Uniwersytet Wrocławski

Online Shoppers Purchasing Intention

Machine Learning, 24/25

Team 7
Denys Tsebulia, 351322
Mafalda Costa, 351255
Mariana Carvalho, 351254

Table of contents

01

Problem statement

Problem statement

Online shoppers purchasing intention:

Predict whether an online shopper, based on a single session, is going to make a purchase or not.

Use case

In the context of a company, our problem provides **valuable insights into customer behaviour**:

- if it was predicted that a specific user will make a purchase from the company's website, it's **worth investing** in advertisements or offers for that specific user.
- on the contrary, the company would not be wasting resources on that user.

02

Data

Data

- The dataset consists of information gathered in a period of one year from **12,330 user sessions**, such that each session corresponds to the activity of a unique user.
- Dataset of **17 features** and **one target**, the **Revenue**, which indicates if a person made a purchase or not.
- Of the 12,330 sessions in the dataset, **84.5%** (10,422) are **negative class samples**, so users that did not make a purchase, and the rest **15.5%** (1908) are **positive class samples**, so users that did make a purchase.
- **8** features (including target) in the dataset are **categorical**.

https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset

03

Data analysis and pre-processing

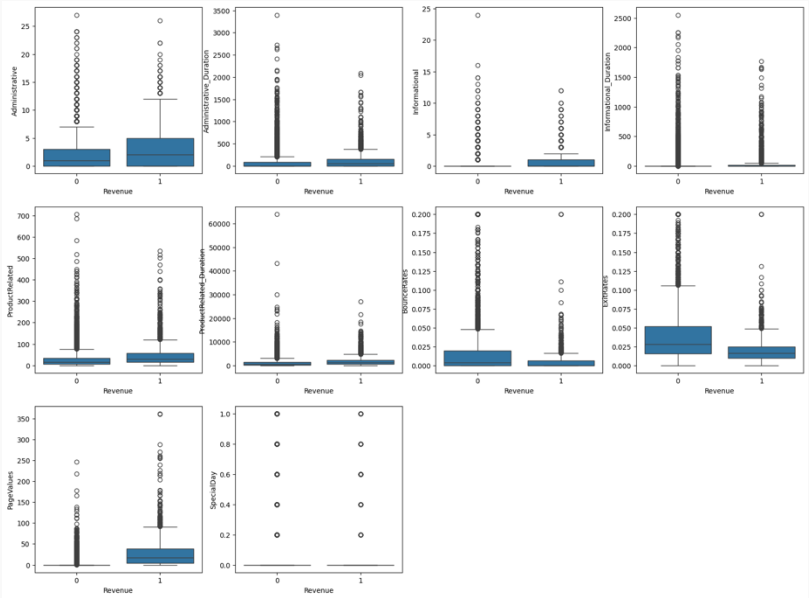
Data analysis

We began by analyzing our data:

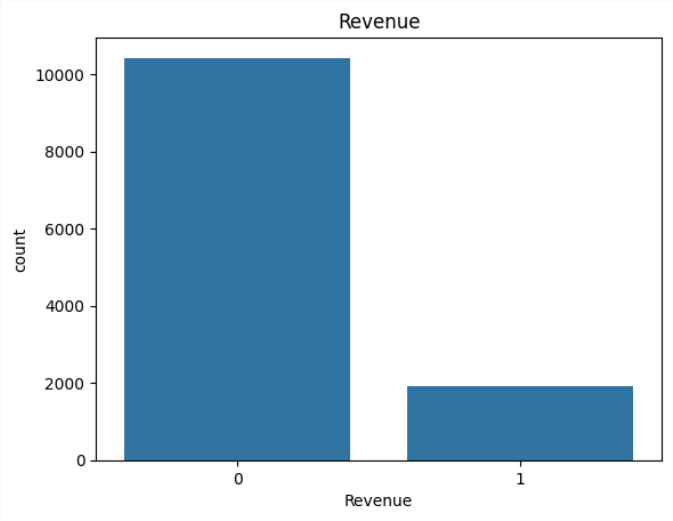
- no **NULL values**;
- **125 duplicated rows**;
- **correlation between the numerical features**;
- **outliers**;
- identified 8 **categorical features**;
- the **target**(“Revenue”) had a very **imbalanced class distribution**;

Data analysis

Outliers of the numerical features.



