Data Mining Project

O projeto utilizará a metodologia CRISP-DM.

Grupo 11

- Bruno Aurélio Rôzza de Moura Campos (14104255)
- Ricardo Carvalho Maisonnave (13101303)

Código Auxiliar para Ocultar o Código no Jupyter

Click aqui para mostrar/ocultar o código

Business Understanding

Business Objectives

Uma organização sem fins lucrativos que sobrevive de captação de recursos irá começar a entrar em contato com as pessoas para solicitar doações. Contudo, há poucos colaboradores e as últimas tentativas não deram um resultado esperado. Por isso, a organização decidiu fazer um projeto de data mining tendo como **critério de sucesso do projeto** uma melhor assertividade na captação financeira.
Saber a remuneração de um indivíduo pode ajudar a organização (já que é possível redirecionar uma porcentagem do Imposto de Renda para doações atraves do FIA - Fundo da Infancia e Adolescência) a fazer os pedidos mais adequado para uma solicitação de apoio e colaboração, ou ainda se eles realmente deveriam entrar em contato com a pessoa, então este é o **critério de sucesso da mineração**.

Assess Situation

- Pessoal disponível: Bruno Campos (Cientista de Dados)
- Recursos computacionais: O conjunto de dados a ser explorado é o Census Income Data Set hospedados em Repositório de Machine Learning UCI.

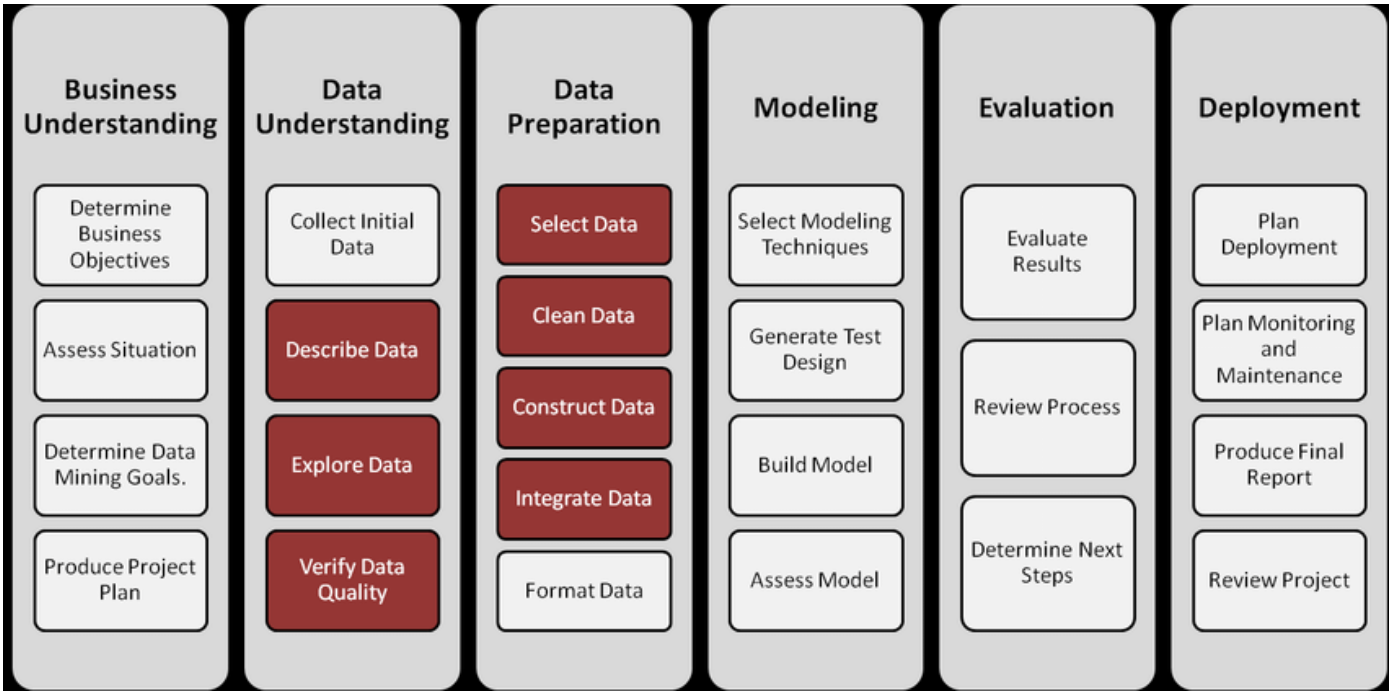
Data Mining Goals

O objetivo, é construir um modelo que pode predizer se um indivíduo possui uma remuneração superior a $50,000.

Então como **artefatos de saída** será uma classificação para quais pessoas se deve entrar em contato, pessoas que ganham acima de $50,000, para pedir doações.

Project Plan

O projeto seguirá as phases e tasks da metodologia CRISP-DM.



- Serão exploradas técnicas de aprendizado supervisonado.
- Como recursos serão utilizados um computador pessoal, internet, material das aulas de matéria de data mining, um Sistema Operacional, um navegador, um ambienten virtual de Python, bibliotecas (detalhadas no arquivo `requirements.txt`).
- Os riscos: não ter energia elétrica, não ter sinal de internet, computador pessoal quebrar, falta de conhecimento em data mining.
- Pontos críticos do projeto: as principais ferramentas para o trabalho serão a linguagem Python e suas bibliotecas.
- Como técnicas preliminares serão utilizadas:
  - análise exploratória dos dados
  - preprocessamento dos dados aplicando técnicas como one-hot-enconding, subsampling, normalização de fetures
  - modelagem preditiva utilizando árvores de decisão e máquinas de suporte vetorial
  - avaliação dos resultados através de metricas como acurácia

