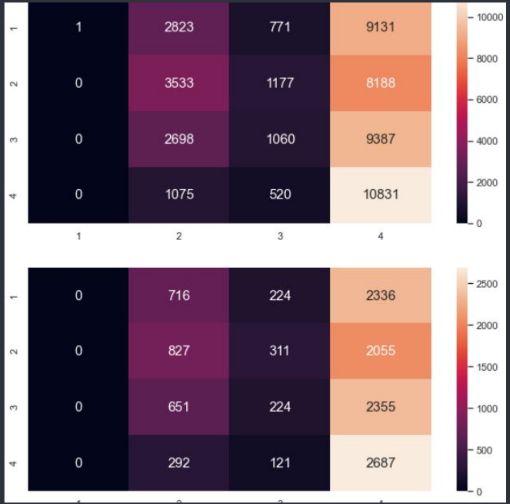
Train

Test



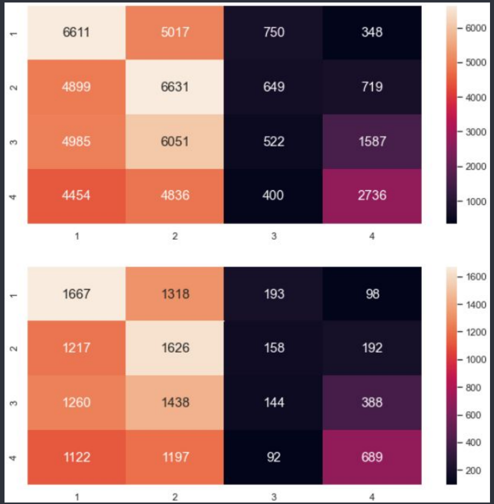
Radial Basis Function

F1 (RBF Kernel): 0.400765



Polynomial

F1 (Polynomial Kernel): 0.204022



Sigmoid

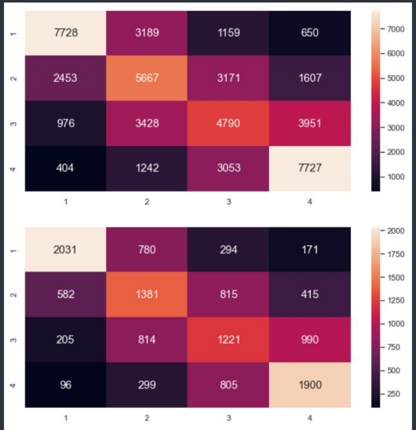
F1 (Sigmoid Kernel): 0.286146

2. Logistic Regression

1) Initial tests with **regular** dataset & default parameters (max_iter = 100000) on various solvers

Train

Test



lbfgs

F1 (lbfgs solver): 0.511295

sag

F1 (sag solver): 0.498697

saga

F1 (saga solver): 0.490747

liblinear

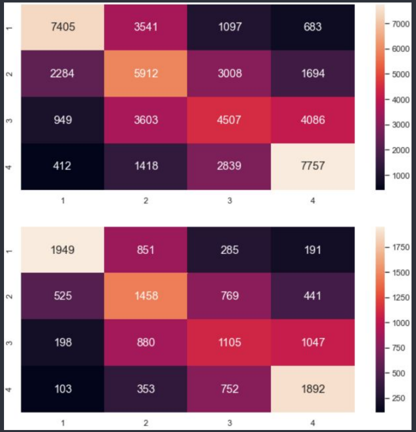
F1 (liblinear solver): 0.502402

2. Logistic Regression

2) Initial tests with **feature-engineered** dataset & default parameters (max_iter = 100000) on various solvers