Class distribution of the target.



Data pre-processing

To handle the information gathered from the analysis and to prepare the data for the models:

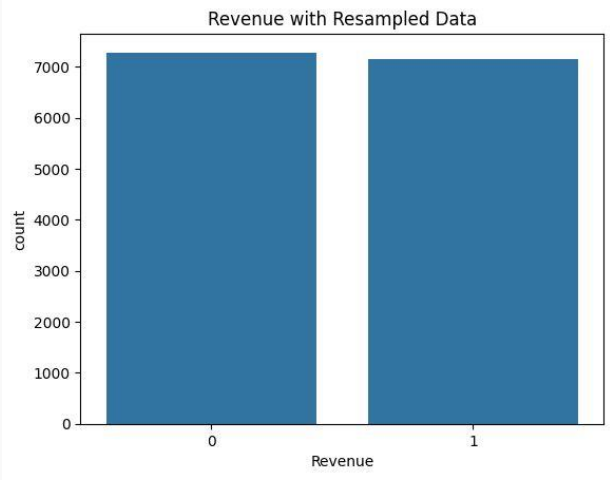
- **removed the duplicate rows**;
- used the **InterquartilleRange (IQR)** to remove the outliers that are outside the 2^{nd} and 98^{th} percentiles.
- used **OneHotEnconder** to transform the categorical features into numerical representations, as a result, the categorical features were expanded into 63 attributes;
- used **MinMaxScaler** to normalize the range of the features' values;

Data pre-processing

To handle the information gathered from the analysis and to prepare the data for the models:

- to **balance the class distribution** used **SMOTEEN** which applies **SMOTE** to the minority class and then uses **EditedNearestNeighbours** to "clean" the majority class.

- did **Feature Selection** with **SelectKBest** and **Mutual Information** as the scoring function. This yields the best combination of features, and it is calculated for each individual model.



04

Classification models

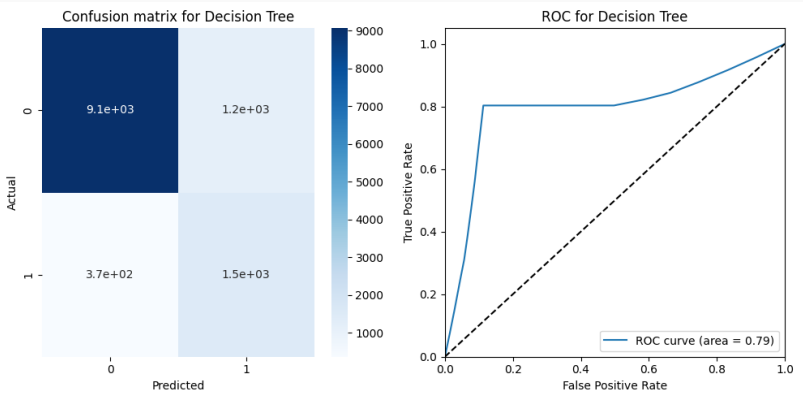
Models' setup

1. Performed **hyperparameter tuning** for every model:
   - **RandomizedSearchCV** with no input parameters to find the **best range of values** for the parameters used by the classifier.
   - **GridSearchCV** to obtain the **best parameters** for the classifier. The input for the search is the range obtained from RandomizedSearchCV.
2. Did **feature selection** as previously mentioned.
3. Used **K-Fold Cross-Validation**, with k=10:
   - By performing the data split many times, we ensure that the **model's performance is evaluated in a robust way**, increasing the generalizability and reducing the chance of overfitting.

