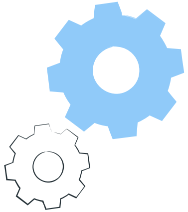
Annual income Classification

whether a person makes over 50K a year

Hassan Teymoori : 1947458



Big data computing 2020/21

Table of Contents

01

The Classification Task

What is the task?

The Classification Task

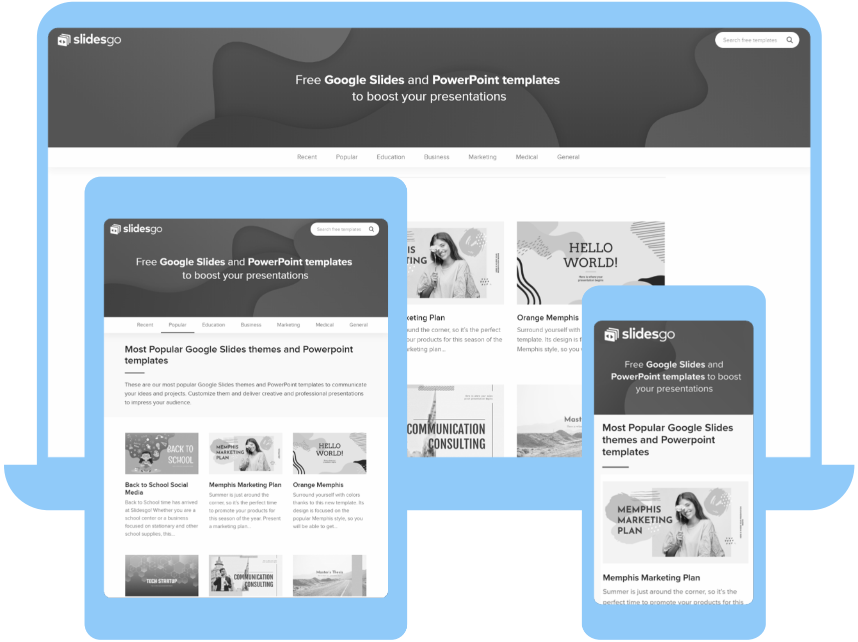
The classification task is, given a new person information, to predict an individual s' earning is **more or less** than 50,000 $ USD.

50,000 $?

02

Possible Applications

What would be the possible app?



Possible application

The recent coronavirus outbreak has seen a tremendous amount of people who signed up for the stimulus checks of $1200 in America after losing their jobs.

Possible application

- The recent coronavirus outbreak has seen a tremendous amount of people who signed up for the stimulus checks of $1200 in America after losing their jobs.
- **Requirement**: their annual incomes < $75,000

Possible application

The recent coronavirus outbreak has seen a tremendous amount of people who signed up for the stimulus checks of $1200 in America after losing their jobs.

Requirement: their annual incomes < $75,000

Problem: a big amount of people who have not done their taxes.

Possible application

The recent coronavirus outbreak has seen a tremendous amount of people who signed up for the stimulus checks of $1200 in America after losing their jobs.

Requirement: their annual incomes < $75,000

Problem: a big amount of people who have not done their taxes.

Understanding the potential annual income of unfiled taxes individuals for government to make strategic steps in taking care of them.

Possible application

The recent coronavirus outbreak has seen a tremendous amount of people who signed up for the stimulus checks of $1200 in America after losing their jobs.

Requirement: their annual incomes < $75,000

Problem: a big amount of people who have not done their taxes.

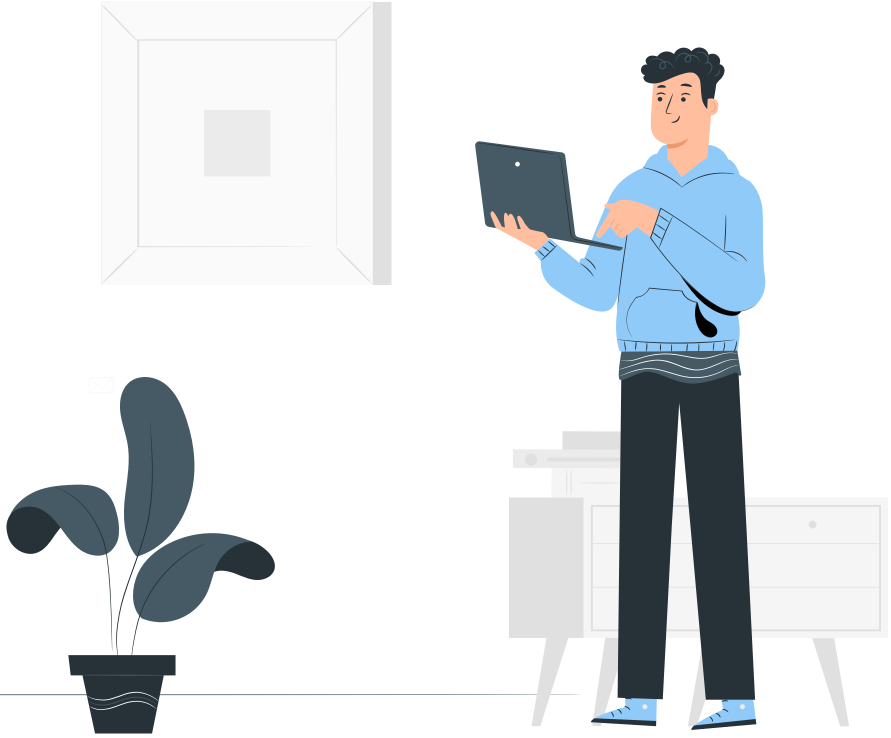
Understanding the potential annual income of unfiled taxes individuals for government to make strategic steps in taking care of them.

