Prediction of Brazil's Individual Income using Machine Learning

Rebeca Nunes Rodrigues

Summary

- Introduction
  - Reference work
  - Applications
- Methodology
  - Data
  - Data Preparation
  - Machine learning models
- Results

Bryant University

HONORS THESIS

Prediction of Individual Level Income: A Machine Learning Approach

BY Michael Matkowski

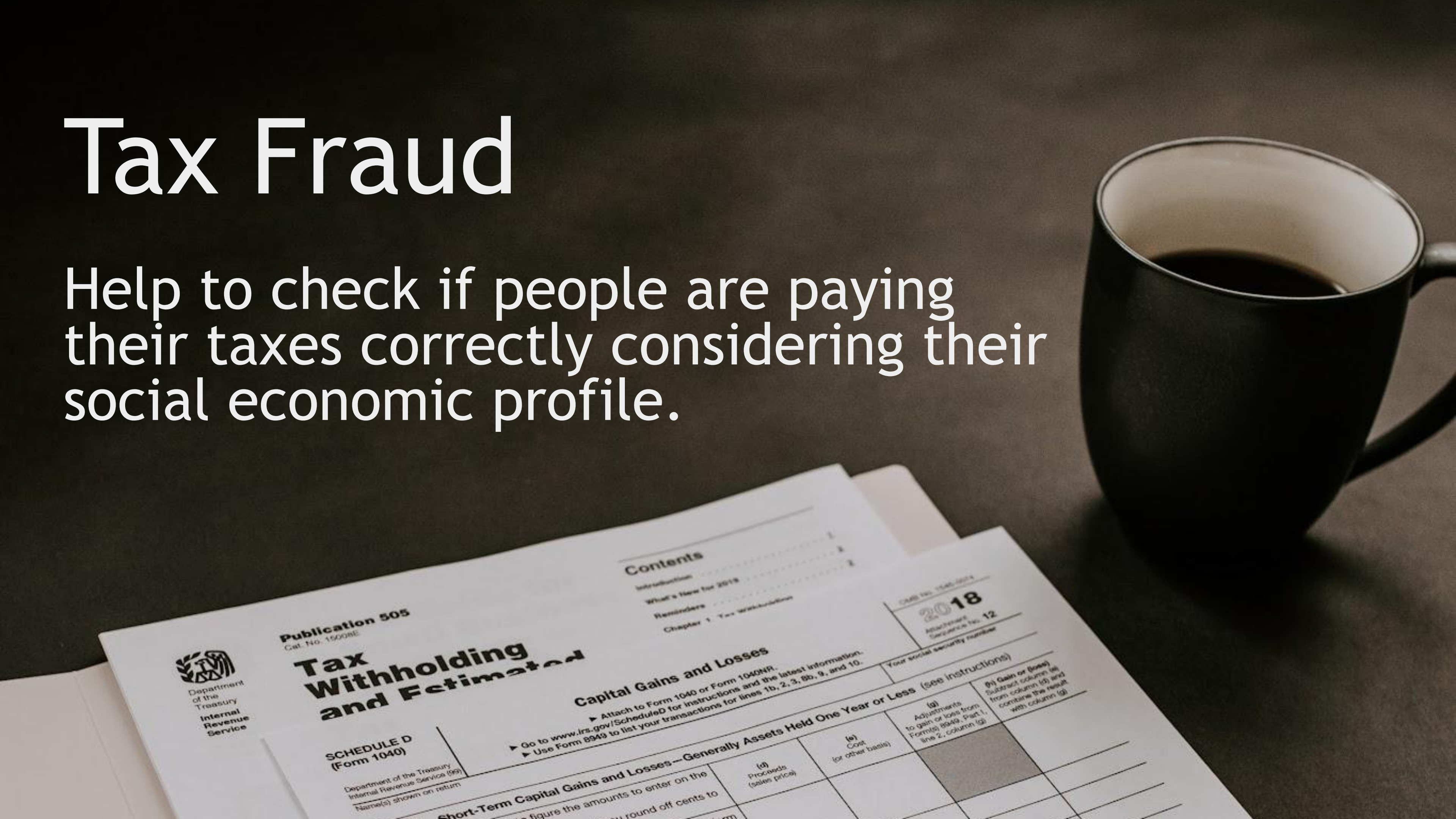
Table 3: Results of Models – R^2 Values

Model	Specification	R^2: Training	R^2: 2019	R^2: 2020
Linear Regression	(Selection from Literature)	0.295	0.289	0.280
LASSO	l1_ratio = 1	0.383	0.374	0.366
Ridge	l1_ratio = 0	0.573	0.565	0.552
Elastic Net	alpha: .1, l1_ratio: 1	0.497	0.484	0.474
Gradient Boosting	n_estimators: 125	0.702	**0.696**	**0.688**
Random Forest	max_features: 15, n_estimators: 100	0.957	0.680	0.666
K-Nearest Neighbors	n_neighbors: 10	0.560	0.400	0.388
Ensemble	Equal Weight – All ML Methods	0.704	0.610	0.600

Potential applications

Tax Fraud

Help to check if people are paying their taxes correctly considering their social economic profile.



Public sector

How a demographic variable impacts salary of a group of people?

For instance, how gender influences income?

Ethical disclaimer

Reinforcement of social biases

Don’t use to discriminate

Data from IBGE
Brazilian Institute of Geography and Statistics

Reputation

Helpful documentation

Data: Summary Statistics

Year	2019 - 2022
Number of observations	582788
Avg Years of Education	10.668318
Percentage of female	41.482151
Race mixed	47.301935
Race white	42.128527
Race black	9.605208
Race asian	0.542736
Race native	0.409926
Race other	0.011668

Data preparation

Convert txt to csv

```
> class Visit: ...

visit2019 = Visit("PNADC_2019_visita1", income_pos=564, eduy_pos=536, workh_pos=601,
                  activity_pos=548, ocupation_pos=550)

visit2020 = Visit("PNADC_2020_visita5", 506, 461, workh_pos=526,
