Classification and K-Nearest Neighbours

I will be using a dataset at the case of customers’ default payments in Taiwan and compares the predictive accuracy of probability of default. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. I will briefly cover Classification and Regression using Nearest Neighbours.

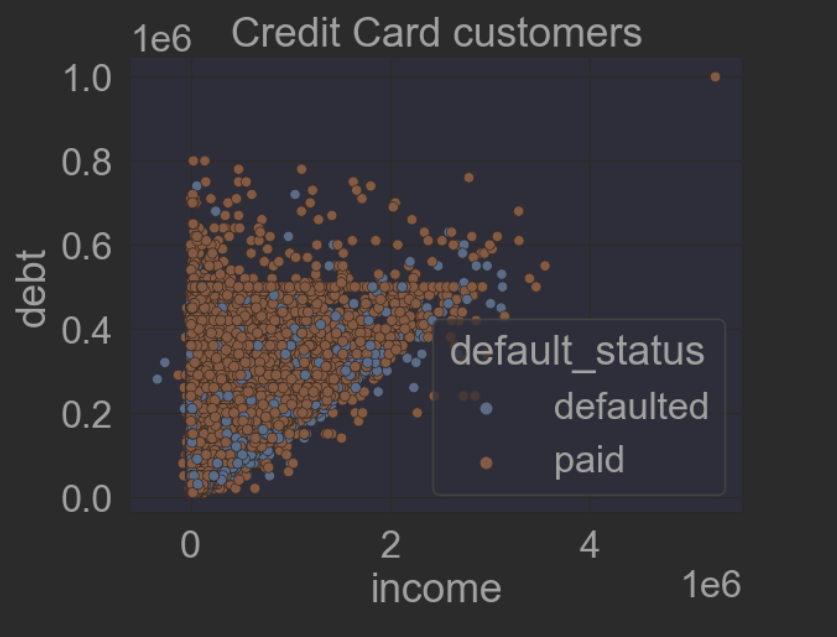
The nearest neighbors classifier simply asks: Which data point in the training set is closest to the data point about which you are trying to make a prediction?

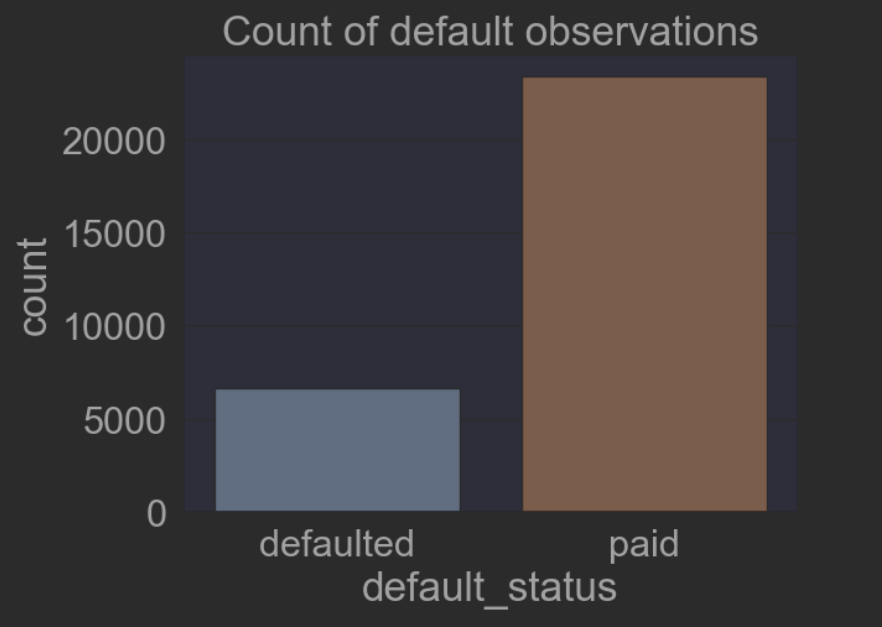
Classification is distinct from regression. In a regression problem the goal is to predict a real-valued outcome, given a set of features. In a classification problem, the goal is to predict the class to which a sample belongs, given a set of features.

Dataset URL: <https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients>

Data Manipulation: New feature income has been created as sum of bill statements.

X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.