The **government** would benefit from a ML model that can help predict an individual's income base on their demographic features.

03

The Dataset

Structure of the dataset



The Dataset

The version in the Kaggle website contains of **32K** rows

The Dataset

The version in the Kaggle website contains of **32K** rows

The version in the UCI website contains of **Almost 50K** rows.

The Dataset

The version in the Kaggle website contains of **32K** rows

The version in the UCI website contains of **Almost 50K** rows.

I assumed the version in the Kaggle is only the **training dataset** and the one in the UCI is the combination of the training set and test set.

The Dataset

The version in the Kaggle website contains of **32K** rows

The version in the UCI website contains of **Almost 50K** rows.

I assumed the version in the Kaggle is only the **training dataset** and the one in the UCI is the combination of the training set and test set.

I have used the second one.

The Dataset

Contains **48,842** records.

The dataset has **15** columns (one is the target variable).

There are **8 categorical** columns

Target column is also **categorical**

There are **6 numerical** columns

The Dataset

```
|-- age: integer (nullable = true)
|-- workclass: string (nullable = true)
|-- fnlwgt: integer (nullable = true)
|-- education: string (nullable = true)
|-- education_num: integer (nullable = true)
|-- marital_status: string (nullable = true)
|-- occupation: string (nullable = true)
|-- relationship: string (nullable = true)
|-- race: string (nullable = true)
|-- sex: string (nullable = true)
|-- capital_gain: integer (nullable = true)
|-- capital_loss: integer (nullable = true)
|-- hours_per_week: integer (nullable = true)
|-- native_country: string (nullable = true)
|-- income: string (nullable = true)
```

Contains **48,842** records.

The dataset has **15** columns (one is the target variable).

There are **8 categorical** columns

Target column is also **categorical**

There are **6 numerical** columns

The Dataset

Categorical Features

workclass	education	marital_status	occupation	relationship	race	sex	native_country
State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	United-States
Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	United-States
Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	United-States
Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	United-States
Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	Cuba

The Dataset

Numerical Features

age	fnlwgt	education_num	capital_gain	capital_loss	hours_per_week
39	77516	13	2174	0	40
50	83311	13	0	0	13
38	215646	9	0	0	40
53	234721	7	0	0	40
28	338409	13	0	0	40