Decision Tree

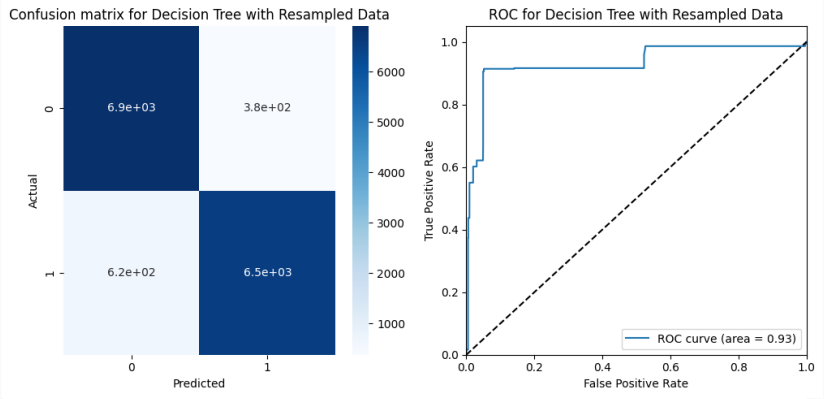
Normal data

2 features were selected by SelectKBest.



Resampled data

11 features were selected by SelectKBest.

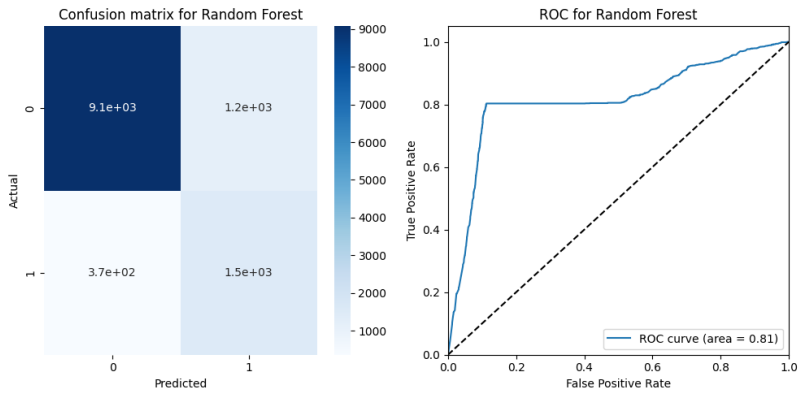


	Model	Accuracy_score	Recall_score	Precision_score	F1_score
0	Decision Tree	0.874049	0.803103	0.565136	0.663425
1	Decision Tree with Resampled Data	0.931362	0.913918	0.945770	0.929571

Random Forest

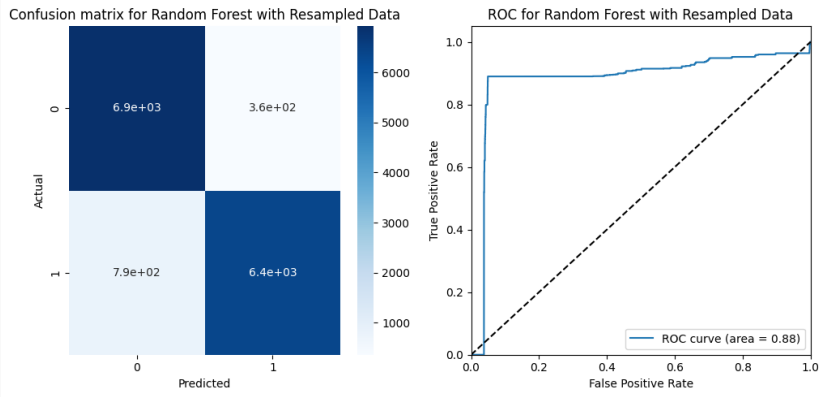
Normal data

54 features were selected by SelectKBest.



Resampled data

3 features were selected by SelectKBest.

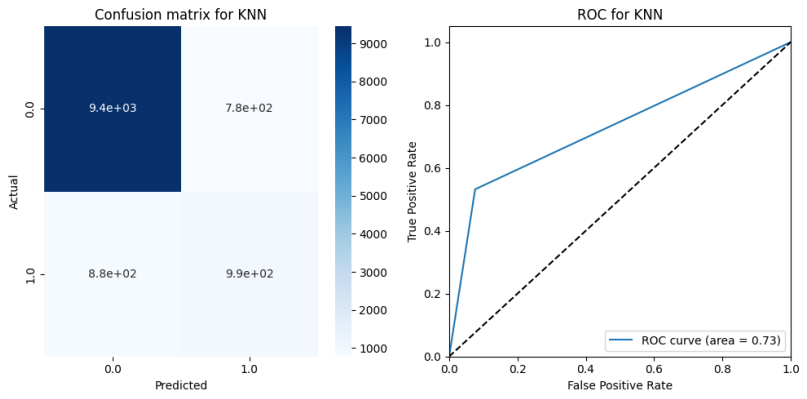


	Model	Accuracy_score	Recall_score	Precision_score	F1_score
0	Random Forest	0.874297	0.803103	0.565775	0.663866
1	Random Forest with Resampled Data	0.920418	0.889603	0.946617	0.917225