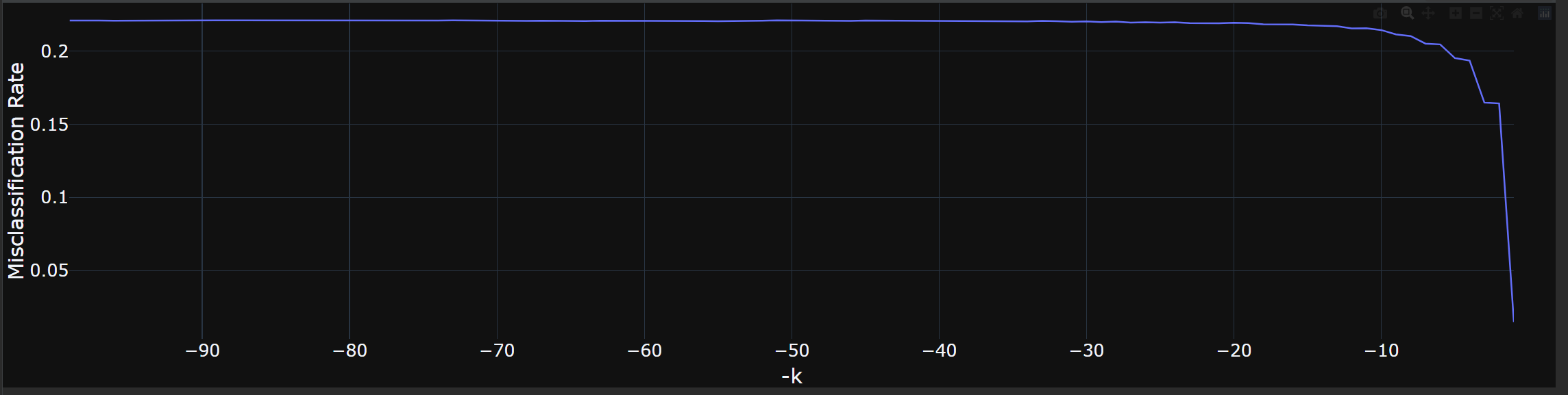


Classification

One approach to improving the nearest neighbor classifier is to consider multiple neighbors and doing a majority vote. There is a tradeoff in selecting k: For a very small k, the model is too complex and the variance is too high. When choosing k, k is a hyperparameter and thus the value you select for it can be consequential for your model.

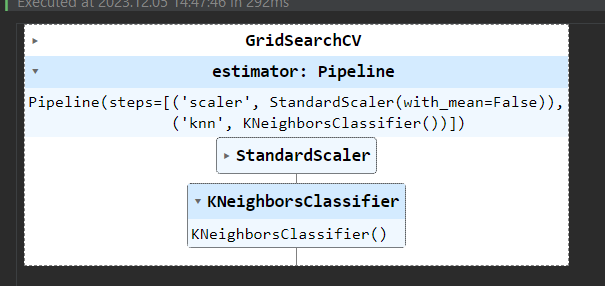
Since the predictions and observations are not numerical quantities, using mean squared error (MSE) is no longer an option.

One option is to use the misclassification rate. In other words, the fraction of predictions that are incorrect.