Target Variable

income
<=50K
<=50K
> 50K
<=50K
> 50K

04

Data Exploration

Observations and challenges

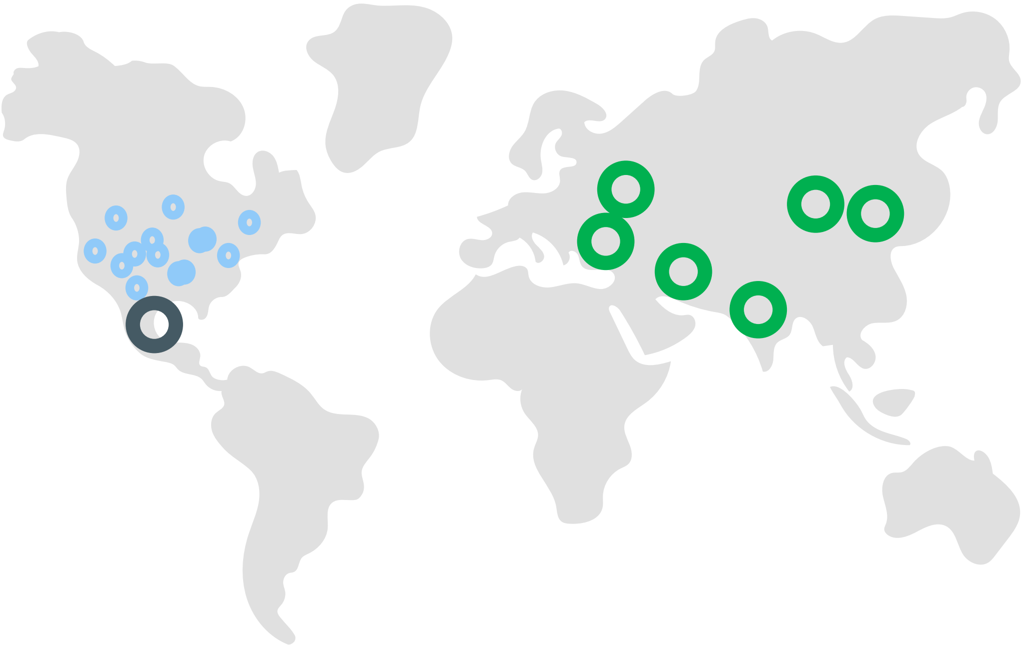
Countries

United State

Most of the data
are take from US

Mexico

Rest of world

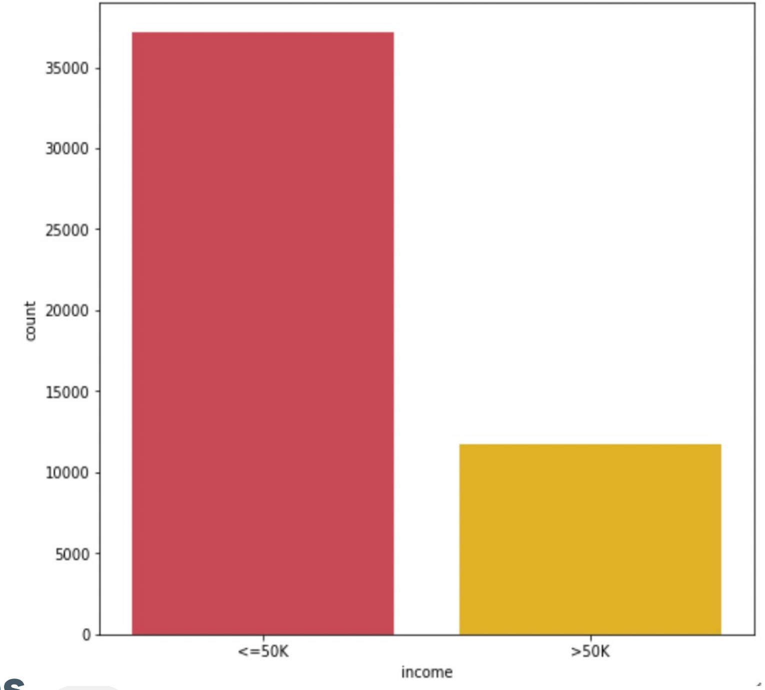


Data Exploration

Unbalanced Dataset

my dataset is not highly unbalanced but making it to be balanced is a good way to make sure the outcome is reliable.

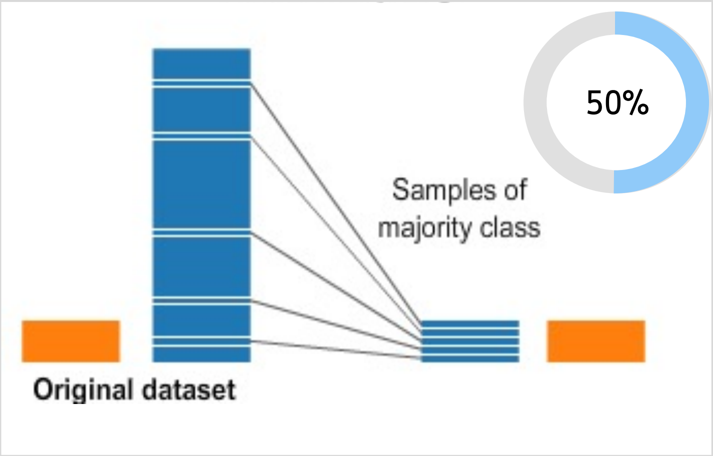
with and without balancing approaches and compared the result



Data Exploration

Unbalanced Dataset

The Solutions?

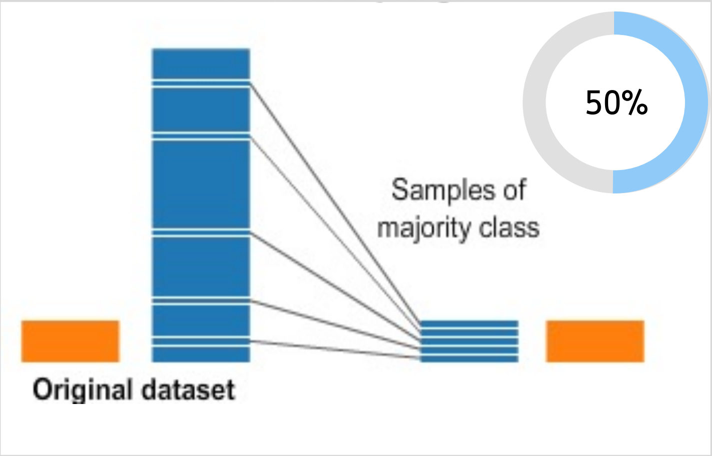


[Source of picture]

Data Exploration

Unbalanced Dataset

The Solutions?



