Predicting Sales for Advertising Media

Objective

After reading this post and completing the accompanying exercises the reader should be able to build a simple machine learning model and use it to make predictions

Contents

1.Introduction

1.1 Business Problem

You are required to design the budget for the next advertising campaign for TV, radio, and newspapers. You are given the advertising budgets and the corresponding sales data for each of the channels for the last 200 campaigns. You decide to evaluate different budget proposals by using a machine learning model that can predict the sales volume for each of the channels.

1.2 Machine learning platform

Orange Data Mining is a free point-and-click, drag-and-drop, software for machine learning. Download the version for your operating system from <https://orangedatamining.com/download> and install it; the Windows file is over 400MB in size

1.3 Datasets

Historical data containing “sales” for 200 budgets: <https://www.statlearning.com/s/Advertising.csv>

New data: <https://github.com/sdfungayi/advertising-model/blob/main/Advertising_Proposals.xlsx>

1.4 Workflow

It is recommended to attempt to build the workflow by following the steps provided in this post or by referring to the workflow image before resorting to the Orange workflow file for the maximum benefit

Workflow file (double click this file to open the workflow in Orange): <https://github.com/sdfungayi/advertising-model/blob/main/Advertising%20-%20Workflow.ows>

Workflow image: <https://github.com/sdfungayi/advertising-model/blob/main/Workflow.JPG>

1.5 Approach

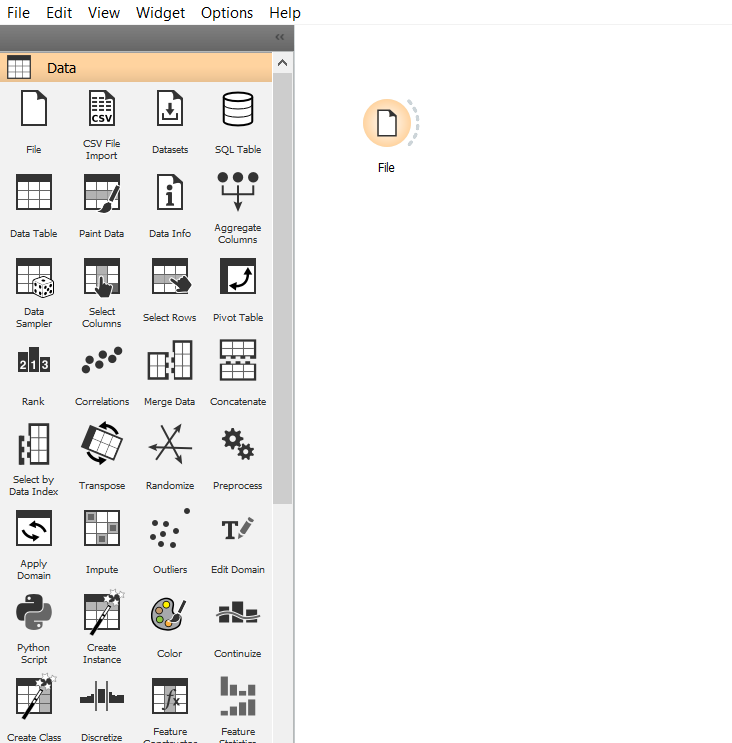
The dataset Avertising.csv will be used to train the machine learning model that will be used to make prediction on new data contained in the dataset “Advertising\_Predictions.xlxs”.

2. Import Data

Double-click the file Advertising.csv and open it in Excel. Study its structure – number of columns, number of rows, types of data, range of value, etc.

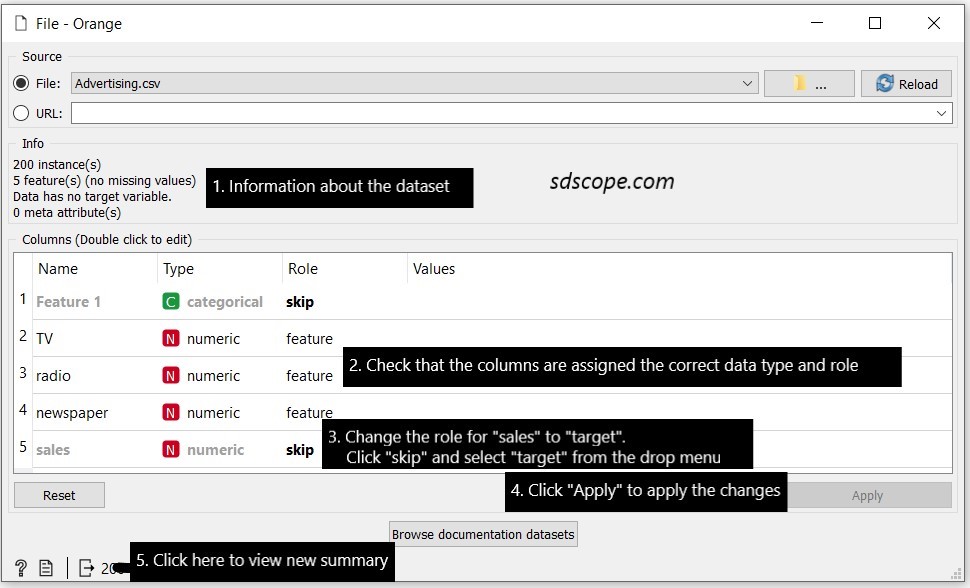
Start Orange software and add the widget **File** from the **Data** tab to the canvas by double-clicking it or by dragging it to the canvas, Figure 1 below

