Flu Shot Learning

Team 10

Grigoriy Morozov, Iuliia Parshchikova

Goal

is to predict how likely individuals are to receive their H1N1 and seasonal flu vaccines.

respondent_id	h1n1_concern	h1n1_knowledge	behavioral_antiviral_meds	behavioral_avoidance	behavioral_face_mask	behavioral_wash_hands	behavioral_large_gathe
0	1.0	0.0	0.0	0.0	0.0	0.0	
1	3.0	2.0	0.0	1.0	0.0	1.0	
2	1.0	1.0	0.0	1.0	0.0	0.0	

Each row in the dataset represents one person who responded to the National 2009 H1N1 Flu Survey.

The features in the dataset

h1n1_concern	Level of concern about the H1N1 flu. 0 = Not at all concerned; 1 = Not very concerned; 2 = Somewhat concerned; 3 = Very concerned
h1n1_knowledge	Level of knowledge about H1N1 flu. 0 = No knowledge; 1 = A little knowledge; 2 = A lot of knowledge.
behavioral_antiviral_meds	Has taken antiviral medications. (binary)
chronic_med_condition	Has any of the following chronic medical conditions: asthma or an other lung condition, diabetes, a heart condition, a kidney condition, sickle cell anemia or other anemia, a neurological or neuromuscular condition, a liver condition, or a weakened immune system caused by a chronic illness or by medicines taken for a chronic illness. (binary)
opinion_seas_sick_from_vacc	Respondent's worry of getting sick from taking seasonal flu vaccine. 1 = Not at all worried; 2 = Not very worried; 3 = Don't know; 4 = Somewhat worried; 5 = Very worried.
age_group	Age group of respondent.

Labels

	respondent_id	h1n1_vaccine	seasonal_vaccine
0	0	0	0
1	1	0	1
2	2	0	0
3	3	0	1
4	4	0	0
...	...	...	...
26702	26702	0	0
26703	26703	0	0
26704	26704	0	1
26705	26705	0	0
26706	26706	0	0

26707 rows × 3 columns

2 target variables:

- h1n1_vaccine - Whether respondent received H1N1 flu vaccine.
- seasonal_vaccine - Whether respondent received seasonal flu vaccine.

Both are binary variables: 0 = No; 1 = Yes. Some respondents didn't get either vaccine, others got only one, and some got both. This is formulated as a multilabel (and not multiclass) problem.

Preprocessing data

- drop column with the highest missing rate (% of NaNs)