---

Data Understanding

Collect Initial Data

- Os dados foram coletados no censo americano de 1994.
- O conjunto de dados a ser explorado é o **Census Income Data Set** hospedados em [Repositório de Machine Learning UCI (https://archive.ics.uci.edu/ml/datasets/Census+Income)](). Neste trabalho foi adaptado para o csv já conter o nome das colunas.

```
'Current working directory: /home/campos/projects/artificial_inteligen
ce/data_science_projects/encontrar_doadores'


CPU times: user 109 ms, sys: 11.9 ms, total: 121 ms
Wall time: 147 ms
```

Describe Data

```
Dataframe:
45222 rows
14 columns
```

	age	workclass	education_level	education-num	marital-status	occupation	relationship	race	se
0	39	State-gov	Bachelors	13.00	Never-married	Adm-clerical	Not-in-family	White	Ma
1	50	Self-emp-not-inc	Bachelors	13.00	Married-civ-spouse	Exec-managerial	Husband	White	Ma
2	38	Private	HS-grad	9.00	Divorced	Handlers-cleaners	Not-in-family	White	Ma
3	53	Private	11th	7.00	Married-civ-spouse	Handlers-cleaners	Husband	Black	Ma
4	28	Private	Bachelors	13.00	Married-civ-spouse	Prof-specialty	Wife	Black	Fema

Dataset General Information

- O conjunto de dados do census esta no formato csv
- Tem 45222 linhas e 14 colunas

Features

- `age` : Idade
- `workclass` : Classe trabalhadora (Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked)
- `education_level` : Nível de educação (Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool)
- `education-num` : Número de anos de estudo concluídos
- `marital-status` : Estado civil (Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse)
- `occupation` : Ocupação profissional (Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces)

- `relationship` : Status de relacionamento (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
- `race` : Raça (White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black)
- `sex` : Sexo (Female, Male)
- `capital-gain` : Ganhos de capital monetário
- `capital-loss` : Perdas de capital monetário
- `hours-per-week` : Média de horas trabalhadas por semana
- `native-country` : País de origem (United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands)

Target

- A variáveis alvo é: `income`

Explore Data

Primeiramente vamos explorar quais são os tipos das colunas.

- Colunas numéricas:

```
['age', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-we
ek']
```

- Colunas categóricos:

```
['workclass',
 'education_level',
 'marital-status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native-country',
 'income']
```

Measure Location

```
------------------------ MEASURES OF LOCALIZATION ------------------
------

TOTAL columns [<class 'numpy.number'>, <class 'object'>]: 14
PERCENTAGE [<class 'numpy.number'>, <class 'object'>] in dataframe: 10
0.0 %
```

	age	workclass	education_level	education-num	marital-status	occupation	relationship	r
count	45222.00	45222	45222	45222.00	45222	45222	45222	45
unique	NaN	7	16	NaN	7	14	6	
top	NaN	Private	HS-grad	NaN	Married-civ-spouse	Craft-repair	Husband	W
freq	NaN	33307	14783	NaN	21055	6020	18666	38
mean	38.55	NaN	NaN	10.12	NaN	NaN	NaN	N
std	13.22	NaN	NaN	2.55	NaN	NaN	NaN	N
min	17.00	NaN	NaN	1.00	NaN	NaN	NaN	N
25%	28.00	NaN	NaN	9.00	NaN	NaN	NaN	N
50%	37.00	NaN	NaN	10.00	NaN	NaN	NaN	N
75%	47.00	NaN	NaN	13.00	NaN	NaN	NaN	N
max	90.00	NaN	NaN	16.00	NaN	NaN	NaN	N

```
------------------------ MEASURES OF LOCALIZATION ------------------
------

TOTAL columns [<class 'numpy.number'>]: 5
PERCENTAGE [<class 'numpy.number'>] in dataframe: 35.71 %
```

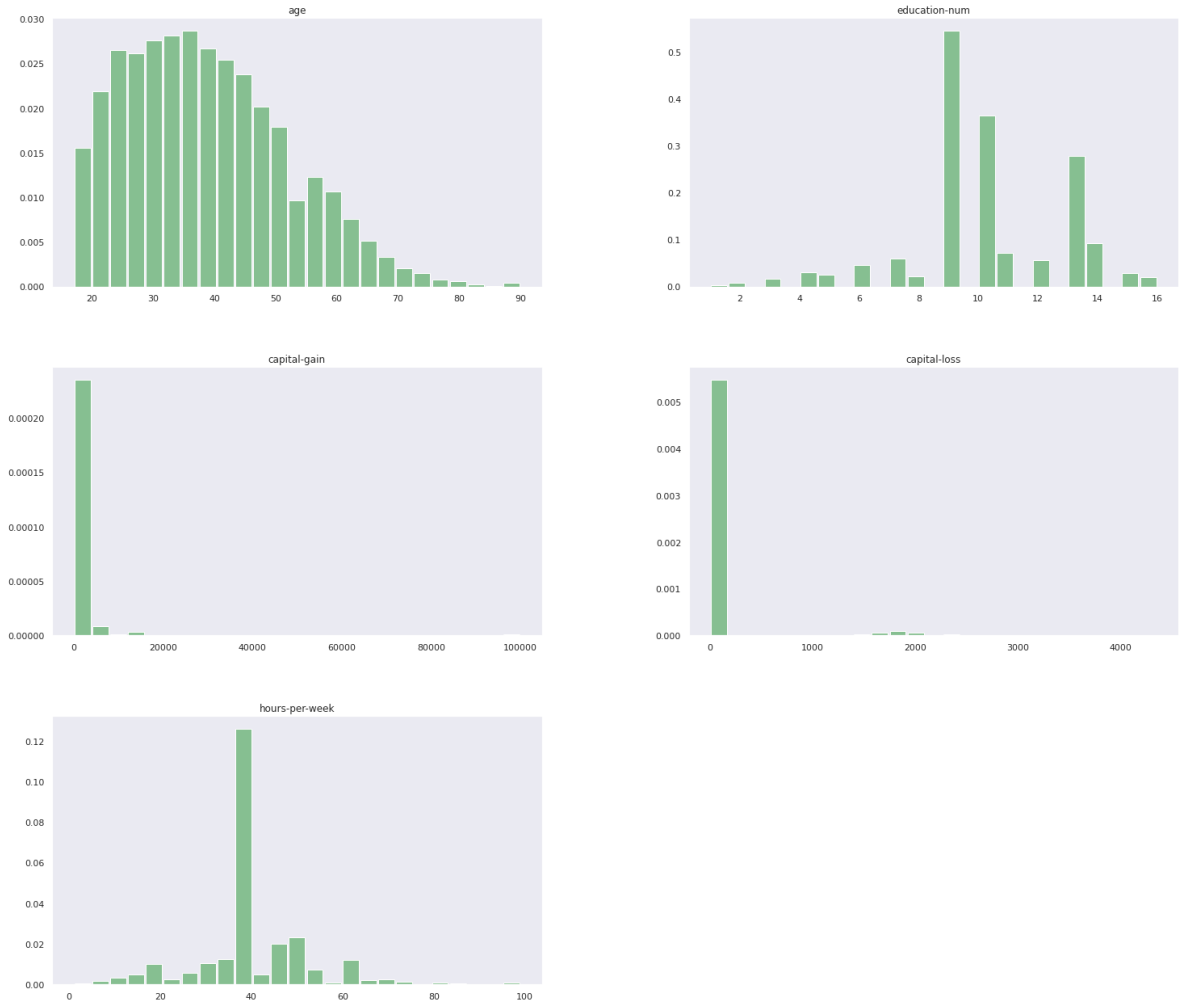
	age	education-num	capital-gain	capital-loss	hours-per-week
count	45222.00	45222.00	45222.00	45222.00	45222.00
mean	38.55	10.12	1101.43	88.60	40.94
std	13.22	2.55	7506.43	404.96	12.01
min	17.00	1.00	0.00	0.00	1.00
25%	28.00	9.00	0.00	0.00	40.00
50%	37.00	10.00	0.00	0.00	40.00
75%	47.00	13.00	0.00	0.00	45.00
max	90.00	16.00	99999.00	4356.00	99.00

```
------------------------- MEASURES OF LOCALIZATION -------------------
------

TOTAL columns [<class 'object'>]: 9
PERCENTAGE [<class 'object'>] in dataframe: 64.29 %
```