K-Nearest Neighbours

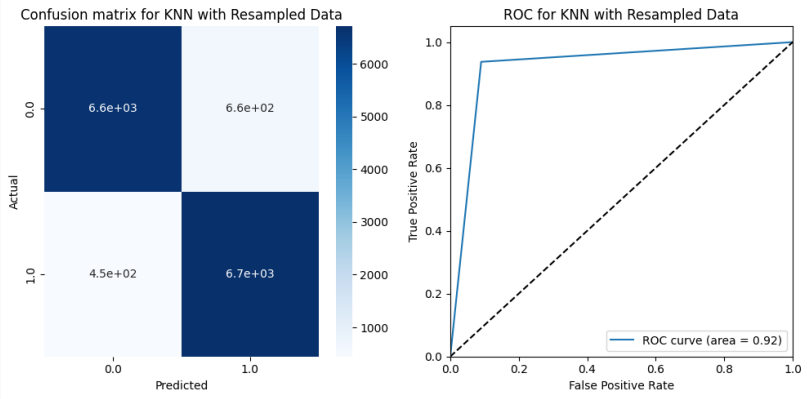
Normal scaled data

5 features were selected by SelectKBest.



Resampled scaled data

12 features were selected by SelectKBest.

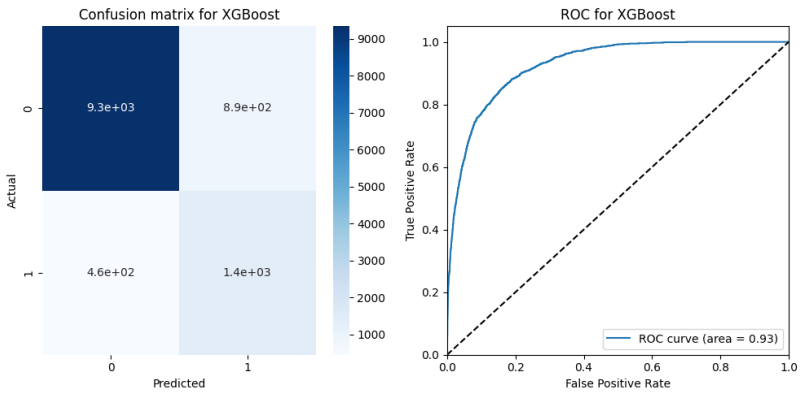


	Model	Accuracy_score	Recall_score	Precision_score	F1_score
0	KNN	0.863381	0.531835	0.561265	0.546154
1	KNN with Resampled Data	0.923397	0.937255	0.910782	0.923829

XGBoost

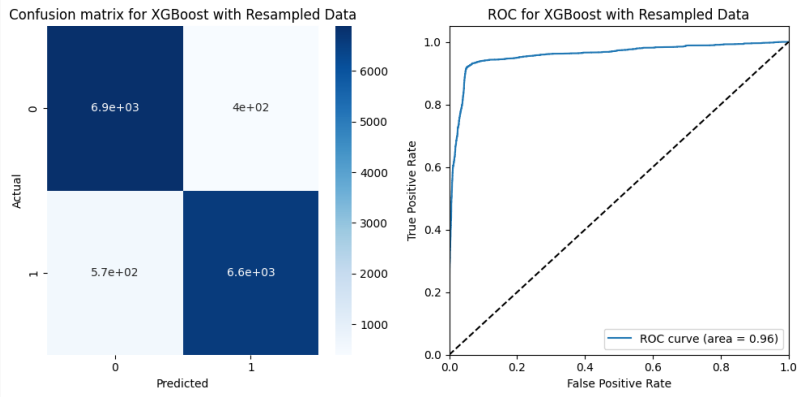
Normal data

48 features were selected by SelectKBest.