Or

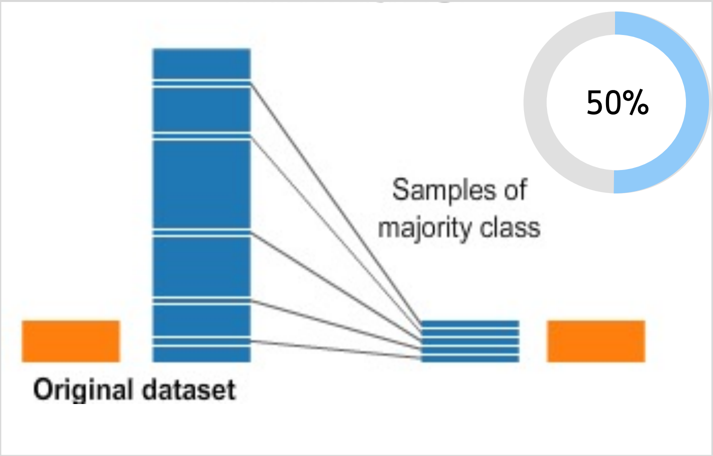
Over-sampling

Source of picture

Data Exploration

Unbalanced Dataset

The Solutions?



Or

Over-sampling

Or

Give weight to samples

Source of picture

Data Exploration

Missing Values

Missing Values encoded as a question mark in 3 categorical columns

	Missing Values	Ratio %
workclass	2809	5.8
occupation	2809	5.8
native_country	857	1.8

Data Exploration

Missing Values

	Missing Values	Ratio %
workclass	2809	5.8
occupation	2809	5.8
native_country	857	1.8

```
`age` = 0
`workclass` = 2809
`fnlwgt` = 0
`education` = 0
`education_num` = 0
`marital_status` = 0
`occupation` = 2809
`relationship` = 0
`race` = 0
`sex` = 0
`capital_gain` = 0
`capital_loss` = 0
`hours_per_week` = 0
`native_country` = 857
`income` = 0
```

Data Exploration

Missing Values

The Solutions?

Remove the missing values

Replace Them

Data Exploration: Summary

Missing Data

balanced dataset.

Categorical features needed to be encoded;

Different scales of feature values;

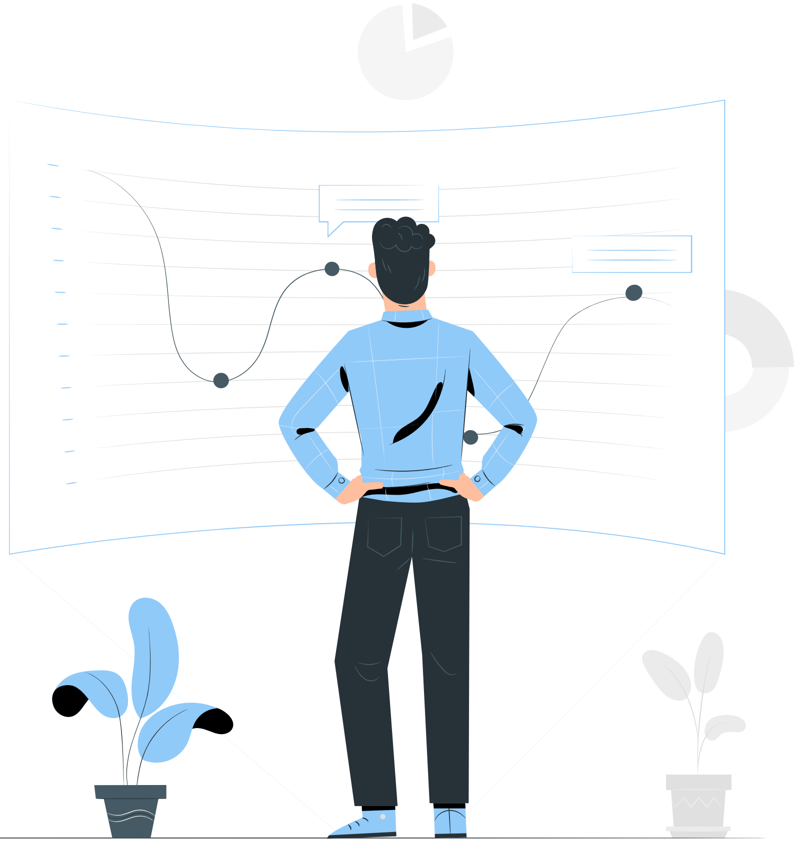
Several outliers on the age variable;

Clear Skewed problem on the age, fnlwgt variable;

05

Learning Pipeline

Methods and results



Learning Pipeline

The first step is to tackle with the data exploration observations.

Another important note is that:

- I do not have testing set.

Learning Pipeline

Balancing the Dataset.