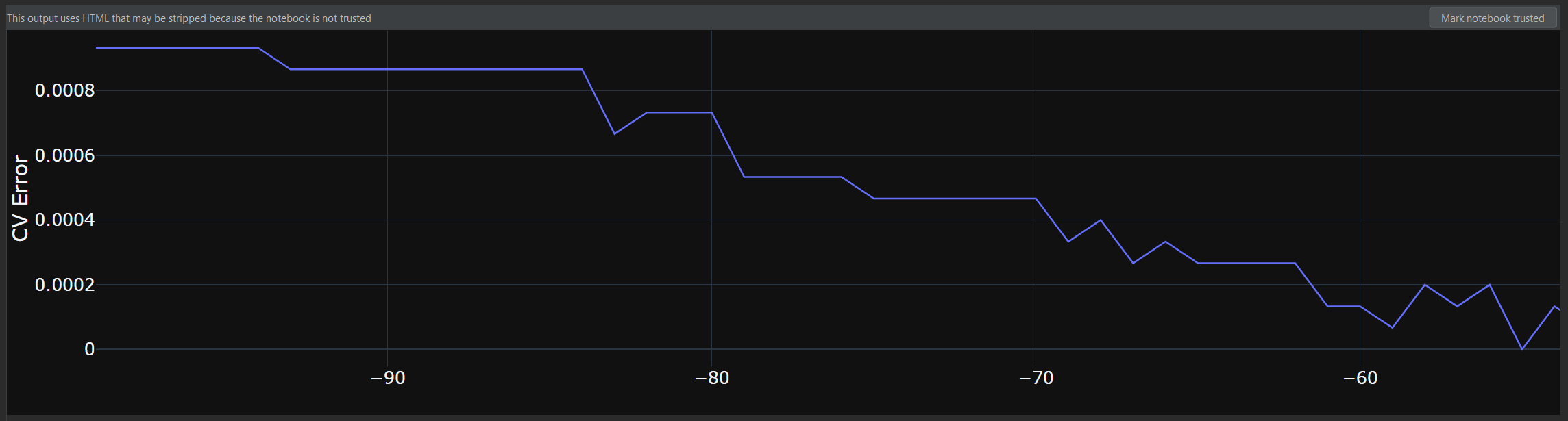


To best match a visual plot of the misclassification rate versus k with the error versus complexity diagram introduced earlier on, plot –k on the x-axis. Note that as the model complexity increases, the misclassification rate decreases. For low complexity models, the error is high, dropping as the model complexity increases.

One method for choosing the best k is to apply cross-validation. In scikit-learn, you can use GridSearchCV to identify the k with the lowest validation error.



After fitting the model, you ask the model\_finder object for its best estimator.



In this instance, it has trained a model with n\_neighbors = 49, with 49 being the optimal value.

For very large k’s the test fails, returning not a number (NaN). This is because it does not make sense to have a k greater than the number of available neighbors.

Before, accuracy was used to assess the quality of a classifier. But accuracy is not always the right metric. This is an example of a situation with imbalanced classes. A confusion matrix paints a more thorough picture of classifier usefulness, retaining usefulness even for wildly imbalanced classes.



For a binary classifier, with only two classes, the confusion matrix provides a count of:

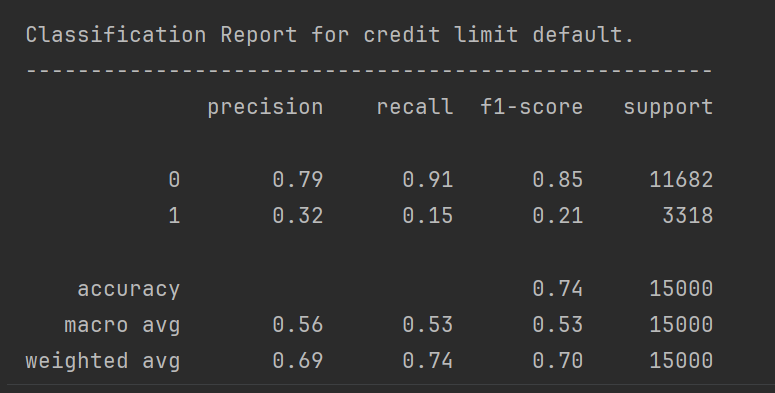
▪ True negative (TN)

▪ False positive (FP)

▪ False negative (FN)

▪ True positive (TP)

True negatives are situations where the model correctly predicts that the customer paid. In this example, there are 1689 of those. False positives are 4947 instances where this model predicted the customer did not paid, but they did pay. False negatives are where the model predicted that the customer has paid, but they actually defaulted, there are 11743 such cases in this example. True positives are where the model correctly predicted that a customer has defaulted, in this instance there are 11621 such cases.



The classification report is often used in machine learning to compute the accuracy of a classification model based on the values from the confusion matrix.

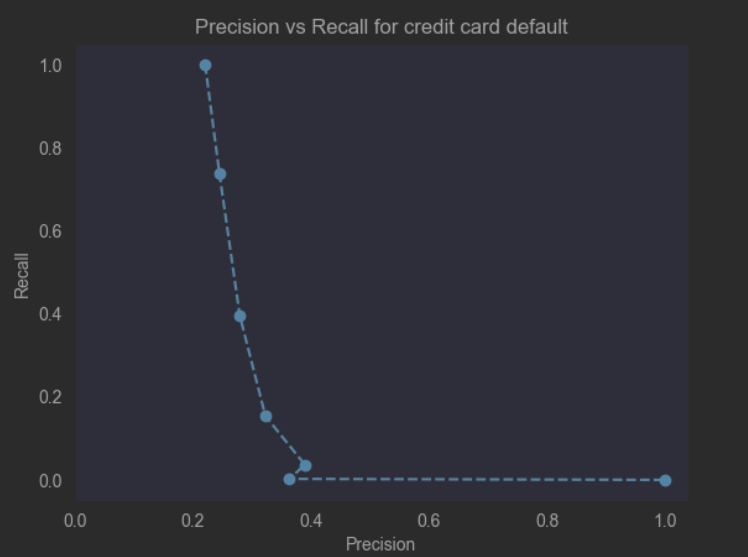
Since emphasis on a single metric can result in models that are not particularly useful, it is important to focus on the trade-offs between metrics instead. One of the most commonly explored trade-offs among these metrics, is between precision and recall.