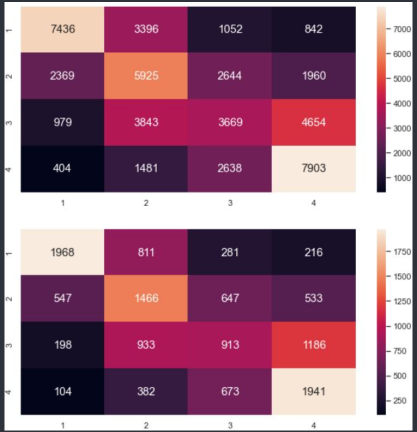
Train

Test



lbfgs

F1 (lbfgs solver): 0.501104



sag

F1 (sag solver): 0.488158

saga

F1 (saga solver): 0.479224

liblinear

F1 (liblinear solver): 0.498409

2. Logistic Regression

3) Hyperparameter tuning

- Changes to max_iter appear to have no effect, so long as large enough
- Newton-CG will be omitted
- RandomizedSearchCV used

RandomizedSearchCV cv value	Best Parameters			
	solver	penalty	C	Score
2	liblinear	l1	10	**0.514015**
3	liblinear	l1	1	0.513038
4	liblinear	l1	10	0.513820

2. Logistic Regression

3) Hyperparameter tuning

- RandomizedSearchCV used

RandomizedSearchCV cv value	Best Parameters			
	solver	penalty	C	Score
2	liblinear	l1	10	**0.514015**
3	liblinear	l1	1	0.513038
4	liblinear	l1	10	0.513820

- Run a *for* loop with C values 1 to 99 (random_state = 42) to determine the optimal C value

2. Logistic Regression

3) Hyperparameter tuning

- Results, optimal parameters and best F1-score:

Train

Test

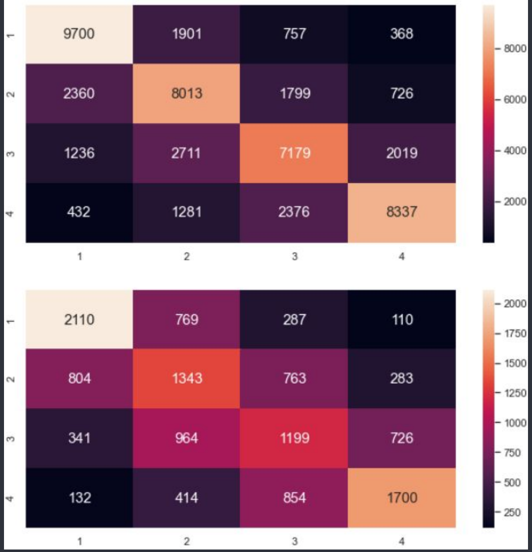
Solver	liblinear
Penalty	l1
C	3
F1-score	0.512655

3. K-nearest Neighbors

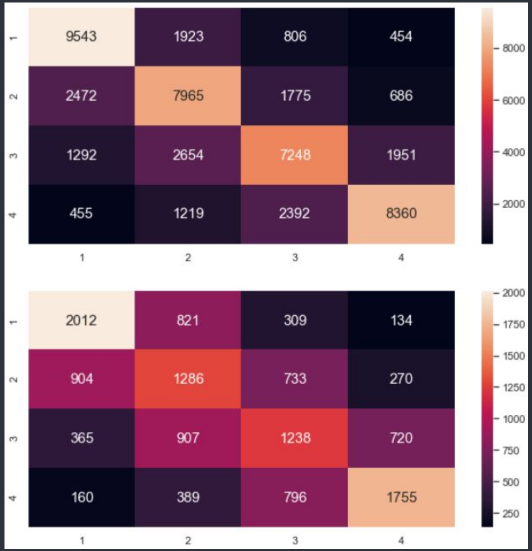
1) Initial tests with default parameters on **regular and feature engineered** datasets