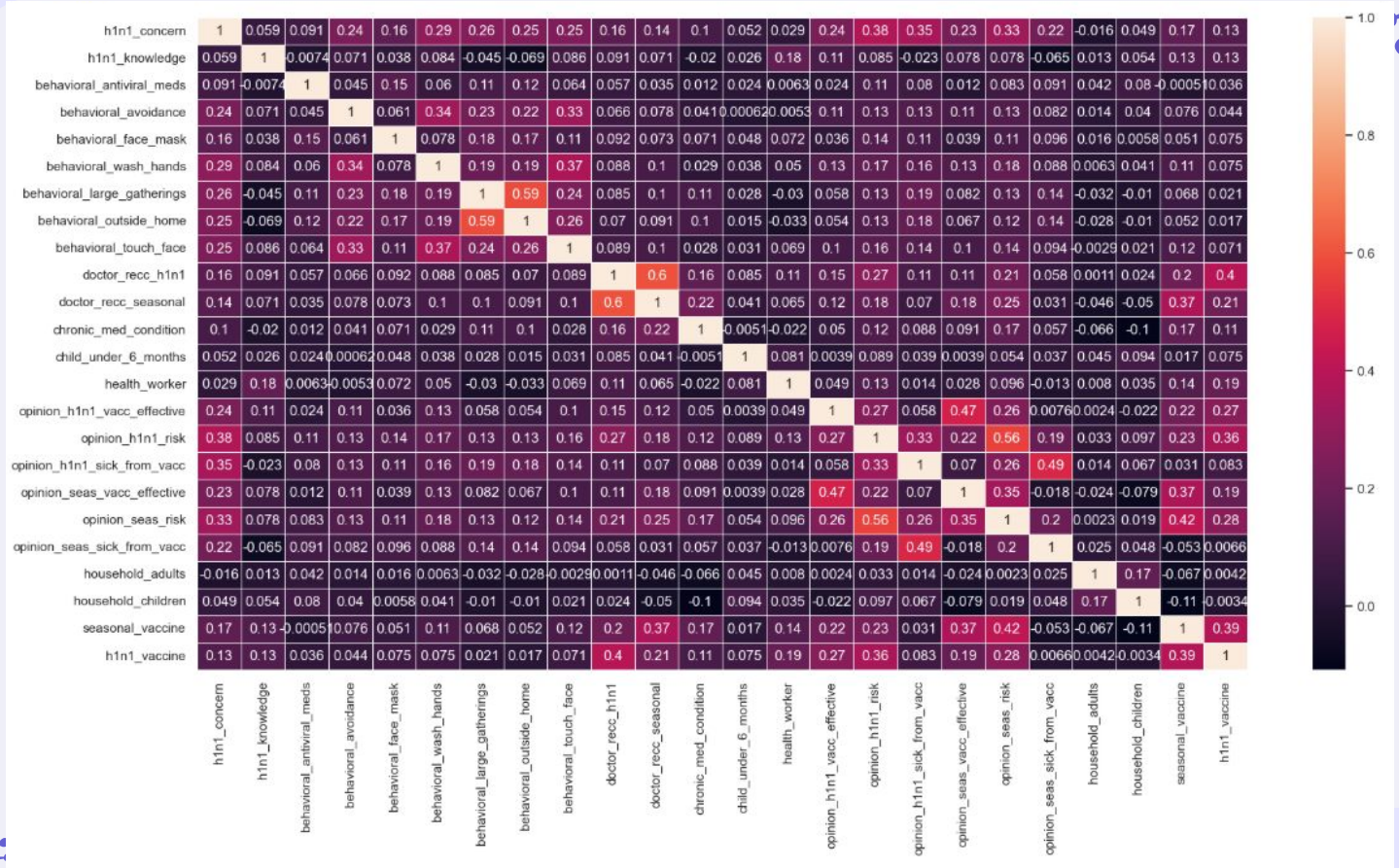
```
health_insurance  : 45.96%
employment_industry: 49.91%
employment_occupation: 50.44%
```

age_group	5
education	4
race	4
sex	2
income_poverty	3
marital_status	2
rent_or_own	2
employment_status	3
hhs_geo_region	10
census_msa	3

- explore columns with unique names (do not drop them)
- use one-hot encoding to remove categorical columns
- apply normalisation to numerical features
- divide into train and test sets

```
Train shape: (15713, 61)
Test shape: (3929, 61)
Train target shape: (15713, 2)
Test target shape: (3929, 2)
```

Correlation between numerical and categorical features



According to the correlation matrix there no reason to drop any columns

Training models

```
reg = MultiOutputClassifier(LogisticRegression(max_iter=5000))
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)
f1 = f1_score(y_test, y_pred, average='macro')
print(f1.round(3))
```