                  activity_pos=473, ocupation_pos=475)

visit2021 = Visit("PNADC_2021_visita5", 489, 461, workh_pos=526,
                  activity_pos=473, ocupation_pos=475)

visit2022 = Visit("PNADC_2022_visita5", 697, 669, workh_pos=734,
                  activity_pos=681, ocupation_pos=683)
```

```
for visit in [visit2019, visit2020, visit2021, visit2022]:
    print("Visit txt name: "+visit.txt_file_name)
    print("Visit csv name: "+visit.csv_file_name)
    counter = 1
    count_valid = 0

    # https://stackoverflow.com/questions/39642082/convert-txt-to-csv-python-script
    with open(visit.txt_file_name, 'r') as in_file, open(visit.csv_file_name, 'w', newline='') as out_file:
        writer = csv.writer(out_file)
        writer.writerow(('YEAR', 'AGE', 'SEX', 'RACE_COLOR', 'STATE', 'HOME_SITUATION', 'AREA_TYPE',
                         'NUM_PEOPLE', 'RELATION_HEAD', 'YEARS_OF_STUDY','WORKED_HOURS' , 'ACTIVITY_GROUP' ,
                         'OCUPATION_GROUP' ,'INCOME', 'LOG_INCOME'))
        group = []

        lines = in_file.read().splitlines()
        for line in lines:
```

Data preparation

Rename categorical values

```python
def decode_activity(activity_code):
    activity_sector_code = {
        '01':'AGRI_FISH_FORESTRY',
        '02':'INDUSTRY',
        '03':'CONSTRUCTION',
        '04':'VEIHICLES_SELL_MAINTENANCE',
        '05':'TRANSPORTATION_WAREHOUSE',
        '06':'HOUSING_FOOD',
        '07':'INFO_COMM_FINANCE_MANAGE',
        '08':'PUBLIC_ADMIN',
        '09':'EDUCATION_HEALTH_SOCIAL',
        '10':'OTHER',
        '11':'DOMESTIC_LABOR',
        '12':'POORLY_DEFINED'
    }
    return activity_sector_code[activity_code]
```

```python
activity_code = line[visit.activity_group_pos-1:
                     visit.activity_group_pos-1+visit.activity_group_len]
row.append(decode_activity(activity_code))
```