Train

Test



Regular



Feature Engineered

F1 (K nearest neighbours): 0.497063

F1 (K nearest neighbours): 0.491893

3. K-nearest Neighbors

2) Hyperparameter tuning

- Changes to leaf_size values appear to have no effect on F1-score
- Only n_neighbors parameter will be tested
- Done on regular dataset
- Run a *for* loop with n_neighbors values between 1 and 199 to determine optimal n_neighbors value

	n	F1
0	1	0.497063
17	171	0.497063
16	161	0.497063
15	151	0.497063
14	141	0.497063
13	131	0.497063
12	121	0.497063
11	111	0.497063
10	101	0.497063
9	91	0.497063
8	81	0.497063
7	71	0.497063
6	61	0.497063
5	51	0.497063
4	41	0.497063
3	31	0.497063
2	21	0.497063
1	11	0.497063
18	181	0.497063
19	191	0.497063

3. K-nearest Neighbors

2) Hyperparameter tuning

- Results, optimal parameters and best F1-score:

Regular	
n_neighbors	48
F1-score	0.519702

Feature-Engineered	
n_neighbors	23
F1-score	0.512647

4. Decision Trees

1) Initial tests with default parameters on **regular** dataset

Regular Dataset

F1 (Dec. Tree): 0.47628

4. Decision Trees

2) Hyperparameter tuning

- Changes to max_depth values of the trees
- Optimal max-depth found: range(7,13)
- Done on regular dataset through cross-validation with GridSearchCV

4. Decision Trees

2) Hyperparameter tuning Results

- Results, optimal parameters and best F1-score:

Train

Test

Regular	
max_depth	11
F1-score	**0.53561**

5. Adaptive Boosting(AdaBoost)

1) Initial tests with default parameters on **regular and feature engineered** datasets

Train

Test

Regular

F1 (Ada): 0.56464

Feature Engineered

F1 (Ada): 0.56271

5. Adaptive Boosting(AdaBoost)

2) Hyperparameter tuning

- Changes to n_estimators values & learning_rate
- Optimal Parameter Ranges
  - n_estimators found: range(400,1050)
  - learning_rate found: range(0.01,1.25)
- Done on regular dataset through cross-validation with RandomSearchCV

5. Adaptive Boosting(AdaBoost)

2) Hyperparameter tuning

- Results, optimal parameters and best F1-score:

Train

Test

Regular	
n_estimators	1000
learning_rate	0.86765
F1-score	0.56669

Feature-Engineered	
F1-score	**0.56688**

6.1) Gradient Boosting(GBM)

1) Initial tests with default parameters on **regular and feature engineered** datasets

Train

Test

Regular

F1 (GBM): 0.57255

Feature Engineered

F1 (GBM): 0.57231

6.1) Gradient Boosting(GBM)

2) Hyperparameter tuning

- Changes to n_estimators values, subsample, max_depth & learning_rate
- Optimal
  - n_estimators: range(400,1050)
  - learning rate: range(0.01,0.20)
  - subsample: range(0.6, 0.95)
  - max_depth: range(3,8)
- Done on regular dataset through cross-validation with RandomSearchCV

6.1) Gradient Boosting(GBM)

2) Hyperparameter tuning

- Results, optimal parameters and best F1-score:

Train

Test

Regular	
n_estimators	450
learning_rate	0.02583
subsample	0.65
depth	6
F1-score	**0.5783**

Feature-Engineered	
F1-score	0.5767

6.2) CatBoost

1) Initial tests with default parameters on **regular and feature engineered** datasets

Train

Test

Regular

Feature Engineered

F1 (CB): 0.57006

F1 (CB): 0.56786

6.2) CatBoost

2) Hyperparameter tuning
   - Changes to iterations value, depth & learning_rate
   - Optimal
     - iterations: range(450,1050)
     - learning rate: range(0.01,0.19)
     - depth: range(4,7)
   - Done on regular dataset through cross-validation with RandomSearchCV

6.2) CatBoost

2) Hyperparameter tuning

- Results, optimal parameters and best F1-score:

Train

Test

Regular	
iterations	1000
learning_rate	0.1149
depth	4
F1-score	**0.57596**

Feature-Engineered	
F1-score	0.57577

6.3) XGBoost

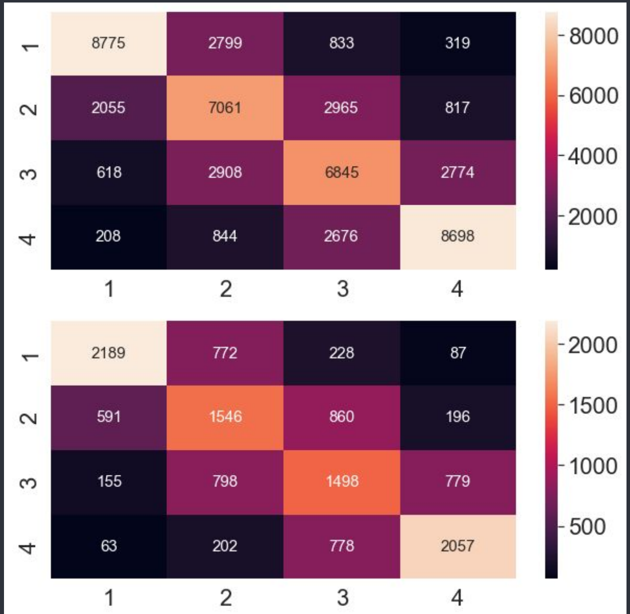
1) Initial tests with default parameters on **regular and feature engineered** datasets

Train

Test

Regular

F1 (XG): 0.57372



Feature Engineered

F1 (XG): 0.57609

6.3) XGBoost

2) Hyperparameter tuning

- Changes to n_estimators values, subsample, max_depth & learning_rate
- Optimal ranges
  - n_estimators: range(300, 500)
  - subsample: range(0.6, 0.8)
  - max_depth: range(2, 6)
- Done on regular dataset through cross-validation with GridSearchCV

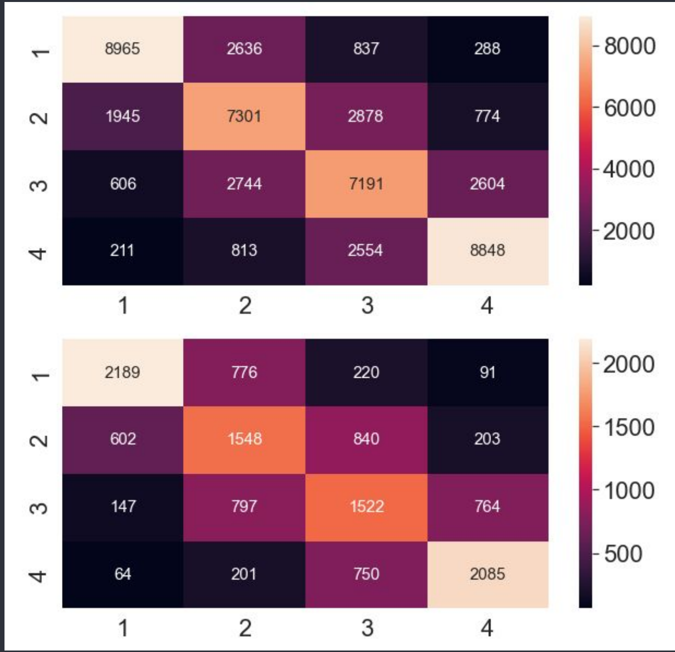
6.2) XGBoost

2) Hyperparameter tuning

- Results, optimal parameters and best F1-score:

Train

Test



Regular	
n_estimators	450
learning_rate	0.02583
subsample	0.65
depth	6
F1-score	0.57214