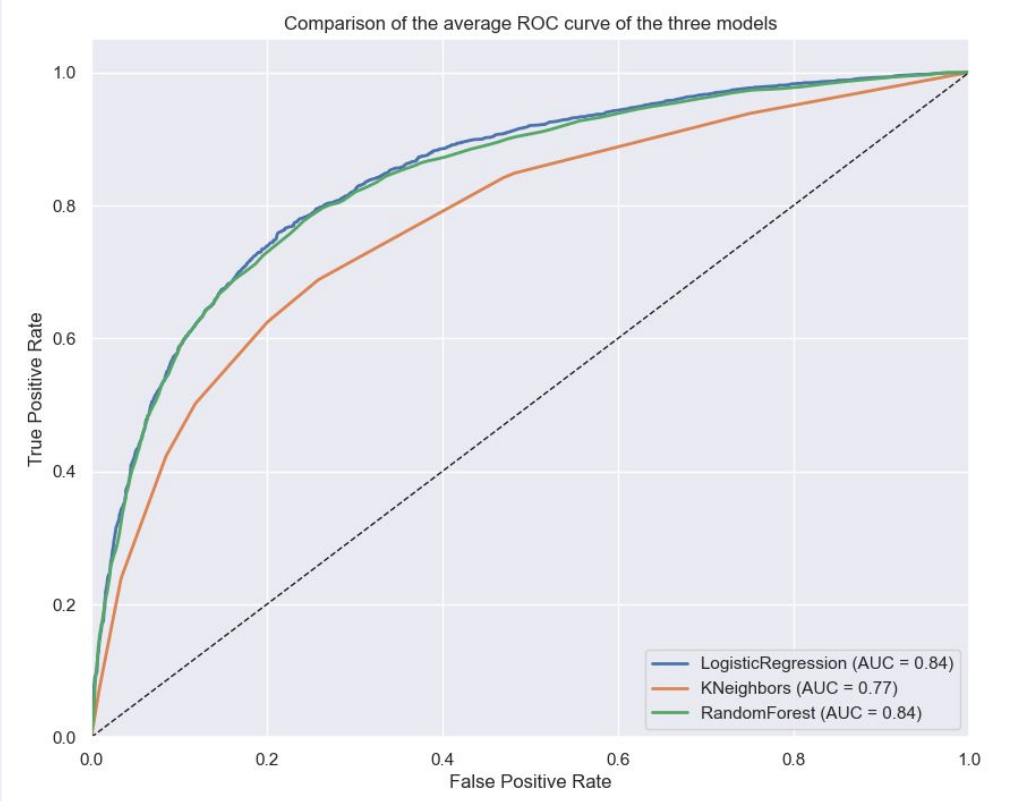
```
0.662
```

```
clf = MultiOutputClassifier(KNeighborsClassifier())
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
f1 = f1_score(y_test, y_pred, average='macro')
print(f1.round(3))
```

```
0.613
```

```
rf = MultiOutputClassifier(RandomForestClassifier())
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
f1 = f1_score(y_test, y_pred, average='macro')
print(f1.round(3))
```

```
0.64
```



GridSearchCV

- LogisticRegression. F1 = 0.662 —> F1 = 0.663

```
{'estimator__C': 2.0, 'estimator__max_iter': 5000, 'estimator__penalty': 'l2'}
```

- KNeighborsClassifier. F1 = 0.613 —> F1 = 0.615

```
{'estimator__n_neighbors': 17,
 'estimator__p': 2,
 'estimator__weights': 'uniform'}
```

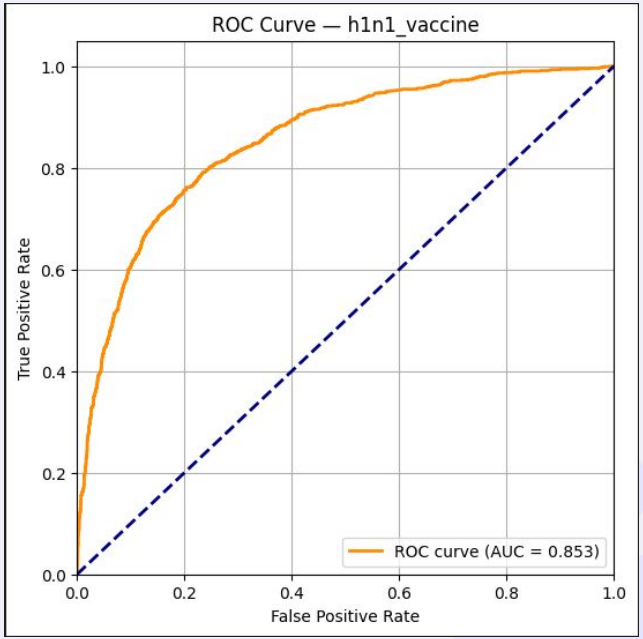
- RandomForestClassifier. F1 = 0.640 —> F1 = 0.653

```
{'estimator__max_features': 'sqrt', 'estimator__n_estimators': 500}
```