	workclass	education_level	marital-status	occupation	relationship	race	sex	native-country
count	45222	45222	45222	45222	45222	45222	45222	45222
unique	7	16	7	14	6	5	2	41
top	Private	HS-grad	Married-civ-spouse	Craft-repair	Husband	White	Male	United-States
freq	33307	14783	21055	6020	18666	38903	30527	41292

Histograms

Nas imagens abaixo se tem os histogramas para os dados numéricos:



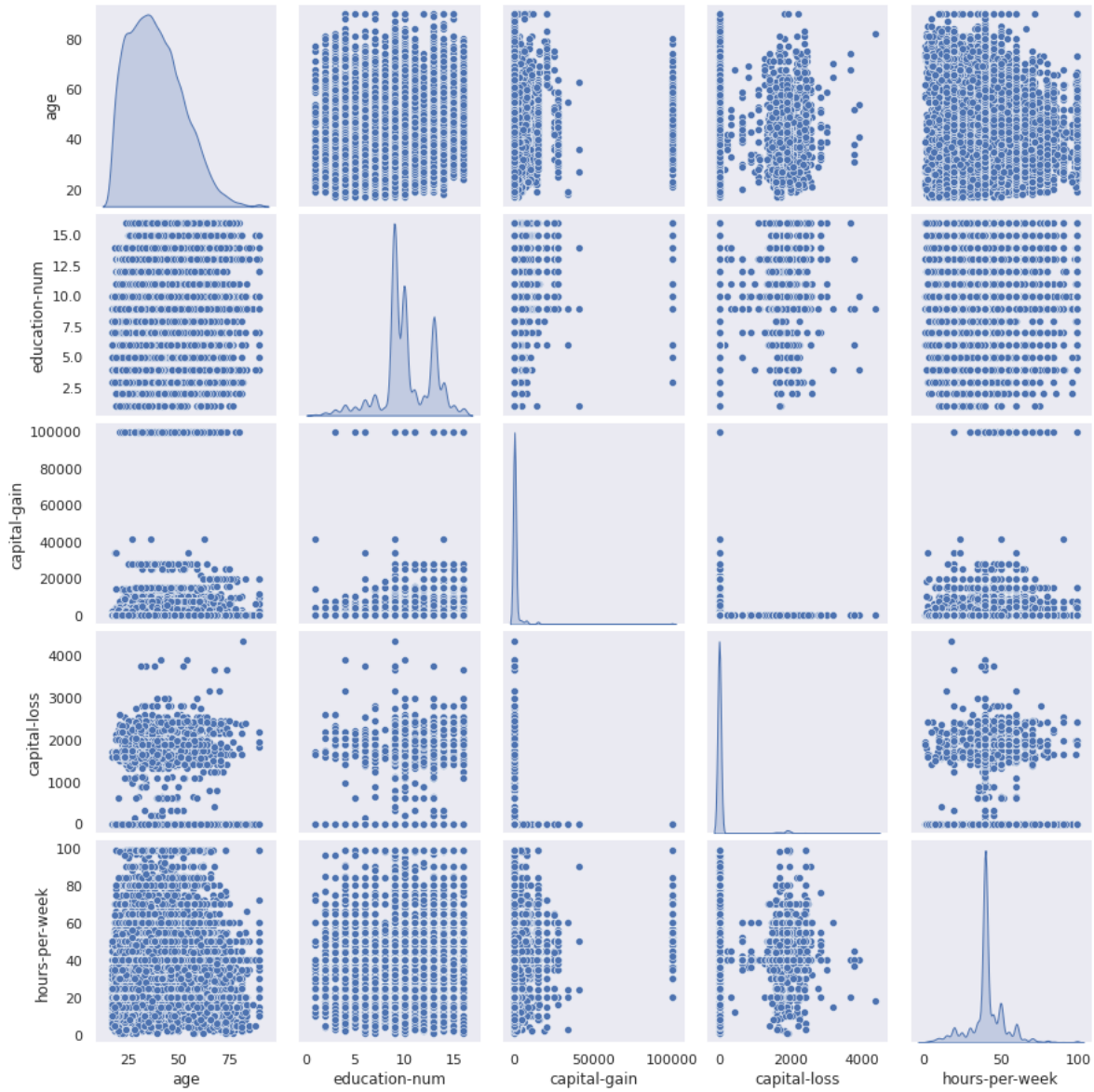
Análise

O `capital-gain` e `capital-loss` aprensentam uma grande distorção quanto a distribuição dos valores.