Data preparation

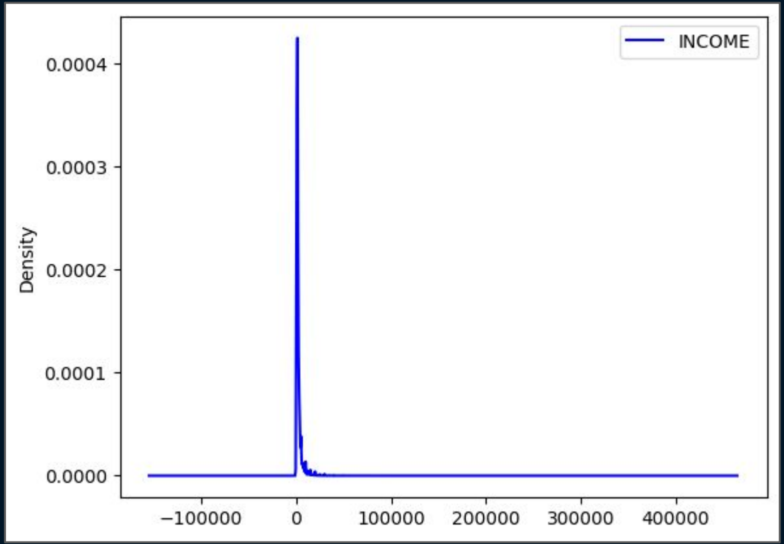
Cleaning

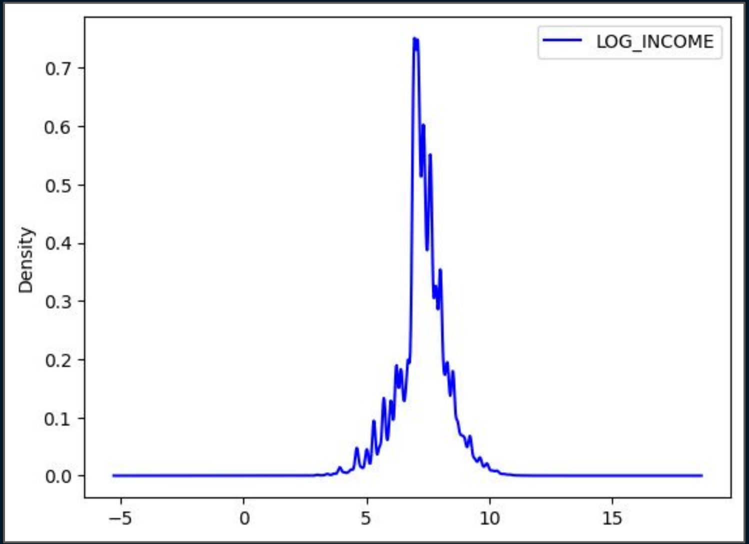
```python
for line in lines:
    # if (count_valid > 30):
    #     break

    income = int(line[visit.income_pos-1:visit.income_pos-1+visit.income_len].strip() or 0)
    if (income <= 0):
        counter = counter + 1
        continue
```

Data preparation

Log normalization

```
log_income = math.log(income)
row.append(log_income)
```





The inputs variables

	Name	Type	Role	Values
1	YEAR	N numeric	feature	
2	AGE	N numeric	feature	
3	SEX	C categorical	feature	F, M
4	RACE_COLOR	C categorical	feature	RDUMMY_ASIAN, RDUMMY_BLACK, RDUMMY_MIXED, RDUMMY_NATIVE, RDUMMY_OTHER, RDUMMY_WHITE
5	STATE	C categorical	feature	SDUMMY_AC, SDUMMY_AL, SDUMMY_AM, SDUMMY_AP, SDUMMY_BA, SDUMMY_CE, SDUMMY_DF, SDUMMY_ES, ...
6	HOME_SITUATI...	C categorical	feature	RURAL, URBAN
7	AREA_TYPE	C categorical	feature	CAPITAL, INTEGRATED, METROPOLITAN, NON_METRO_INTEG
8	NUM_PEOPLE	N numeric	feature	
9	RELATION_HEAD	C categorical	feature	CHILD_BOTH, CHILD_HEAD, DOMESTIC_EMPLOYEE, DOMESTRIC_EMPLOYEE_RELATIVE, FREE_NON_RELATIVE, GRANDCHILD,...
10	YEARS_OF_STU...	N numeric	feature	
11	WORKED_HOURS	N numeric	feature	
12	ACTIVITY_GROUP	C categorical	feature	AGRI_FISH_FORESTRY, CONSTRUCTION, DOMESTIC_LABOR, EDUCATION_HEALTH_SOCIAL, HOUSING_FOOD, INDUSTRY, ...
13	OCUPATION_G...	C categorical	feature	ARMY, DIRECTOR_MANAGER, ELEMENTARY, MACHINERY_OPERATOR, MANAGEMENT_SUPPORT, POORLY_DEFINED, ...
14	INCOME	N numeric	**skip**	
15	**LOG_INCOME**	N **numeric**	**target**	