- Down-sampling (under sampling)
- Over-sampling
- Give weight to samples

Providing The test set

- Split The dataset into two portions
  - Training set 80%
  - Test set 20%

Learning Pipeline

Categorical Features

Transform to numerical features:

- String indexer
- One-Hot-Encoder
- Vector Assembler

Learning Pipeline

Method that I used

- Logistic Regression
- Decision Tree
- Random Forest
- SVM
- Gradient Boosted Decision Tree

Learning Pipeline: Methods

	Single Model training	HP-tuning and Cross validation	with standard scaling	Without Down Sampling
Logistic Regression	✓	✓	✓	-
Decision Tree	X	✓	no need	✓
Random Forest	X	✓	no need	-
SVM	X	✓	✓	✓
G-B Decision Tree	X	✓	no need	-

Learning Pipeline: Logistic Regression

```
____________________*****   Best Model: Logistic Regression   *****____________________

Best model according to k-fold cross validation has:

                lambda       : 0.0
                maxIter      : 50
                fitIntercept : True
                alpha        : 0.0

Command took 0.03 seconds -- by teymoori.1947458@studenti.uniroma1.it at 06/07/2021, 11:01:48 on final-bdc-
```

Learning Pipeline: Logistic Regression

	Single Model Without HP-tuning (Not Scaled)	Best Model Resulting Cross Validation and HP-tuning (Not Scaled)
Area under ROC	0.900	0.901
Area under PR	0.893	0.911
Accuracy	81.4 %	82.3 %
Precision	81.6 %	82.4 %
Recall	81.4 %	82.3 %
F1-Score	0.814	0.822

Learning Pipeline: Logistic Regression

	Single Model Without HP-tuning (Not Scaled)	Best Model Resulting Cross Validation and HP-tuning (Not Scaled)	Same pipeline in Second Column with **Scaled data**
Area under ROC	0.900	0.901	0.900
Area under PR	0.893	0.911	0.897
Accuracy	81.4 %	82.3 %	80.6 %
Precision	81.6 %	82.4 %	80.8 %
Recall	81.4 %	82.3 %	80.6 %
F1-Score	0.814	0.822	0.806

Learning Pipeline: Logistic Regression

I expected by applying the scaling the result would be better.

Why Not?

Without Scaling	With Scaling
lambda : 0.0	lambda : 0.0
maxIter : 50	maxIter : 25
fitIntercept : True	fitIntercept : True
alpha : 0.0	alpha : 0.1
Accuracy: 82.3 %	Accuracy: 80.6 %

My bad: I should have tried other hyper-parameters: (MaxIter > 25)

Learning Pipeline: Logistic Regression

Check generalization (overfitting)
Best Model

Training Set	Test Set
areaUnderROC: 0.907	areaUnderROC: 0.901

My model is not overfitted and it would be able to generalize well to the new data as like my test set

Learning Pipeline: Decision Tree

```
__________________________*****   Best Model: Decision Tree   *****________________________

Best model according to k-fold cross validation has:

                 maxDepth        : 24
                 minInfoGain        : 0.0
                 impurity        : entropy
```

Learning Pipeline: Decision Tree

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.839
Area under PR	0.856
Accuracy	78.7 %
Precision	78.8 %
Recall	78.7 %
F1–Score	0.787

Learning Pipeline: Decision Tree

Check generalization (overfitting)
Best Model

Training Set	Test Set
areaUnderROC: 0.817	areaUnderROC: 0.839

My model is not overfitted and it would be able to generalize well to the new data as like my test set