Figure 1. Import the dataset into Orange software



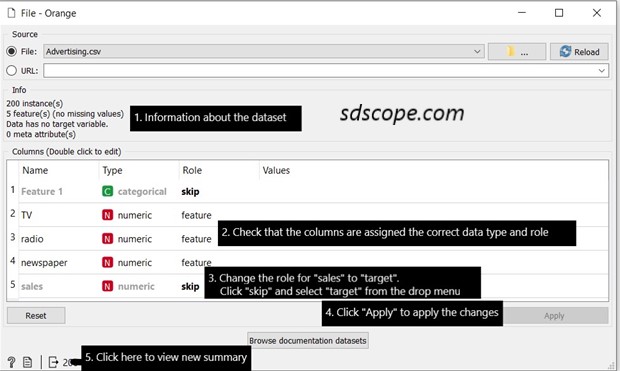
Double-click the widget File and navigate to the folder where the file Advertising.xlxs is locatedlocation of and import the dataset into the software as Save your work as often as possible to avoid loss

Figure 1. Import the dataset into Orange software



Double-click the widget **File** in the **Data** tab on the left side of the window and study the summary of the dataset, Figure 2 below. Does it match the Excel file?

Figure 2. Summary of the dataset



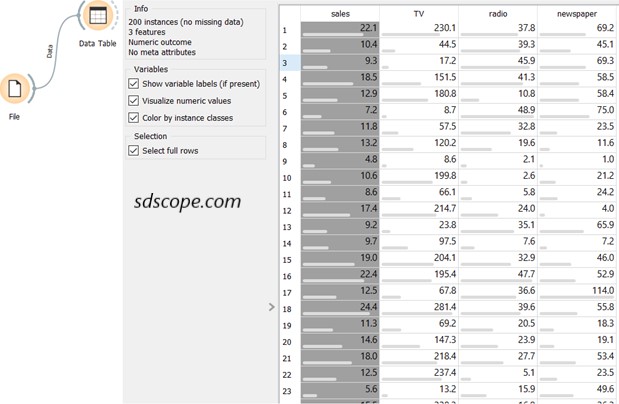
A dataset is a simple, flat table containing historical data of the scenario for which predictions are to be made. Each row, referred to as an example or instance, contains data for a single observation about a problem. Each column, referred to as an attribute, variable or dimension contains a feature of the observation. In supervised machine learning columns are divided into a set of descriptive features also called standard features, regular features or simply features, and a single target feature also called a label or an outcome. In the Advertising dataset the columns “TV”, “radio” and “newspaper” are the descriptive features and the column “sales” is the target or label; so this is a supervised machine learning problem.

This is a linear regression problem as we attempt to model the relationship between a set of numeric descriptive variables (“TV”, “radio” and “newspaper”) and a single numeric outcome.

There are several types of data: numeric, text, categorical, image data, network data, etc. Assigning the correct type to a feature is important because it determines how the data will be stored and how it wil used by machine learning algorithms. All data in the Advertising dataset is of type “numeric”, that is, they can be defined on a continuous scale. Data types are covered in detail in the post “Working with Data”.

Add the widget **Data Table** and connect it to the **File** widget, Figure 3. Open **Data Table** and view the features of the dataset that will be used in building the model: 3 features and a numeric outcome/target. Notice that the first column, the row ID, marked “skip” in Figure 2, will not be used for modeling; this is because it does not contain data about the problem.