Feature-Engineered	
F1-score	**0.57609**

{

CONCLUSION
&
DATA-DRIVEN INSIGHTS

}

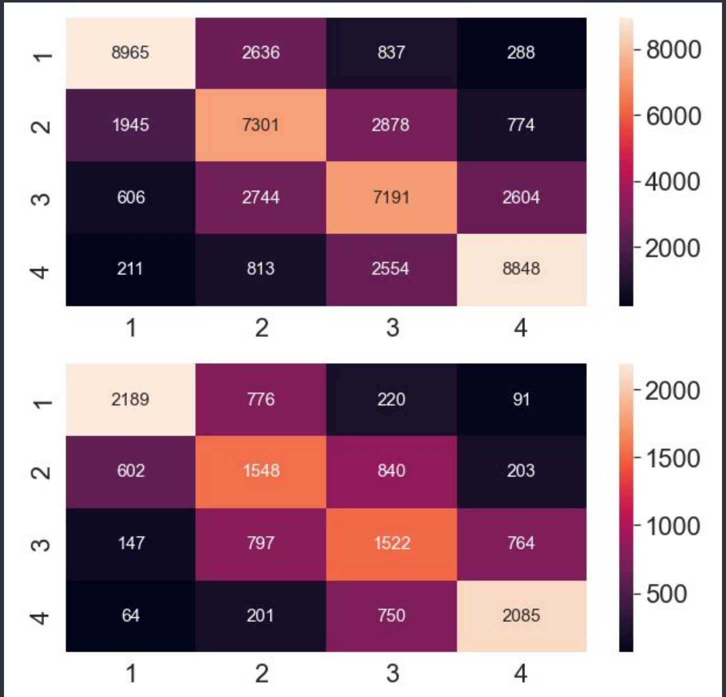
ML Outcomes

Rank	Name	F1-Score (Test)	Model Fit	Important Features
1	**XGBoost**	**0.57609**	**Good Fit**	**Marital Status, Working Hours, Age**
2	**CatBoost**	**0.57596**	**Good Fit**	**Industry, Education level, Citizen Status**
3	Gradient Boosting	0.57836	Over-Fitted	Occupation, Working Hours, Education level
4	Adaptive Boosting	0.56669	Good Fit	Industry, Occupation, Age
5	Decision Trees	0.53561	Good Fit	Industry, Age, Occupation
6	Logistic Regression	0.51265	Good Fit	Pro_cert, Cert_Need, Sex
7	K-Nearest Neighbor	0.51264	Good Fit	NA
8	Support Vector Machines (SVMs)	0.40077	Over-Fitted	NA