Learning Pipeline: Decision Tree

Important Features

Workclass

Education

Marital Status

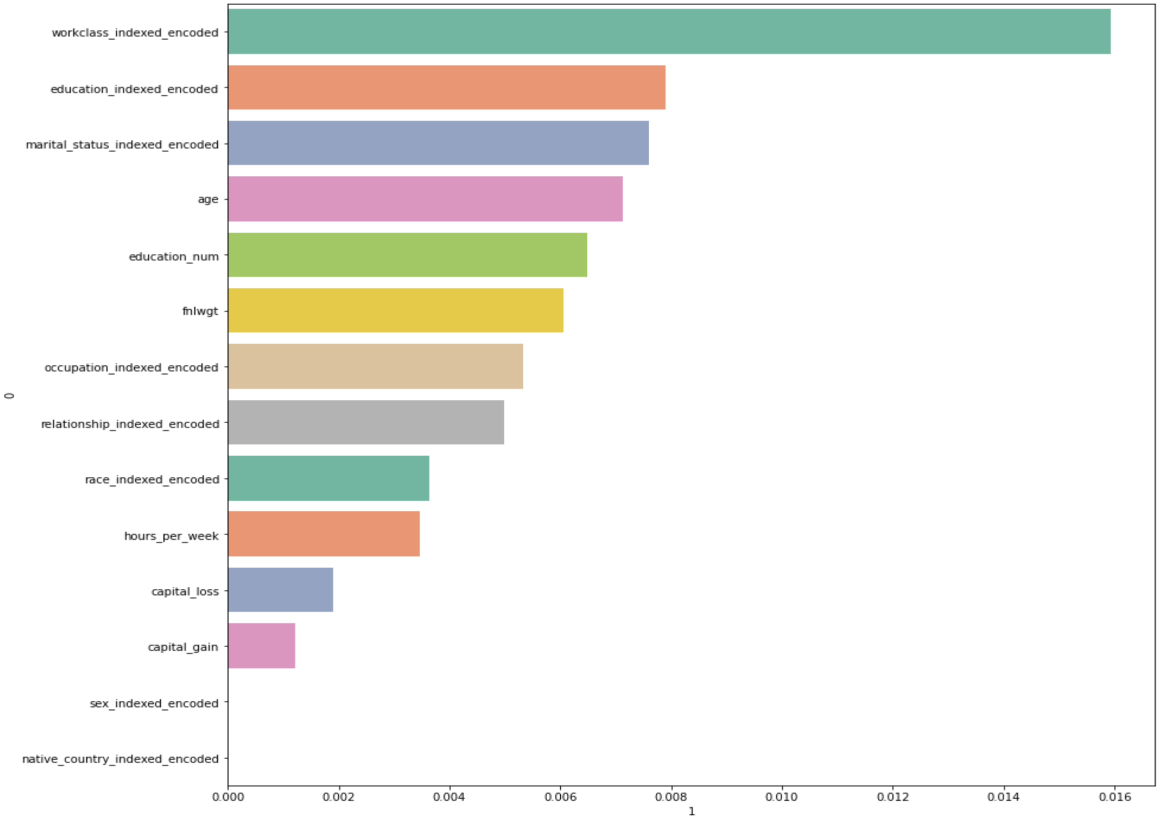
Age

Education_Num

Final weight

Occupation

Which all of these are expected to be important when talking about one's income



Learning Pipeline: SVM

```
_____________________________*****    Best Model: SVM    *****_____________________________

Best model according to k-fold cross validation has:

                 regParam          : 0.01
                 maxIter           : 50
```

Learning Pipeline: SVM

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.893
Area under PR	0.901
Accuracy	80.7 %
Precision	81.2 %
Recall	80.7 %
F1–Score	0.806

Learning Pipeline: SVM

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning	Same pipeline in Second Column with **Scaled data**
Area under ROC	0.893	0.894
Area under PR	0.901	0.889
Accuracy	80.7 %	80.4 %
Precision	81.2 %	80.9 %
Recall	80.7 %	80.4 %
F1–Score	0.806	0.803

Learning Pipeline: SVM

I expected by applying the scaling the result would be better.

Why Not?

Without Scaling

```
regParam          : 0.01
maxIter           : 50
```

Area under PR : 0.901

With Scaling

```
regParam_std      : 0.01
maxIter_std       : 100
```

Area under PR : 0.889

Learning Pipeline: SVM

Why Not?

This time I have tried even many more values for Hypermeters with Scaling data

Learning Pipeline: SVM

Why Not?

This time I have tried even many more values for Hypermeters with Scaling data

There's no reason to believe that the new scaling is any better.

It’s true that the rescaled features will all vary in comparable units.

However, it is also possible that the original scaling happened to encode the data such that some important features had more prominence in the model.

Learning Pipeline: SVM

Why Not?

This time I have tried even many more values for Hypermeters with Scaling data

There's no reason to believe that the new scaling is any better.

It's true that the rescaled features will all vary in comparable units.

However, it is also possible that the original scaling happened to encode the data such that some important features had more prominence in the model.

A feature in my dataset (fnlwgt) is happened to encode the data such that they become more important than others.

Learning Pipeline: SVM

Why Not?

This time I have tried even many more values for Hypermeters with Scaling data

There's no reason to believe that the new scaling is any better.

It’s true that the rescaled features will all vary in comparable units.

However, it is also possible that the original scaling happened to encode the data such that some important features had more prominence in the model.

A feature in my dataset (fnlwgt) is happened to encode the data such that they become more important than others.

The new scale results they all appear on similar scales and are all treated as equally important.

Learning Pipeline: SVM

Check generalization (overfitting)
Best Model

Training Set	Test Set
areaUnderROC: 0.896	areaUnderROC: 0.893

My model is not overfitted and it would be able to generalize well to the new data as like my test set

Learning Pipeline: Random Forest

I could not apply param_grid at one shot:
Failed all the time both on databricks and Google Colab

Create small Grids with different hyperparameters
And run every small Grid separately!

Even if fails, I won't lose everything. Some of the
small grids were able to finish their job.

Successful grids will be documented separately.