Figure 3. View the features that will be used for modeling



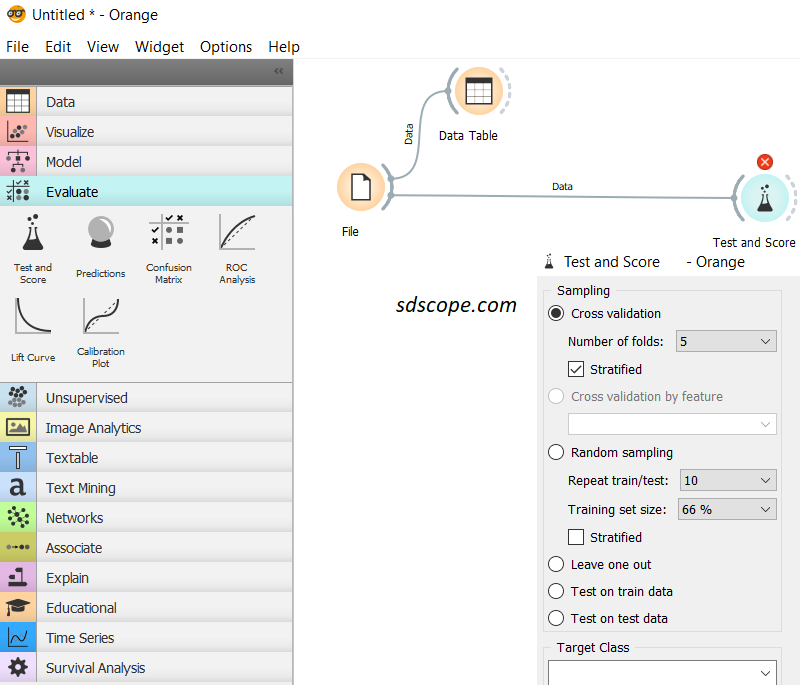
Modeling

The goal of machine learning is find the model that generalizes well, that is, makes correct predictions for queries on new, never-before seen data. The process involves using a subset of the data, called the “train set”, to create a model then testing the model with the remaining portion of the data, called the “test set”, on candidate machine learning algorithms and evaluating the results against some performance criteria with the objective of getting the best performance on the test set.

Contrast this with statistical inference where the goal is to find the model that best characterizes the relationship between the input variables and the outcome variable, for example, the line that minimizes mean absolute error across all in a linear regression model. The process involves using the whole dataset to create the model – no train set and not test set. While this model can be used to make predictions, it is quite unlikely to make good predictions on new never-before-seen data.

In Orange software open the **Evaluate** tab and add the **Test and Score** widget to the canvas. Connect the widget to the **File** widget, open it, set sampling to “Cross validation” and “Number of folds” to 5. See Figure 4 below. The **Test and Score** widget calculates the performance of each model built during the process.

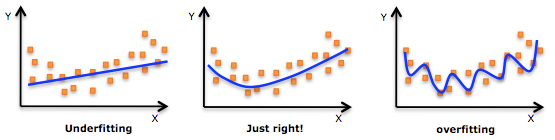
Figure 4. View the features that will be used for modeling



*k-fold cross validation* is one of the several model validation techniques used to handle the problem of “overfitting” in machine learning and is regarded as the gold standard for evaluating the performance of a model on unseen data. Overfitting occurs when a model learns a dataset too intricately to be useful on any other dataset since no two datasets are exactly identical in the real world. An overfitted model mistakes random variations in the dataset for persistent patterns, so would yield highly accurate predictions on training data but have a high error rate on unseen data. An ideal model strikes a balance between capturing significant trends and ignoring minor variations in data, making it useful for making predictions in production. See Figure 5 below.

Figure 5. Overfitting and Underfitting

*Source:* [*Quora*](https://www.quora.com/What-are-the-key-trade-offs-between-overfitting-and-underfitting)

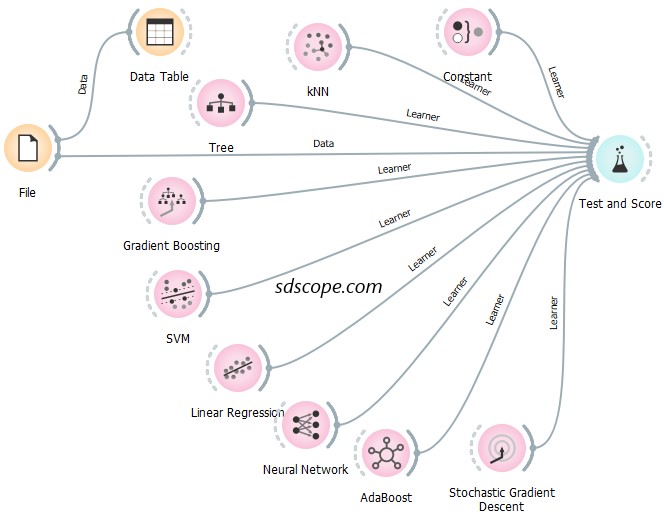


Methods for handling overfitting and underfitting as well as model validation techniques are discussed in detail in a separate post.