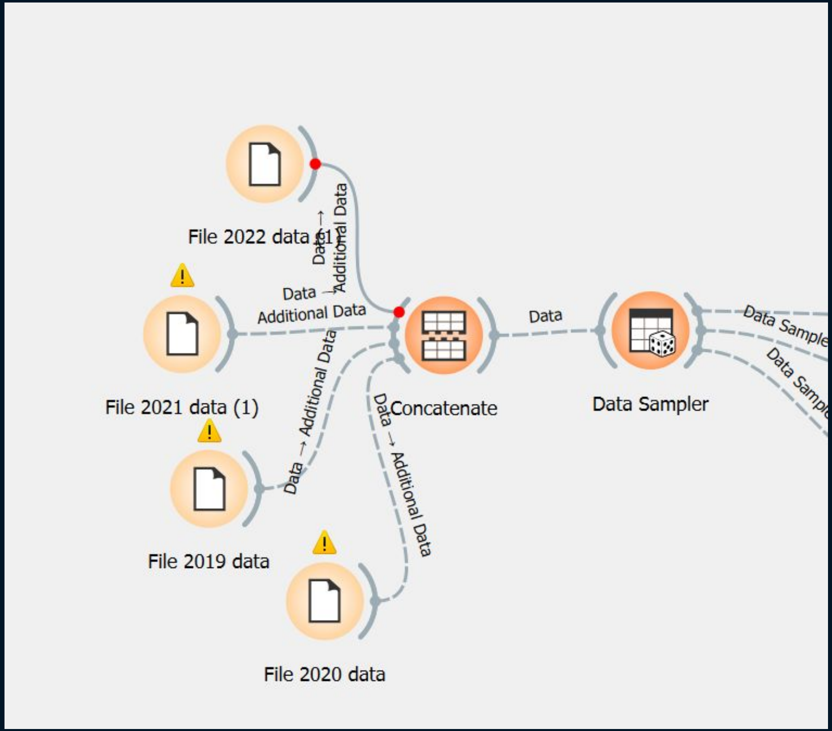
Data preparation

Pre-processing with Orange

Concatenate different datasets

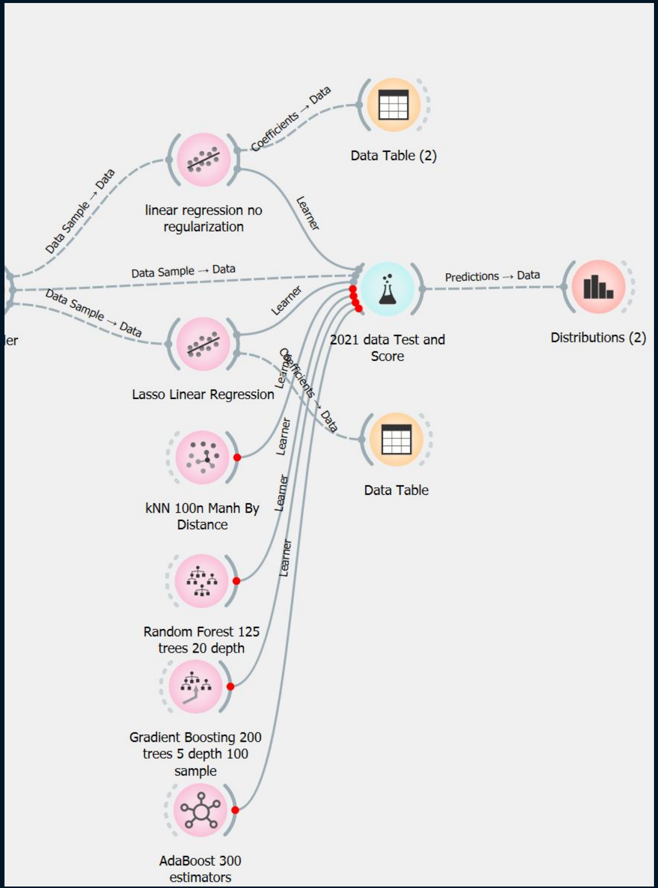
Data Sampler 20% - used to make training faster during parametrization

Embedded preprocessor in the learners: one-hot encoding of categorical input.