Learning Pipeline: Random Forest

Documented Result.
Complete summary of each small grid is in the notebook

```
_______________*****    Best Model: Random Forest

according to k-fold cross validation has:

      maxDepth          : 12
      impurity          : gini
      numTrees          : 100
```

```
_______________*****    Best Model: Random Forest

according to k-fold cross validation has:

      maxDepth          : 14
      impurity          : gini
      numTrees          : 100
```

```
_______________*****    Best Model: Random Forest

according to k-fold cross validation has:

      maxDepth          : 10
      impurity          : gini
      numTrees          : 120
```

```
_______________*****    Best Model: Random Forest

according to k-fold cross validation has:

      maxDepth          : 15
      impurity          : gini
      numTrees          : 120
```

Learning Pipeline: Random Forest

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.919
Area under PR	0.915
Accuracy	83.6 %
Precision	83.41 %
Recall	83.6 %
F1–Score	0.835

Learning Pipeline: Random Forest

Check generalization (overfitting)
Best Model

Training Set	Test Set
areaUnderROC: 0.912	areaUnderROC: 0.919

My model is not overfitted and it would be able to generalize well to the new data as like my test set

Learning Pipeline: Gradient Boosted Decision Tree

The same approach of small grids. Even with small grid
Failed many times after hours of running!

I could not train too many models to find the best one.
This is all I have for Gradient Boosted Decision Tree

```
________________*****   Best Model: Gradient Boosted DT   *****

according to k-fold cross validation has:

      maxDepth          : 5
      maxIter          : 50
```

Learning Pipeline: Gradient Boosted Decision Tree

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.915
Area under PR	0.921
Accuracy	82.7 %
Precision	82.9 %
Recall	82.7 %
F1–Score	0.826

Learning Pipeline: Gradient Boosted Decision Tree

Check generalization (overfitting)
Best Model

Training Set	Test Set
areaUnderROC: 0.903	areaUnderROC: 0.915

My model is not overfitted and it would be able to generalize well to the new data as like my test set

Note: Without down-sampling

Same Pipeline has been applied to
Decision Tree & SVM

The result and comparison between all models
are summarized in the next slide

Comparison: with/ without Down-sampling

	AUC	AU-PR	Accuracy	PR	Recall	F1 Score
DT: Down-sampled data	0.839	0.856	78.7%	78.8%	78.7%	0.787
DT: No change on dataset	0.810	0.618	83.9%	83.2%	83.9%	0.834
SVM: Down-sampled data	0.893	0.901	80.7%	81.2%	80.7%	0.806
SVM: No change on dataset	0.897	0.751	84.5%	83.8%	84.5%	0.836

Best Model?

What is the best model?

Decision Tree

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.839
Area under PR	0.856
Accuracy	78.7 %

SVM

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.893
Area under PR	0.901
Accuracy	80.7 %
Precision	81.2 %
Recall	80.7 %
F1–Score	0.806

Logistic Regression

	Single Model Without HP-tuning (Not Scaled)	Best Model Resulting Cross Validation and HP-tuning (Not Scaled)
Area under ROC	0.900	0.901
Area under PR	0.893	0.911
Accuracy	81.4 %	82.3 %
Precision	81.6 %	82.4 %
Recall	81.4 %	82.3 %
F1-Score	0.814	0.822

GBDT

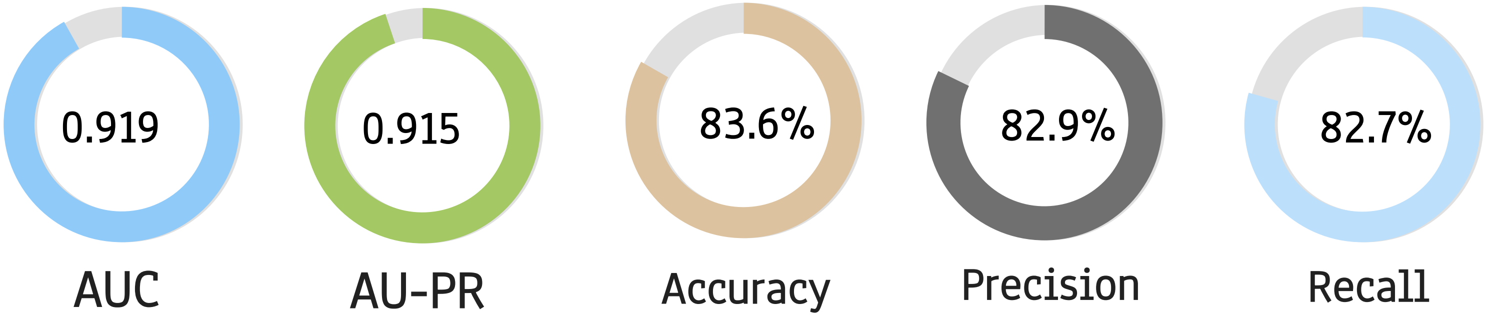
Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.915
Area under PR	0.921
Accuracy	82.7 %
Precision	82.9 %
Recall	82.7 %
F1–Score	0.826

Random Forest

Metrics used for evaluation	Best Model Resulting Cross Validation and HP-tuning
Area under ROC	0.919
Area under PR	0.915
Accuracy	83.6 %
Precision	83.41 %
Recall	83.6 %
F1–Score	0.835