In Orange open the **Model** tab and add all the widgets to the canvas. These widgets represent the machine learning algorithms that will be tried in this exercise. It is impossible to know in advance which machine learning algorithm will perform the best for a given problem; the only way is to try as many algorithms as possible. This is known as the “No Free Lunch Theorem”.

Connect the widgets to the **Test and Score**. Delete the widgets that do not connect to **Test and Score** with a solid line (for example, **Stacking**, **Load Model** and **Save Model**) or connect with a solid line but trigger a red warning in **Test and Score** (for example, **Logistic Regression**, **Naïve Bayes** and **Random Forest**) because they do not apply for this problem. See Figure 5 below.

Figure 5. Creating and evaluating models



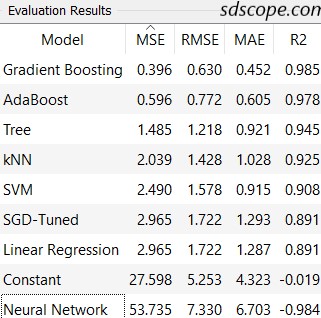
Model Selection

Model selection involves selecting the model with the best performance against an evaluation metric that was selected during the business problem understanding stage and is based on the business goal of the project. The key regression evaluation metrics are Mean Square Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Coefficient of Determination also known as R-Squared (R2).

The widget **Constant** calculates the performance baseline for the machine learning problem. Baseline performance is calculated from random predictions (guesswork). It provides a reference point from which to compare other machine learning algorithms. Algorithms that perform better the baseline are considered good and taken to the next stage of the process while those that fare worse are immediately dropped.

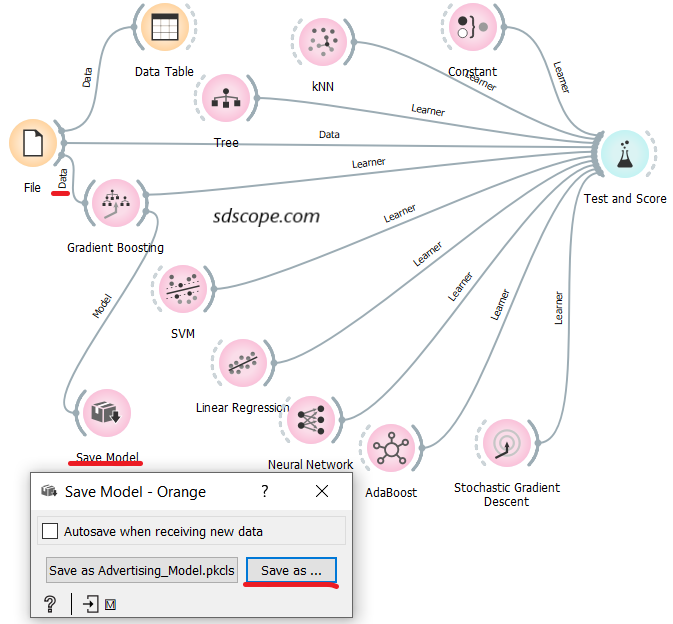
Open **Test and Score** and study the evaluation results, Figure 6. In this exercise it is assumed that MSE was the selected evaluation metric. Neural Network performs worse than **Constant** and should be dropped from further consideration (this is irrelevant in this exercise since no further experimentation will be done on algorithms). Click the heading “MSE” to sort the results so that the lowest MSE value goes to the top. **Gradient Boosting** has the lowest MSE and will be selected to make predictions on new data.

Figure 6. Model evaluation results



Add the widget **Save Model** from the **Model** tab and connect it to the output of Gradient Boosting; also connect the output of File to Gradient Boost as shown in Figure 7 below. Open **Save Model** and give the model a name. Notice that the model is simply a file generated by running an algorithm over a set of data to recognize certain types of patterns in the data.

Figure 7. Saving the model



Making Predictions

This step involves using the saved model to reason over new, never-before-seen data and make predictions about those data.