Consider the meaning of these terms in the context of a credit card default model:

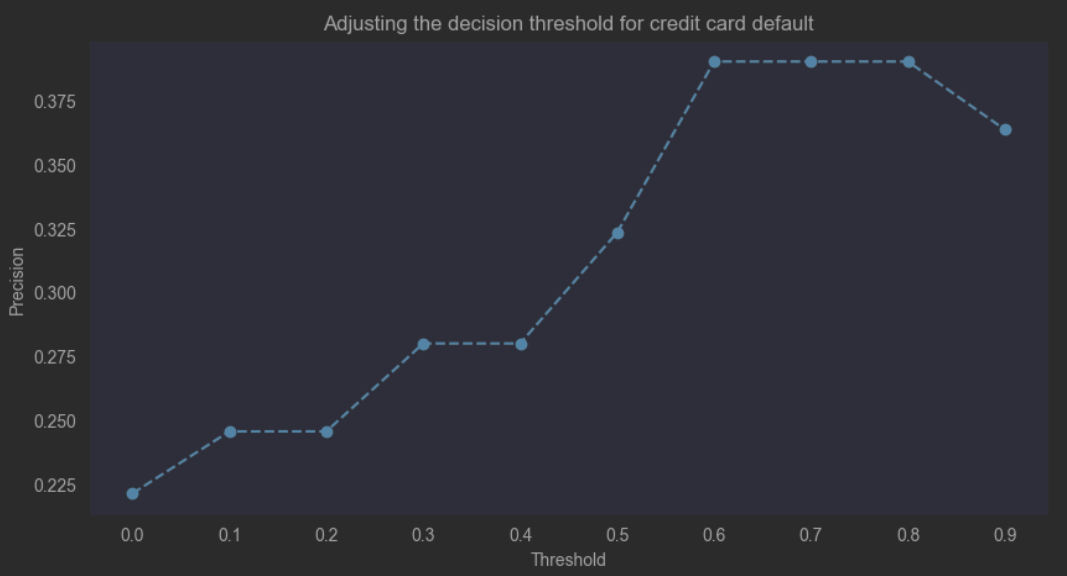
▪ Precision: If you predict that a person will pay its credit debt, what is the chance that you are correct?

▪ Recall: If a person that will pay exists, what is the chance that you correctly identify them as someone who will pay?

In a business context, for example, both these metrics are important. High precision means that customers who receive loans are likely to pay them back. High recall means that potential quality customers are not overlooked.