Machine learning models

1. Linear regression without regularization
2. LASSO - Linear regression
3. k-NN
4. Random forest
5. Gradient Boosting
6. AdaBoost



Machine learning models: simple learners

- Linear regression without regularization
- LASSO - Linear regression
  - Regularization strength 0.01
- k-NN
  - 100 neighbors
  - Manhattan metric
  - Weight by distances

Machine learning models: ensemble learners

- Random forest
  - 125 trees
  - Limit depth of individual trees: 20
  - Minimum sample size to split: 5
- Gradient Boosting
  - 200 trees
  - Learning rate 0.1
  - Limit depth of individual trees: 5
  - Minimum sample size to split: 100
- AdaBoost
  - 300 estimators (trees)
  - Learning rate: 1

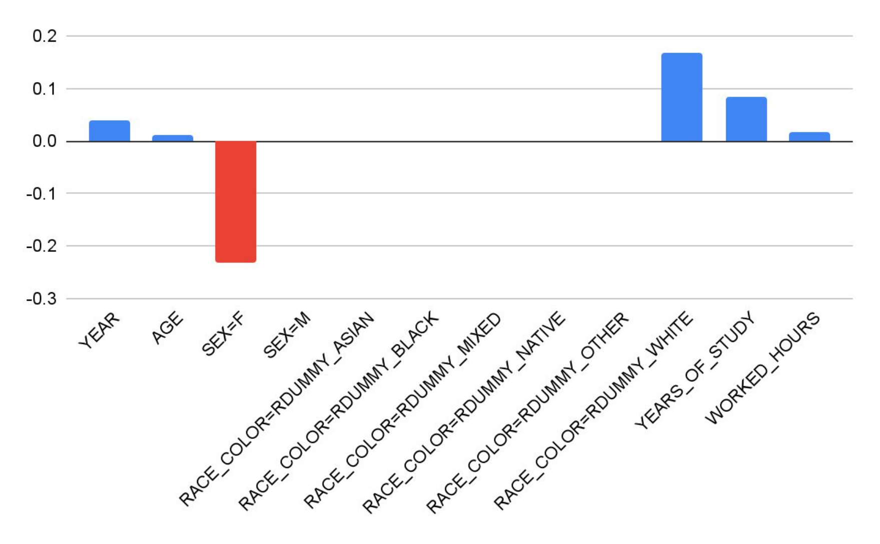
RESULT: Evaluation metrics using 20% of the samples

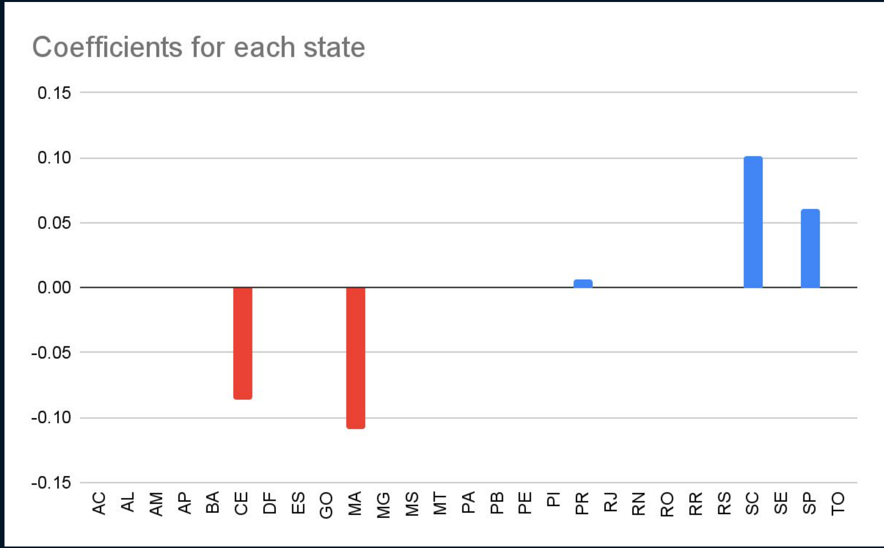
Model	MSE	MAE	R2 Coefficient of determination
Linear Regression	0.455	0.501	0.511
Lasso Linear Regression	0.507	0.527	0.456
kNN	0.476	0.503	0.489
Random Forest	0.418	0.471	0.552
Gradient Boosting	0.386	0.454	0.585
AdaBoost	0.415	0.465	0.554