XGBoost Model

h1n1 vaccine

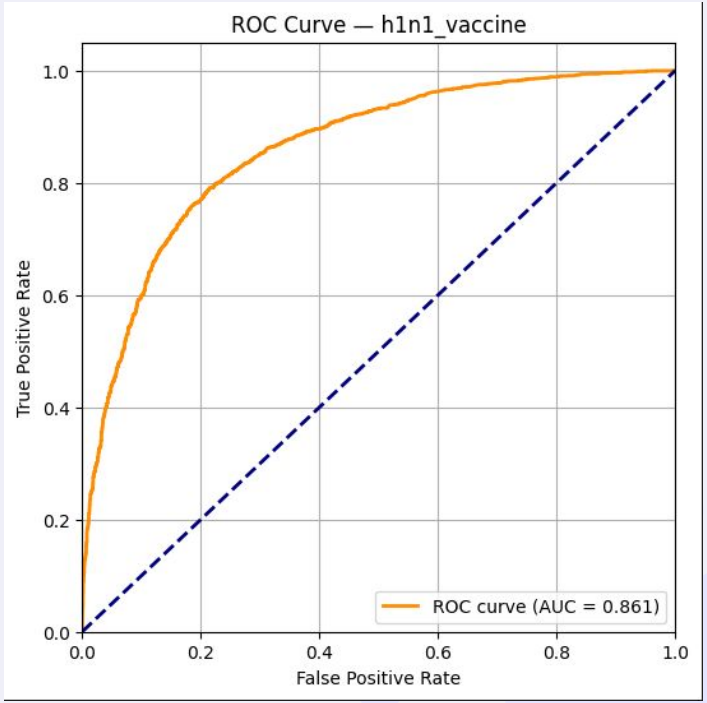
```
Overall Accuracy: 0.830 ± 0.006
Overall F1: 0.533 ± 0.013
Overall AUC: 0.842 ± 0.008
```



XGBoost Model

season vaccine

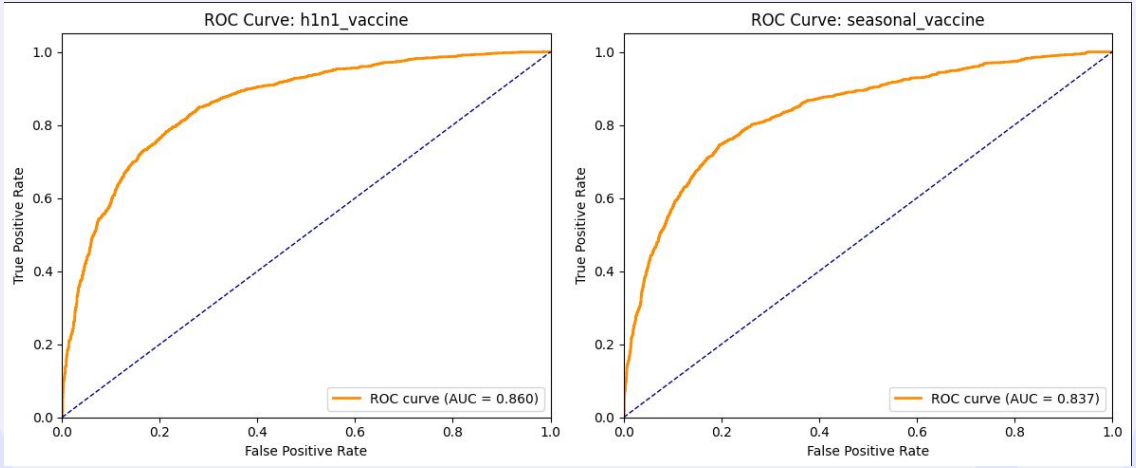
```
Overall Accuracy: 0.782 ± 0.005
Overall F1: 0.767 ± 0.006
Overall AUC: 0.858 ± 0.005
```