Correlations

Nas imagens abaixo se tem os gráficos e matriz de correlação para os dados numéricos:

Observations



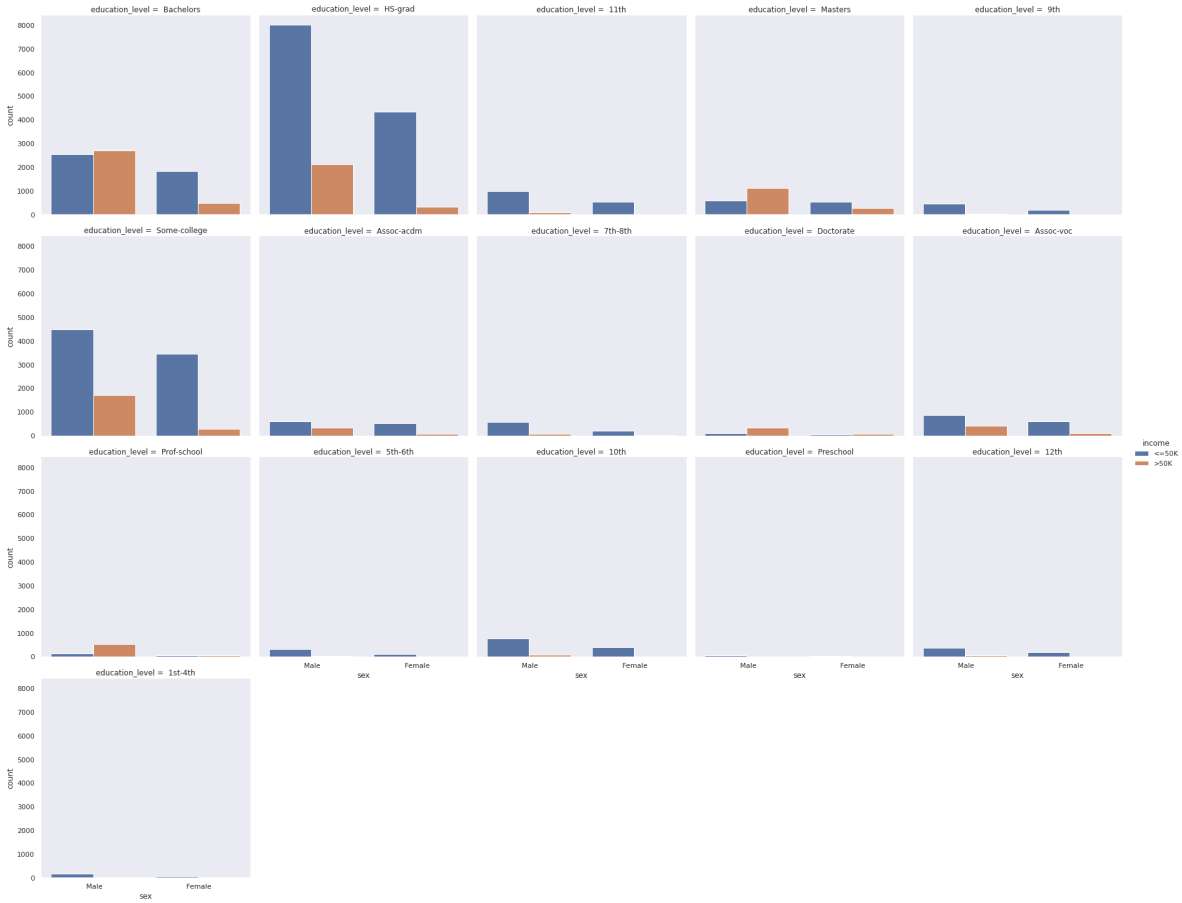
- Número de registros com remuneração anual superior à $50,000:

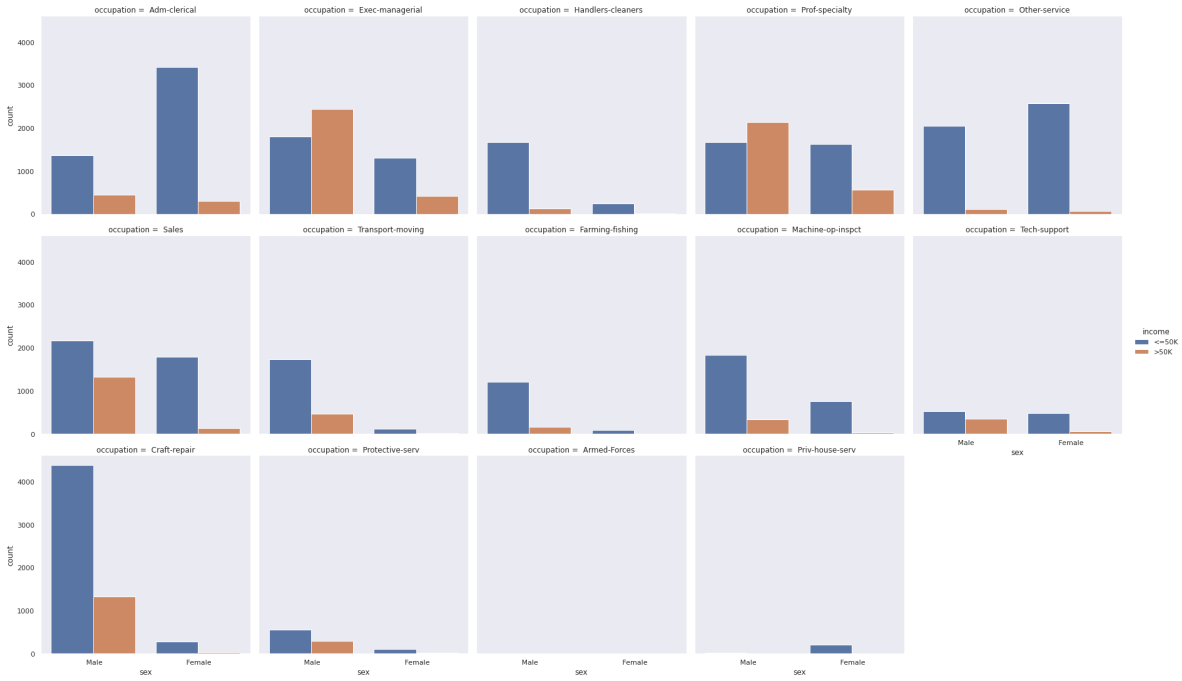
```
11208
```

- O percentual de indivíduos com remuneração anual superior à $50,000

```
24.78439697492371
```

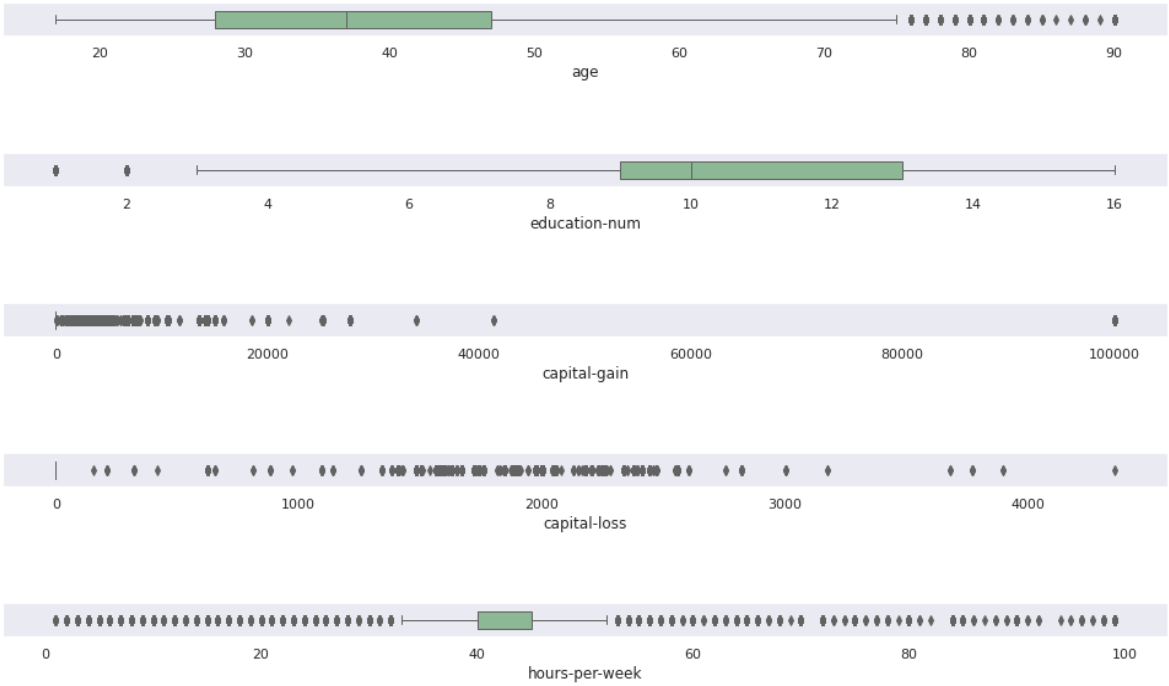
- Um ponto a ser analisado é a relação entre as features de um indivíduo com sua renda:
  - Grupo de gráfico 01: contagem de pessoas que ganham acima ou abaixo de 50 mil com base em seu sexo e educação





- Grupo de gráficos 02: contagem de pessoas que ganham acima ou abaixo de 50 mil com base em seu sexo e ocupação

- Análise da distribuição dos dados
  - Abaixo é analisado de forma visual os outliers com auxilio de box-plot



NOTAS

Ao entender o problema e explorar os dados é possível dizer que os 5 atributos que são mais importantes para predição são estes:

- `ocupação` : diferentes empregos têm diferentes escalas de pagamento. Alguns empregos pagam mais do que outros.
- `educação` : as pessoas que concluíram um nível superior de educação estão mais bem equipadas para realizar trabalhos mais técnicos / especializados e bem remunerados.
- `Idade` : a medida que as pessoas envelhecem, elas acumulam mais bem-estar.
- `classe de trabalho` : a classe trabalhadora a que pertencem também pode ser correlacionada com quanto dinheiro eles ganham.
- `horas por semana` : quem trabalha mais horas por semana, provavelmente ganhará mais.

Verify Data Quality

1. Análise do carregamento do dataset e seus metadados

- Verificar se há linhas faltantes
- Verificar se há dados faltantes nas células
- Verificar se há linhas duplicadas

```
RangeIndex(start=0, stop=45222, step=1)
```

- O dataset não possui valores de índices faltantes, logo não há linhas faltantes.

```
The dataframe NOT contains missing values.
```

- O dataset não possui valores faltantes em suas células, logo não há dados faltantes.

```
------------------------ DUPLICATED DATA ------------------------

SHAPE of data: 45222
TOTAL duplicated data: 5982
PERCENTAGE duplicated data: 1.3e+01 %
```

- O dataset possui 5922 linhas duplicadas.