RESULT: Evaluation metrics using 100% of the samples

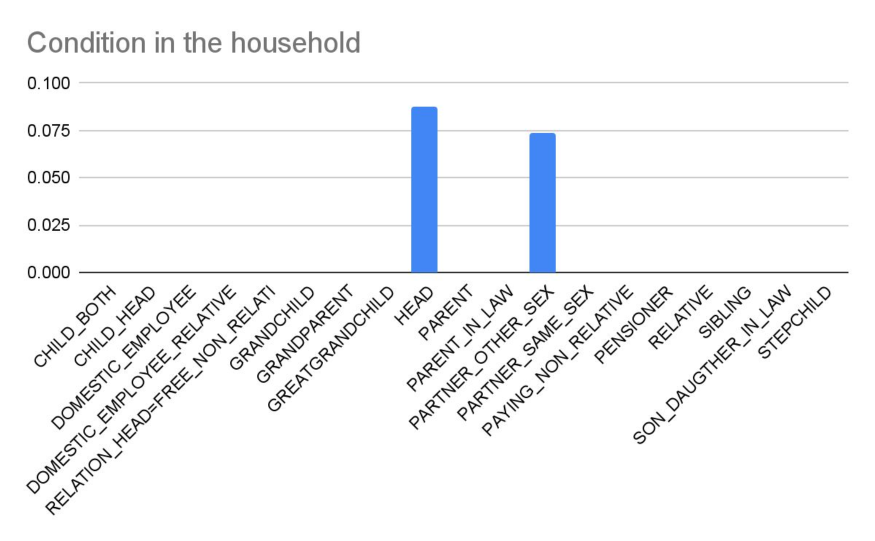
Model	MSE	MAE	R2 Coefficient of determination
Linear Regression	0.456	0.501	0.506
Lasso Linear Regression	0.509	0.526	0.45
kNN	0.447	0.486	0.516
Random Forest	0.396	0.455	0.571
Gradient Boosting	0.381	0.449	0.588
AdaBoost	0.421	0.474	0.545