Best Method: Random Forest

Best Model

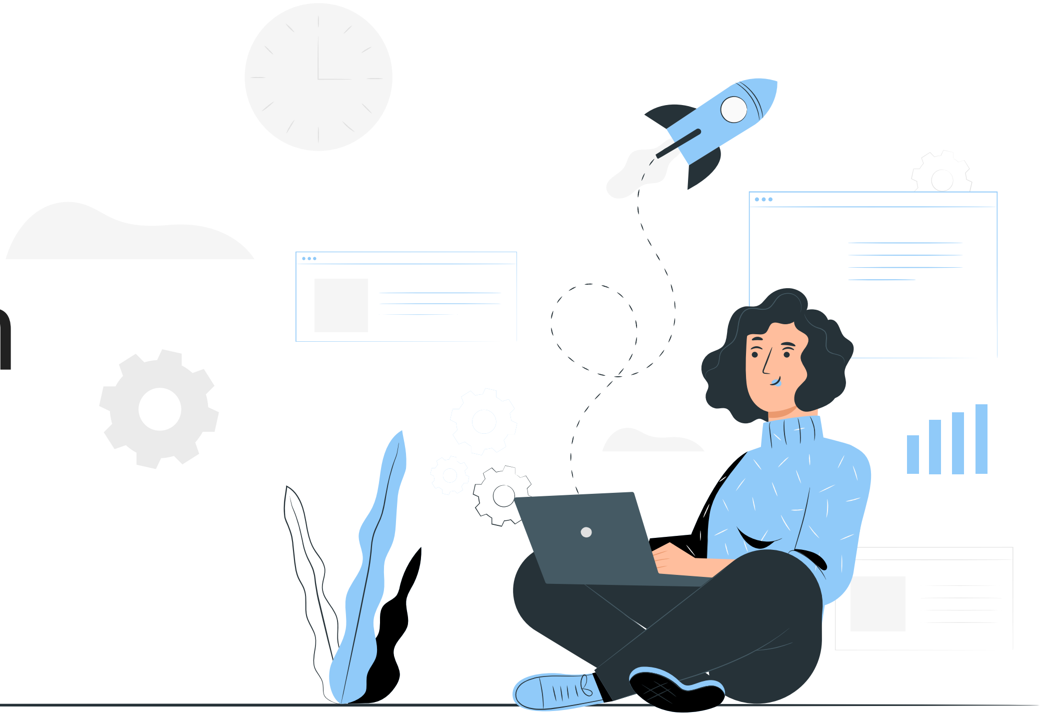


Max-Depth: 15
Impurity: gini
NumTrees: 120

06

Conclusion

Outcome and future work



Conclusion

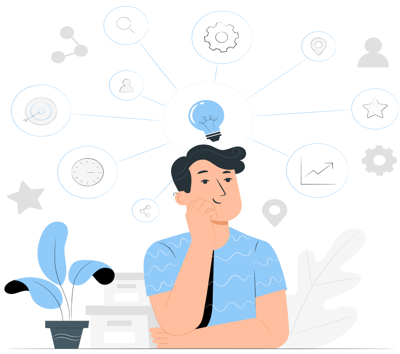
- Different methods have been compared
- The best model was using Random Forest with:
  - Max-depth 15
  - Trees 120
  - Impurity gini

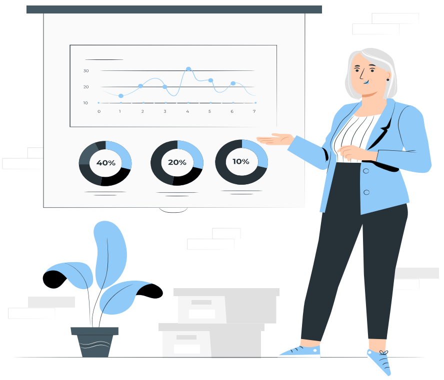
- The results of gbtr are also good, but training takes much more time

- There could be a way to improve the performance. (future work)

Future Work

- Handling Skewed data using different approaches like: **Log Transform**
- Apply other methods: like **Naïve bayes** and compare the result.
- Other strategy to balance the dataset like : **SMOTE**





Thanks!

teymoori.1947458@studenti.uniroma1.it

Big data computing Project
2020/21

Resources

The main resources are the course repositories, part of the codes are taken from the material notebooks.

- https://github.com/gtolomei/big-data-computing
- https://towardsdatascience.com/top-3-methods-for-handling-skewed-data-1334e0debf45
- https://github.com/gtolomei/big-data-computing/blob/master/notebooks/Classification.ipynb
- https://github.com/gtolomei/big-data-computing/blob/master/slides/10_Logistic_Regression.pdf
- https://medium.com/@junwan01/oversampling-and-undersampling-with-pyspark-5dbc25cdf253
- https://stackoverflow.com/questions/50363463/linearsvc-missing-in-apache-spark-2-1-non-linear-kernels-in-spark-2-2
- https://stackoverflow.com/a/38781980
- https://stackoverflow.com/a/63910523
- https://spark.apache.org/docs/3.0.0-preview/mllib-decision-tree.html
- And Many more.....