NOTAS

2. Aderência dos dados com o domínio do problema

- Todas as colunas podem influenciar o salário de uma pessoas, então é possível afirmar que as colunas estão de acordo com o domínio do problema.
- É necessário analisar o conteúdo dos dados para saber se realmente são relevantes e se enquadram dentro da coluna que estão.

```
The categorical column workclass contains this values:

[' State-gov' ' Self-emp-not-inc' ' Private' ' Federal-gov' ' Local-go
v'
 ' Self-emp-inc' ' Without-pay']

The categorical column education_level contains this values:

[' Bachelors' ' HS-grad' ' 11th' ' Masters' ' 9th' ' Some-college'
 ' Assoc-acdm' ' 7th-8th' ' Doctorate' ' Assoc-voc' ' Prof-school'
 ' 5th-6th' ' 10th' ' Preschool' ' 12th' ' 1st-4th']

The categorical column marital-status contains this values:

[' Never-married' ' Married-civ-spouse' ' Divorced'
 ' Married-spouse-absent' ' Separated' ' Married-AF-spouse' ' Widowe
d']

The categorical column occupation contains this values:

[' Adm-clerical' ' Exec-managerial' ' Handlers-cleaners' ' Prof-specia
lty'
 ' Other-service' ' Sales' ' Transport-moving' ' Farming-fishing'
 ' Machine-op-inspct' ' Tech-support' ' Craft-repair' ' Protective-ser
v'
 ' Armed-Forces' ' Priv-house-serv']

The categorical column relationship contains this values:

[' Not-in-family' ' Husband' ' Wife' ' Own-child' ' Unmarried'
 ' Other-relative']

The categorical column race contains this values:

[' White' ' Black' ' Asian-Pac-Islander' ' Amer-Indian-Eskimo' ' Othe
r']

The categorical column sex contains this values:

[' Male' ' Female']

The categorical column native-country contains this values:

[' United-States' ' Cuba' ' Jamaica' ' India' ' Mexico' ' Puerto-Rico'
 ' Honduras' ' England' ' Canada' ' Germany' ' Iran' ' Philippines'
 ' Poland' ' Columbia' ' Cambodia' ' Thailand' ' Ecuador' ' Laos'
 ' Taiwan' ' Haiti' ' Portugal' ' Dominican-Republic' ' El-Salvador'
 ' France' ' Guatemala' ' Italy' ' China' ' South' ' Japan' ' Yugoslav
ia'
 ' Peru' ' Outlying-US(Guam-USVI-etc)' ' Scotland' ' Trinadad&Tobago'
 ' Greece' ' Nicaragua' ' Vietnam' ' Hong' ' Ireland' ' Hungary'
 ' Holand-Netherlands']

The categorical column income contains this values:

['<=50K' '>50K']
```

NOTAS

Todos as categorias estão condizentes com as suas respectivas colunas.

Data Preparation

O primeiro passo será a criação de um dataframe para a variável alvo e outro com somente as fetures.

In [669]:

```
income_raw.head()
```

```
0    <=50K
1    <=50K
2    <=50K
3    <=50K
4    <=50K
Name: income, dtype: object
```

In [670]:

```
features_raw.head()
```

	age	workclass	education_level	education-num	marital-status	occupation	relationship	race	se
0	39	State-gov	Bachelors	13.00	Never-married	Adm-clerical	Not-in-family	White	Ma
1	50	Self-emp-not-inc	Bachelors	13.00	Married-civ-spouse	Exec-managerial	Husband	White	Ma
2	38	Private	HS-grad	9.00	Divorced	Handlers-cleaners	Not-in-family	White	Ma
3	53	Private	11th	7.00	Married-civ-spouse	Handlers-cleaners	Husband	Black	Ma
4	28	Private	Bachelors	13.00	Married-civ-spouse	Prof-specialty	Wife	Black	Fema

Select Data

Conforme visto em Verify Data Quality, todos as colunas e observações estão de acordo com o problema, por isso será utilizado todo o dataset neste momento.

	age	workclass	education_level	education-num	marital-status	occupation	relationship	race	
45221	35	Self-emp-inc	Bachelors	13.00	Married-civ-spouse	Exec-managerial	Husband	White	M

Clean Data

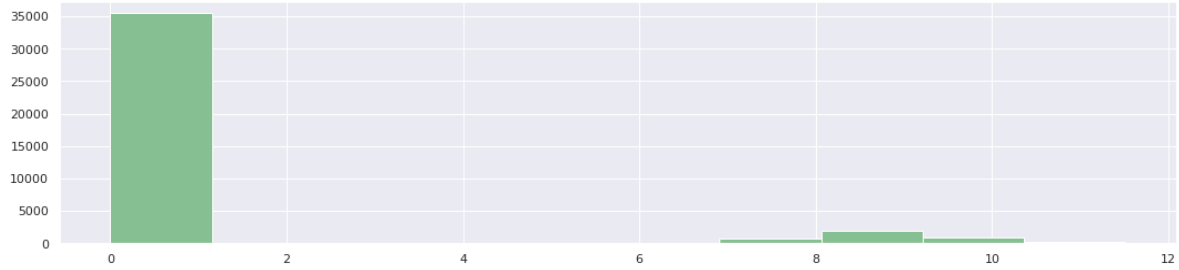
- Conforme visto anteriormente há dados duplicados que precisam ser tratados.