XGBoost Model

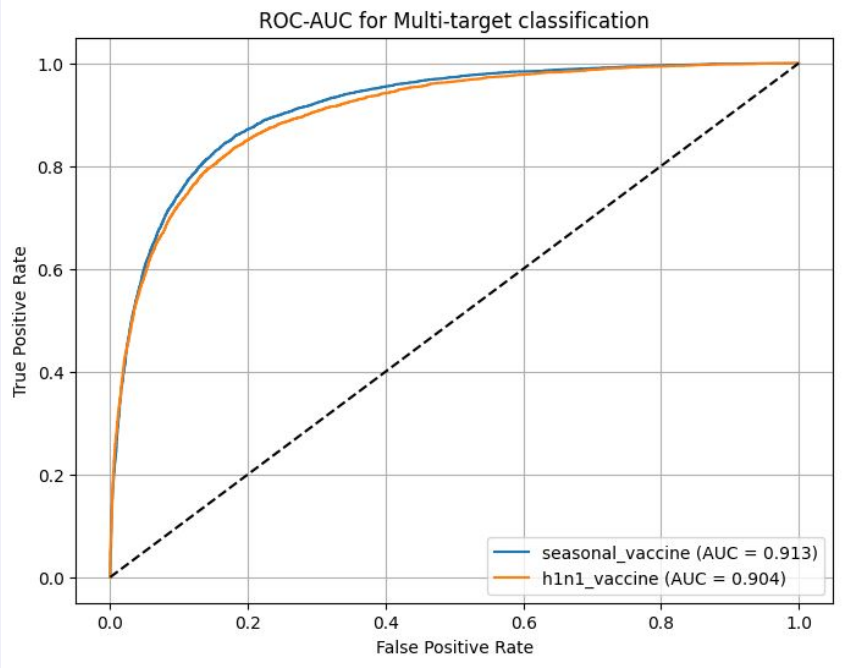
Both vaccines

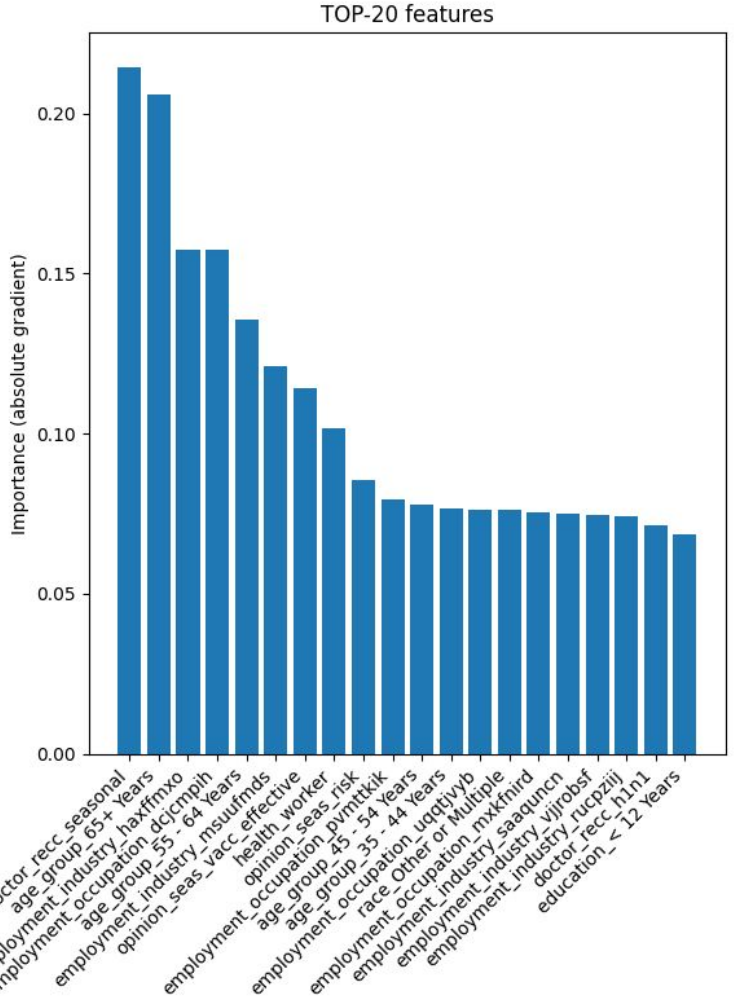
```
Fold 1: Accuracy = 0.810, F1 = 0.675
Fold 2: Accuracy = 0.805, F1 = 0.672
Fold 3: Accuracy = 0.808, F1 = 0.659
Fold 4: Accuracy = 0.805, F1 = 0.670
Fold 5: Accuracy = 0.798, F1 = 0.663

Overall Accuracy: 0.805 ± 0.004
Overall F1-macro: 0.668 ± 0.006
```