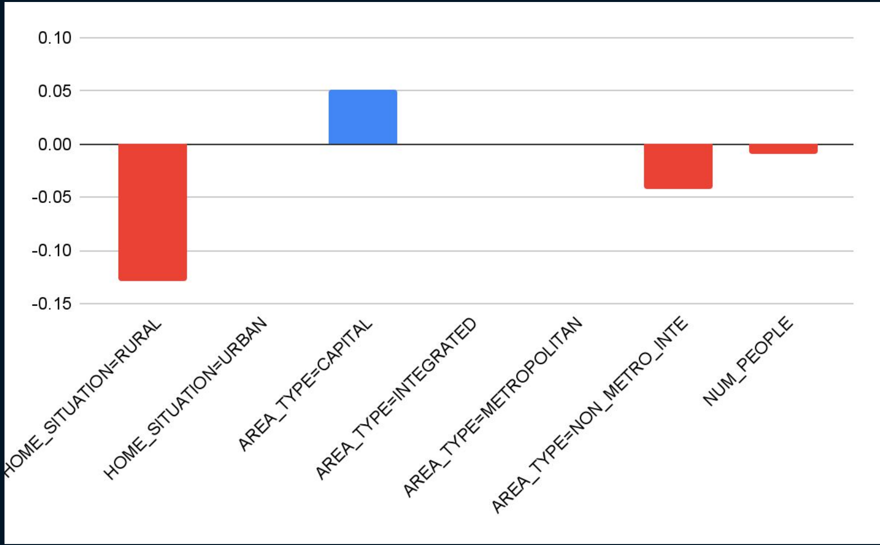
Interpretability: Lasso coefficients



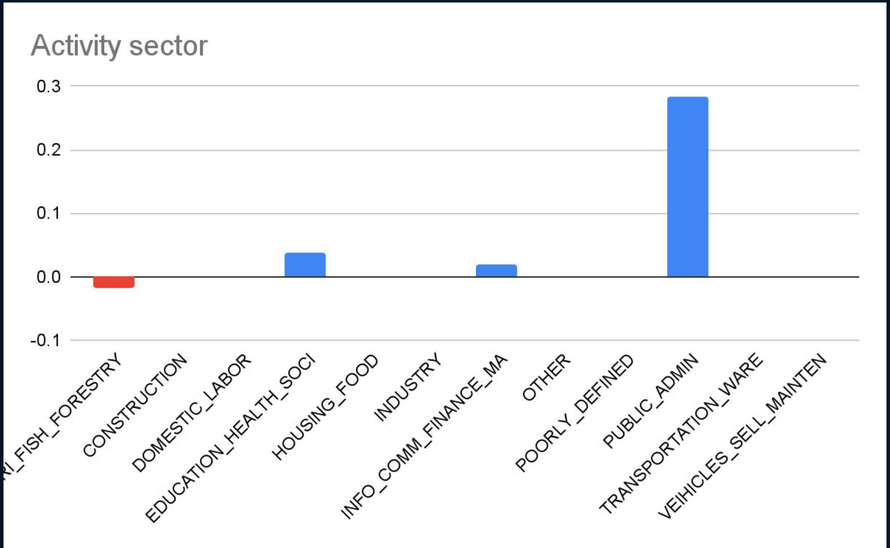


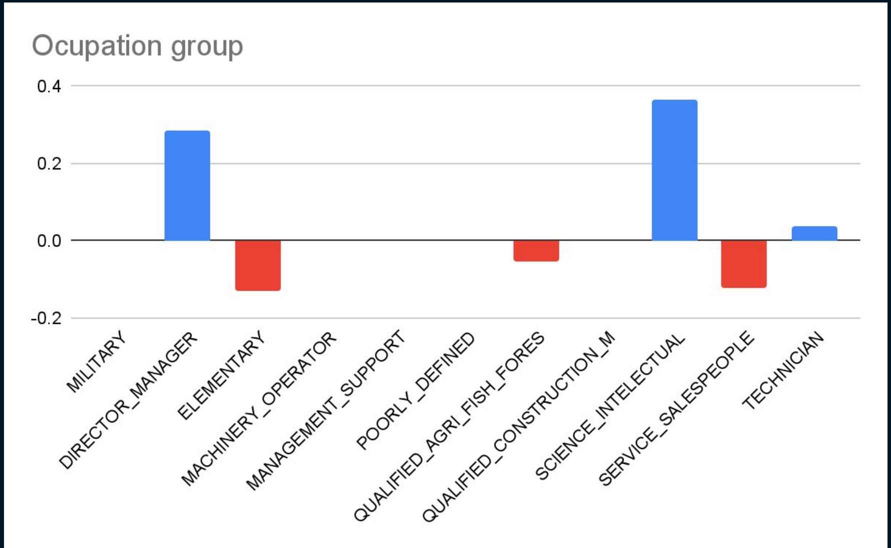
Interpretability: Lasso coefficients





Interpretability: Lasso coefficients





Further improvements

- Add more features: Do they receive benefits (social assistance program)? Is the person employed? etc…
- Visualize the coefficients related to geospatial data in a map
- Preprocessing: change normalization of inputs, feature ranking, feature extraction, outliers detection, etc
- Approach the problem using classification:
  - Target: income above threshold, income below or equal threshold
  - Target: range of income: 0-1 minimum wage, 2-5 minimum wage, 5+ minimum wage

References

- IBGE Data
- "Prediction of Individual Income: A Machine Learning Approach" by Michael Matkowski
- Coding Systems for Categorical Variables in Regression Analysis
- Log normalization | Python
- Linear Regression — Orange Visual Programming 3 documentation
- Linear regression analysis using orange, Missing value treatment, outlier, Normality, box plot draw
- Regression — Orange Data Mining Library 3 documentation

THANKS

Rebeca Nunes
al435871@uji.es