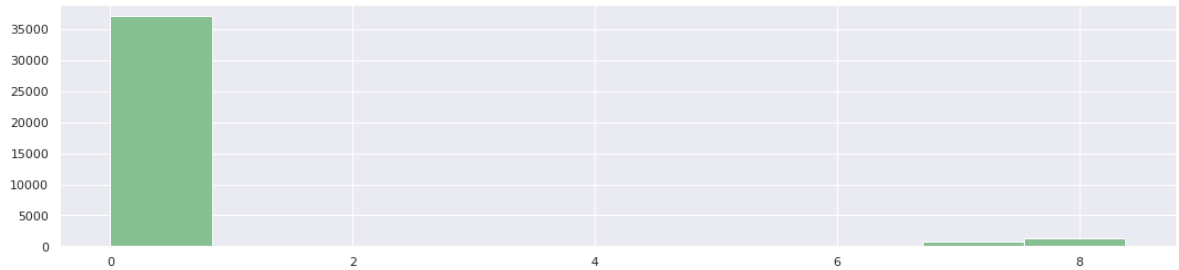
```
Size dataframe =  (39240, 14)
```

Construct Data

Continuos Data

- Conforme analisado anteriormente, há dados contínuos que aprensentam distorções. É necessário transformar o intervalo de valores para não afetarem o desempenho de um algoritmo de aprendizado. Para corrigir isso, será aplicado uma transformação logarítmica.





A normalização dos dados contínuos garante que cada feature seja tratado igualmente por um algortimo de aprendizado. Por isso, será aplicado a ténica de minmax scalar.

	age	workclass	education_level	education-num	marital-status	occupation	relationship	race	sex
0	0.30	State-gov	Bachelors	0.80	Never-married	Adm-clerical	Not-in-family	White	Male

NOTAS

Há várias colunas que não são numéicas. Geralmente, os algoritmos de aprendizagem esperam que a entrada seja numérica. As features não numéricos, *categóricas*, precisam ser convertidas. Utilizaremos **one-hot code**.

	age	education-num	capital-gain	capital-loss	hours-per-week	workclass_ Federal-gov	workclass_ Local-gov	workclass_ Private	workclass_ Self-emp-inc
0	39	13.00	2174.00	0.00	40.00	0	0	0	0
1	50	13.00	0.00	0.00	13.00	0	0	0	0
2	38	9.00	0.00	0.00	40.00	0	0	1	0
3	53	7.00	0.00	0.00	40.00	0	0	1	0
4	28	13.00	0.00	0.00	40.00	0	0	1	0

```
103 total features after one-hot encoding.
```

Para simplificar a variável alvo vou usar somente uma coluna chamada **income** que só tem valores 0 e 1.

```
0    0
1    0
2    0
3    0
4    0
5    0
6    0
7    1
8    1
9    1
Name: income, dtype: int64
```

Divisão dos dados

```
Training set has 36177 samples.
Testing set has 9045 samples.
```

Pipeline de Treinamento e Avaliação

Árvore de Decisão: baseline

Como uma árvore de decisão pode lidar com dados numéricos e categóricos, é uma boa candidata para este tipo de problema. Outro fator relevante é que uma árvore de decisão é fácil de interpretar, ou seja, saberemos o que acontece nos bastidores para interpretar os resultados.

```
Begin training DecisionTreeClassifier()...

Time Elapsed = 0.37488245964050293 seconds


{'acc_test': 0.819126589275843, 'f_test': 0.617413525600073}
```

Máquina de Suporte Vetorial

A SVM foi escolhida devido à sua eficácia para uma alta dimensionalidade. Depois de incorporar variáveis fictícias, temos mais de 100 colunas em nosso conjunto de dados então se torna um bom cenário para testar uma SVM. Além disso, o conjunto de dados não é tão grande para ser um impedimento.

```
Begin training SVC()...

Time Elapsed = 113.49529767036438 seconds


{'acc_test': 0.7966832504145936, 'f_test': 0.3057516611295681}
```

AdaBoost

Os métodos de ensemble, como por exemplo o Adaboost, são mais robustos do que os previsores únicos pois garantem uma boa generalização.

```
Begin training AdaBoostClassifier()...

Time Elapsed = 2.820767641067505 seconds


{'acc_test': 0.8576008844665561, 'f_test': 0.6299100804543303}
```

Análise de Resultados

- Olhando os resultados acima, dos três modelos, AdaBoost é o mais adequado para nossa tarefa. É o classificador que tem o melhor desempenho nos dados de teste, tanto em termos de precisão quanto de f-score.
- O AdaBoost foi o treinamento mais rápido, enquanto que o treinamento mais lento foi a SVM.

Model Tuning

Vou aplicar uma **GridSearchCV** com diferentes combinações de hiperparâmetro afim de ajustar o modelo e obter resultados ainda melhores.

Para Adaboost será ajustado os seguintes hiperparâmetro:

- n_estimators
- taxa de aprendizagem
- parâmetros do baseline (árvore de decisão)

```
CPU times: user 14min 36s, sys: 620 ms, total: 14min 37s
Wall time: 14min 41s
```

```
Begin training AdaBoostClassifier(base_estimator=DecisionTreeClassifie
r(max_depth=2,
                                                       min_samples_s
plit=6),
                   learning_rate=0.5, n_estimators=150)...

Time Elapsed = 10.8744215965271 seconds
```

```
{'acc_test': 0.8709784411276948, 'f_test': 0.6739252862774545}
```

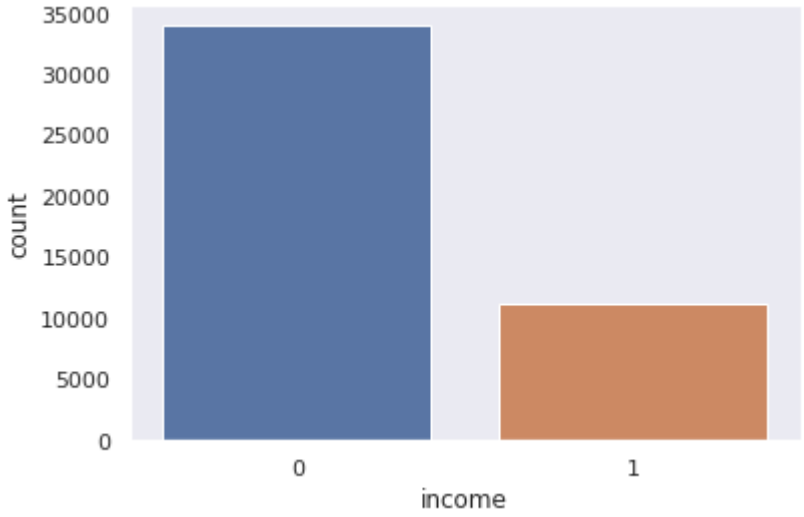
```
Optimized Model
------
Final accuracy: 0.8710
Final F-score:  0.7525
AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=2,
                                                       min_samples_s
plit=6),
                   learning_rate=0.5, n_estimators=150)
```

NOTAS

- O modelo otimizado tem uma precisão de 0,8710 e F-score de 0,7525.
- A Otimização do modelo se sobresaiu em relação ao modelo preliminar.