Resampled data

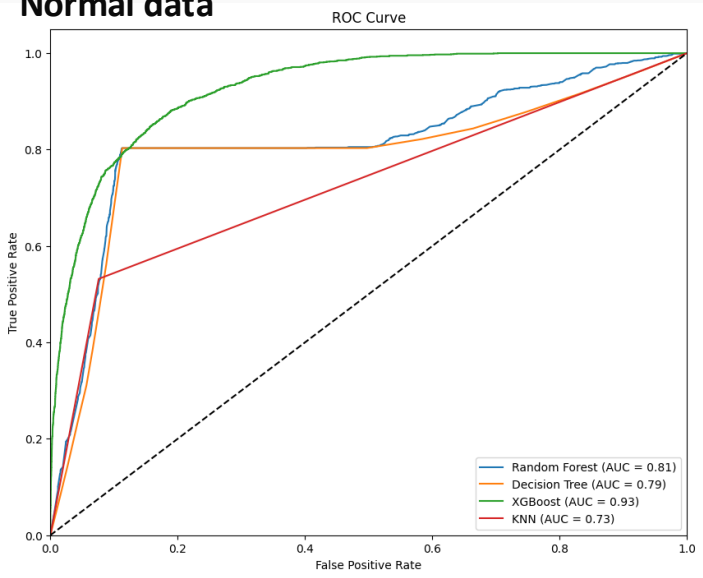
11 features were selected by SelectKBest.



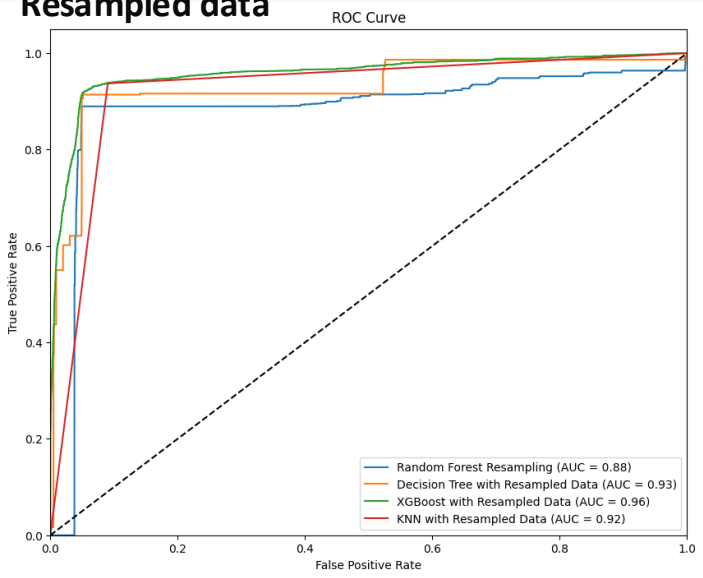
	Model	Accuracy_score	Recall_score	Precision_score	F1_score
0	XGBoost	0.889018	0.756019	0.614615	0.678023
1	XGBoost with Resampled Data	0.932955	0.920347	0.943013	0.931542

Comparing all models

Normal data



Resampled data



	Model	Accuracy_score	Recall_score	Precision_score	F1_score
0	Random Forest with Resampled Data	0.920418	0.889603	0.946617	0.917225
1	Decision Tree with Resampled Data	0.931362	0.913918	0.945770	0.929571
2	XGBoost with Resampled Data	0.932955	0.920347	0.943013	0.931542
3	KNN with Resampled Data	0.923397	0.937255	0.910782	0.923829

GitHub repository:

https://github.com/marianaosiecka/ML_project_uwr_2025.git

References

- https://www.kaggle.com/datasets/imakash3011/online-shoppers-purchasing-intention-dataset/data
- https://www.kaggle.com/code/sasakitetsuya/clustering-and-predict-modeling-by-pycaret
- https://www.kaggle.com/code/abhishekvaishnav/eda-and-prediction#Random-Forest-Classifier
- https://www.kaggle.com/code/saifuddinlokhand/analysis-dataset-with-93-accuracy
- https://link.springer.com/article/10.1007/s00521-018-3523-0

Thank you for your attention!

:)