Open a new workflow in Orange software (in the menu click *File* then *New*) and give it a name

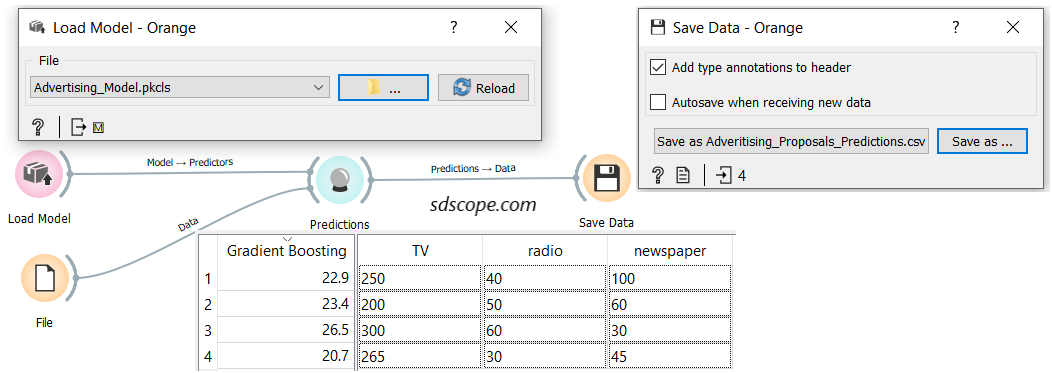
Add the widgets **File** and **Save Data** from the **Data** tab, the widget **Prediction** from the **Evaluate** tab and the widget **Load Model** from the **Model** tab to the canvas. Connect the widgets as shown in Figure 8 below.

Open **Load Model**, navigate to the folder where the model was saved and load the model

Open the **File** widget and import the file “Advertising\_Proposals” which contains new data (several proposed budgets) for which “sales” will be predicted

Open the **Predictions** widget; the predictions for each advertising budget proposal are contained in the column **Gradient Boosting**. The predictions can be accessed in Excel in the file specified in **Save Data** widget.

Figure 8. Making predictions



Note that new data must be within the range of values in the training set otherwise the predictions would likely be wild. It must also have data columns of the same type as the inputs selected to build the model. The model will ignore data columns of the same type as the inputs that were deselected when the model was built. It will also ignore data columns of the same type as the target column and will predict these columns.

Congratulations on building your first machine learning model and using it to solve a business problem.

Several steps were simplified or overlooked to allow the reader to build their first model in a gentle and fast way. Some of them are listed below:

* Framing the problem: this is the process of converting the business problem into a machine learning problem and involves understanding the background of the business problem, defining the business objectives, choosing the performance metric, how the model predictions will be integrated into the business environment, and establishing success and failure criteria, among several things
* Data exploration
* Feature engineering, for example, to transform the data into a form suitable for linear regression modeling
* Model selection: this is generally a trade-off amongst certain qualities including simplicity and transparency to humans, evaluation metrics, speed to train and test, and scalability
* Regularization to combat overfitting
* Tuning the selected model to improve its performance
* Ensemble methods: combining models to produce improved performance
* Model interpretability: easily understanding how the model works, for example, determining the impact of features on the model outcome; this is especially important in regulated industries such as healthcare and banking
* Prediction explanations: determining which features have the greatest impact on the predictions made by the model; helps users “comply with regulations, easily explain model outcomes to stakeholders, and identify high-impact factors to help focus their business strategies.” [2]
* Model deployment
* Model monitoring
* Bias, trust and ethics

Key Takeaways

It is possible to build and use a machine learning model without writing a single line of code

A dataset for a supervised machine learning problem consists of a set of descriptive features and a single label/outcome

The goal of machine learning is find the model that generalizes well on new data

A train set is used to create a model and a test set is used to validate the model

It is not possible to know in advance which machine learning algorithm will perform the best for a given problem; the only way is to try as many algorithms as possible (No Free Lunch Theorem)

Baseline performance provides a reference point from which to compare other machine learning algorithms

A machine learning model is simply a file generated by running an algorithm over a set of data to recognize certain types of patterns in the data

Prediction data must be similar to the training set in terms of both structure and range

Next Steps

Read and understand what linear regression is, how it works and its assumptions

Explore the Advertising dataset using the widgets in the **Visualize** tab (connect each of the widgets to the output of the **File** tab)

Find open source datasets on which to practise data visualization and building regression models

Reference

1. https://towardsdatascience.com/the-actual-difference-between-statistics-and-machine-learning-64b49f07ea3

2. https://www.datarobot.com/wiki/prediction-explanations

3.