Análise da Distribuição da Variável Alvo

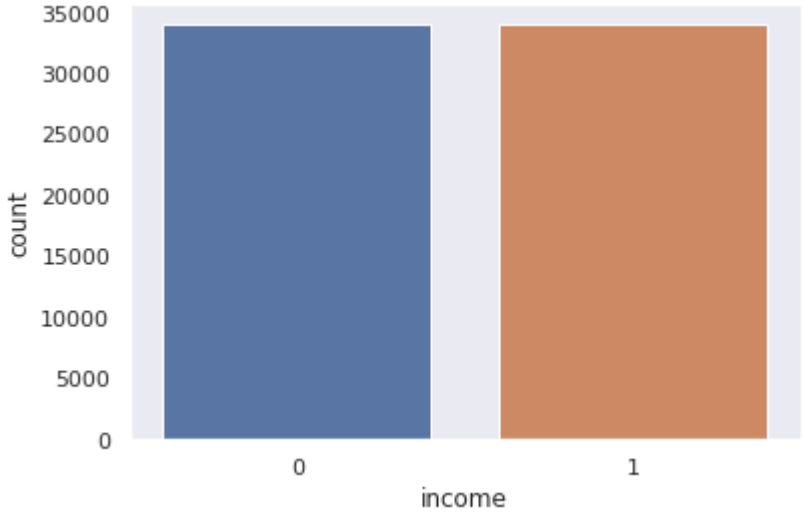
```
Distribution data = [34014 11208]
```



É possível notar que os dados estão desbalanceados. Vou testar a técnica de under e oversample conforme recomendação do Professor.

Oversample

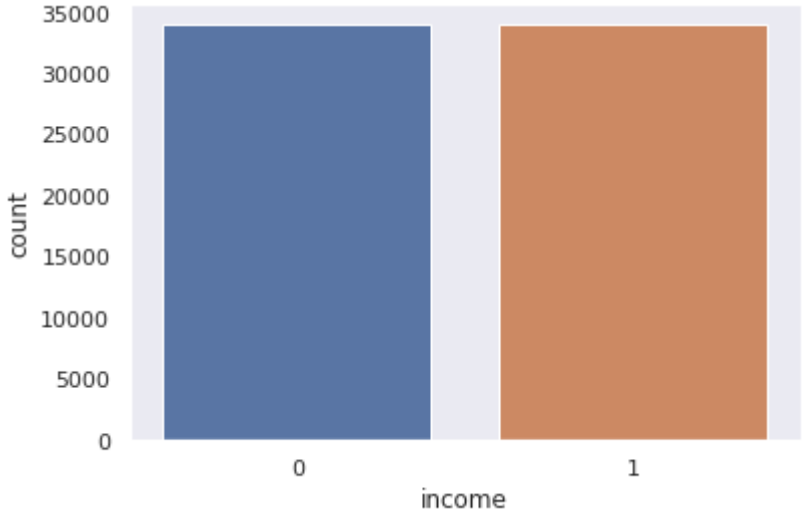
```
Distribution data = [34014 34014]
```



```
Begin training AdaBoostClassifier(base_estimator=DecisionTreeClassifie
r(max_depth=2,
                                                    min_samples_s
plit=6),
                   learning_rate=0.5, n_estimators=150)...

Time Elapsed = 15.828125953674316 seconds



{'acc_test': 0.8907508672899395, 'f_test': 0.8982911985018728}
```

Undersample

```
Distribution data = [34014 34014]
```




```
Begin training AdaBoostClassifier(base_estimator=DecisionTreeClassifie
r(max_depth=2,
                                                    min_samples_s
plit=6),
                   learning_rate=0.5, n_estimators=150)...

Time Elapsed = 15.035061836242676 seconds



{'acc_test': 0.888340095254895, 'f_test': 0.89621758262335}
```

NOTAS

O melhor resultado foi obtido com o modelo de AdaBoost sendo que foi otimizado com gridSearch e feito mais um tunagem com oversample.

Resultados:

```
{'acc_test': 0.8907508672899395, 'f_test': 0.8982911985018728}
```

Melhores resultados

Pesquisei qual o melhor resultado encontrado neste dataset. No kaggle tem competições utilizando este dataset onde o líder da competição conseguiu 0.87584

- https://www.kaggle.com/c/census-income/leaderboard (https://www.kaggle.com/c/census-income/leaderboard)
- https://www.kaggle.com/c/adult-census-income/leaderboard (https://www.kaggle.com/c/adult-census-income/leaderboard)
- https://www.kaggle.com/c/cs189-sp16-hw5-census/leaderboard (https://www.kaggle.com/c/cs189-sp16-hw5-census/leaderboard)
- https://www.kaggle.com/c/test-competition-ag/leaderboard (https://www.kaggle.com/c/test-competition-ag/leaderboard)

Storage Data Cleaning

```
'saved data at data/cleansing/df.csv'
```

References

- [1] https://www.researchgate.net/figure/phases-and-generic-tasks-of-CRISP-DM_fig1_283430974 (https://www.researchgate.net/figure/phases-and-generic-tasks-of-CRISP-DM_fig1_283430974)