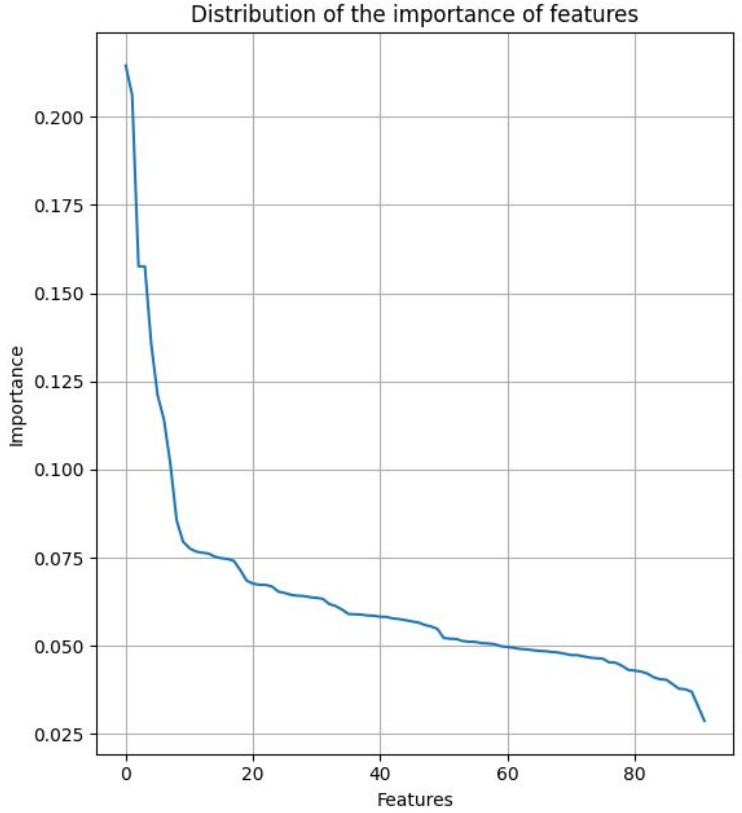


Multi-Layer Perceptron Model

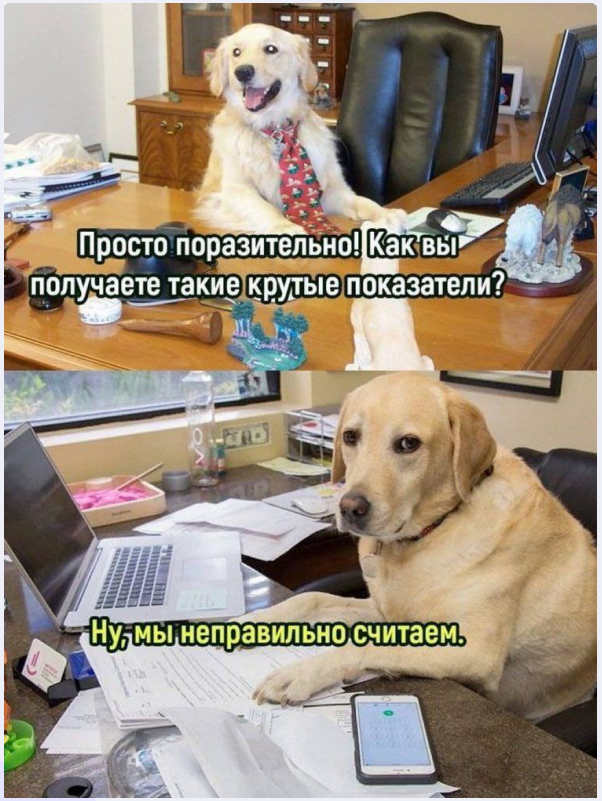
```
Best param: {'hidden_layers': (128, 64), 'activation': 'relu', 'lr': 0.01, 'weight_decay': 0.001}
F1: 0.6969
```




TOP-20 features
Importance (absolute gradient)
0.00
0.05
0.10
0.15
0.20
doctor_recc_seasonal
age_group_65+ Years
employment_industry_haxffmxo
employment_occupation_dcjcmpih
age_group_55 - 64 Years
employment_industry_msuufmds
opinion_seas_vacc_effective
health_worker
opinion_seas_risk
employment_occupation_pvmttkik
age_group_45 - 54 Years
age_group_35 - 44 Years
employment_occupation_uqqtjvyb
race_Other or Multiple
employment_occupation_mxkfnird
employment_industry_saaquncn
employment_industry_vjjrobsf
employment_industry_rucpziij
doctor_recc_h1n1
education_< 12 Years


Distribution of the importance of features
Importance
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0
20
40
60
80
Features

THANKS!



- That's amazing! How do you get such great results?
- Well, we're calculating it wrong.