ML Outcomes

Train

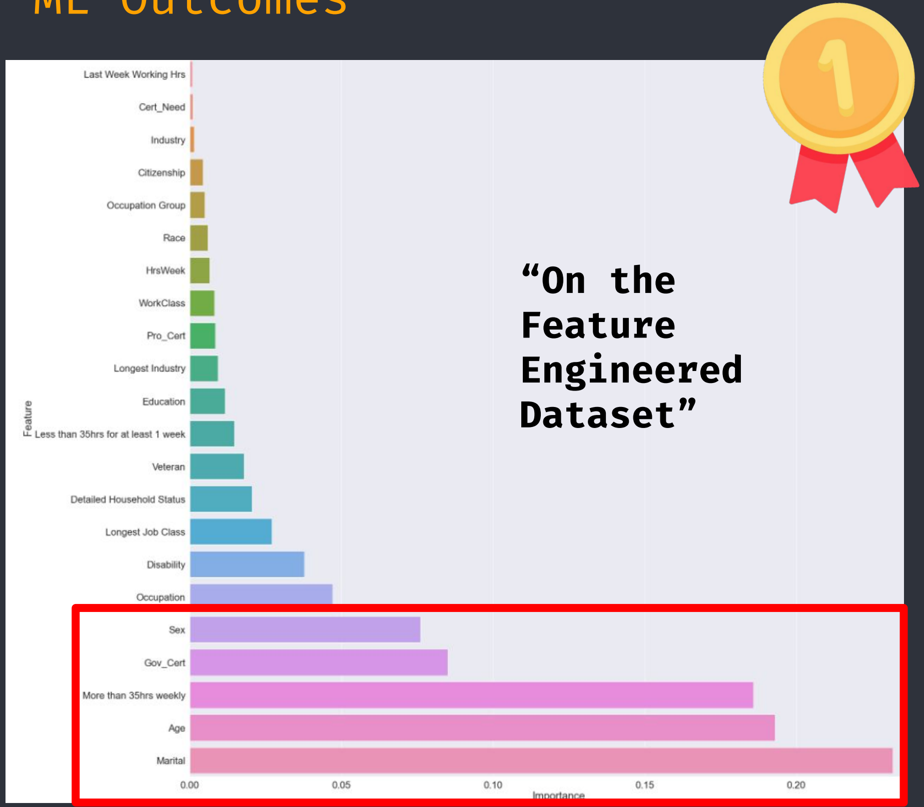
Test



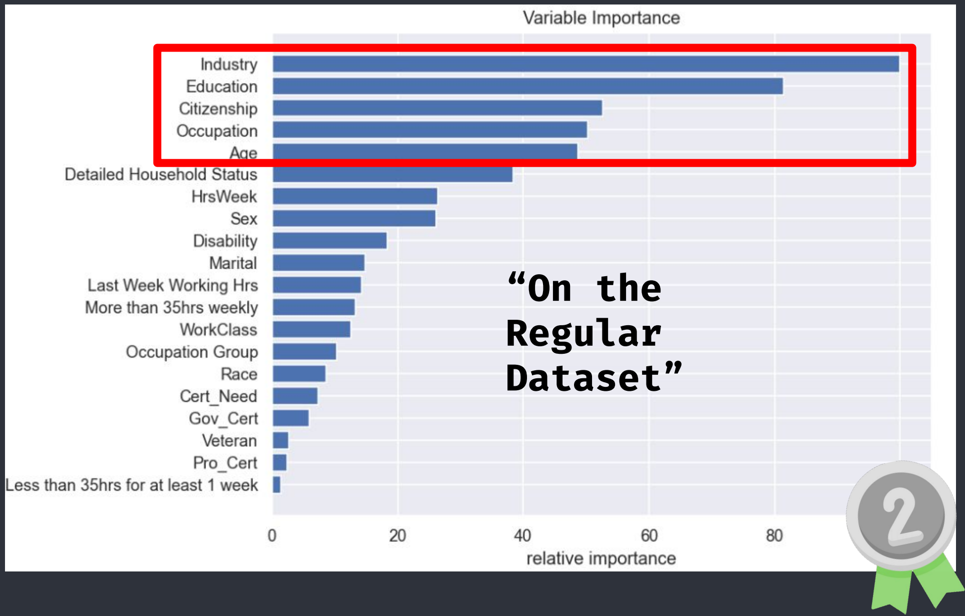
XGBoost

CatBoost

ML Outcomes



"XGBoost": Marital Status, Working Hours, Age



"CatBoost": Industry, Education level, Citizen Status

Overall Insights and Observations

What are the most important factors affecting one's salary?

Can we build a classification model to help in predicting an income range for a job-seeker?

How much salary should I expect? Am I being underpaid?

Rank	Name	F1-Score (Test)	Model Fit	Important Features
1	XGBoost	0.57609	Good Fit	Marital Status, Working Hours, Age
2	CatBoost	0.57596	Good Fit	Industry, Education level, Citizen Status
3	Gradient Boosting	0.57836	Over-Fitted	Occupation, Working Hours, Education level

Salary can be affected by non-quantifiable data:

- Connections
- EQ
- Negotiation skills
- Individual interview & presentation performance
- Company's budget availability
- Compensation methods
  - Stock options
  - Housing
  - Education

Links:

GitHub Link:	https://github.com/auglxw/SC1015_proj
Youtube Link:	https://www.youtube.com/watch?v=Dvs-F70_L8Q

Special Thanks to Dr. Sourav & TA Zhou Shaowen for their guidance throughout this project.