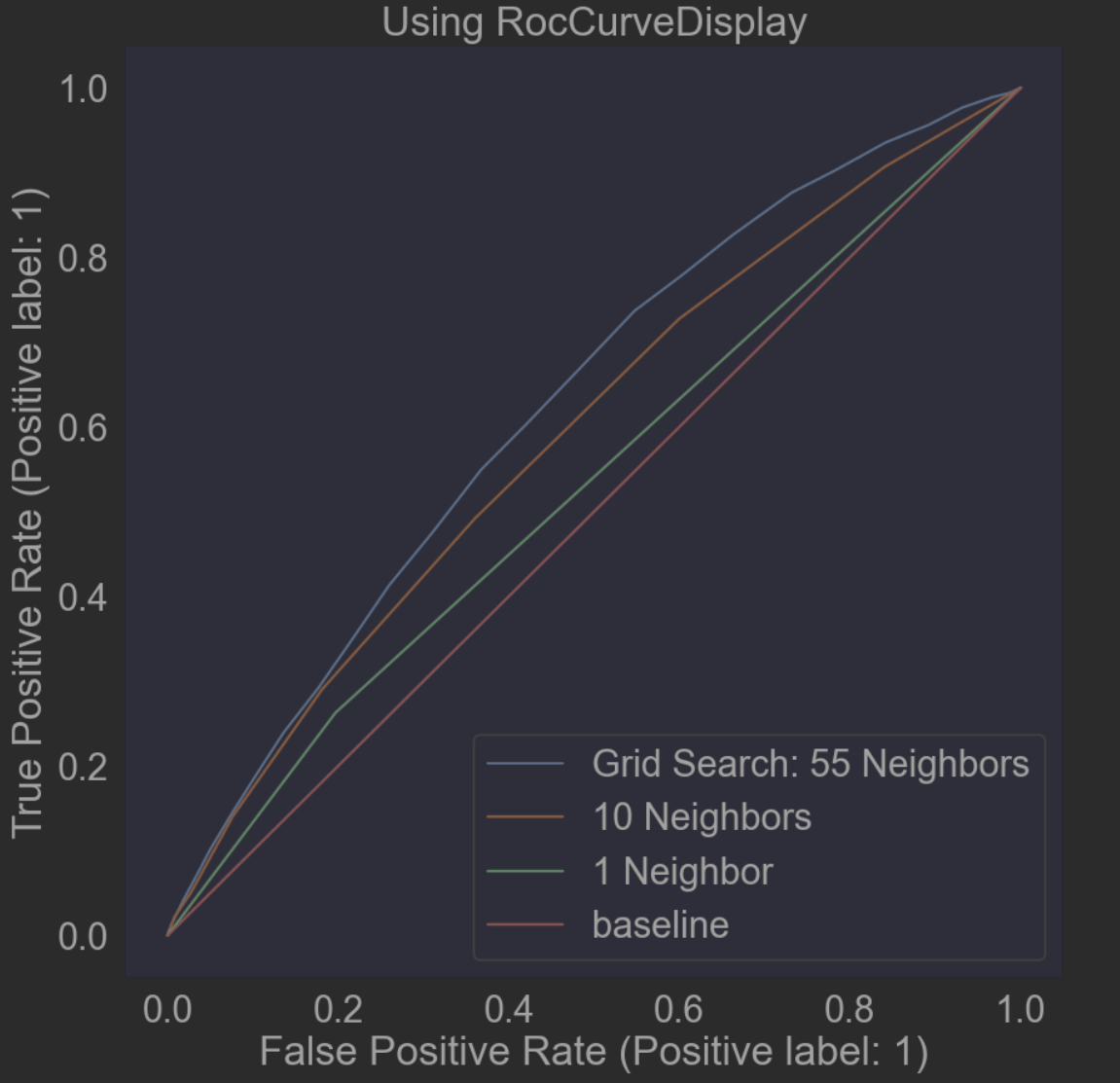


Scikit-learn provides a useful function called **precision\_recall\_curve** that is used to generate precision and recall values for various classifier thresholds.



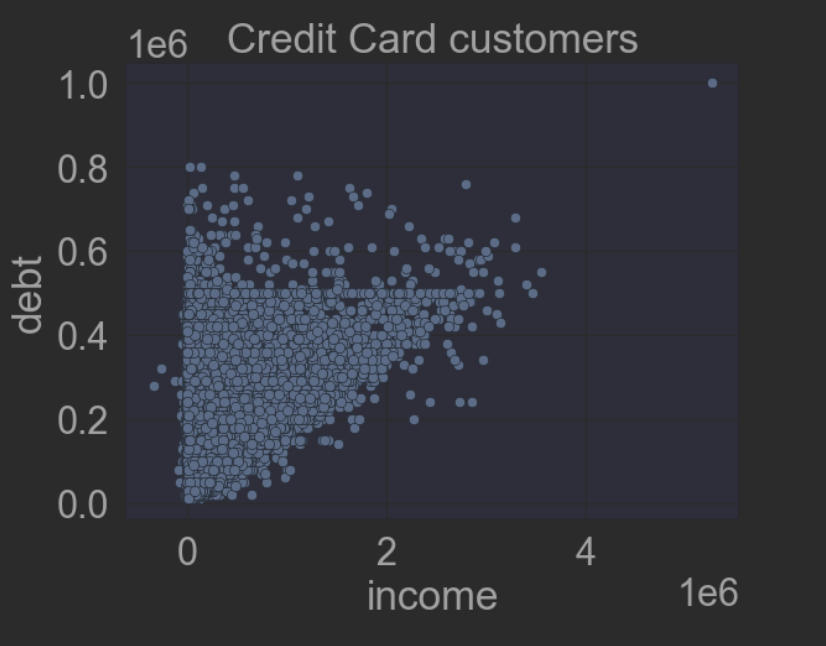
Recall that a probability threshold (T) defines the cut-off point which separates classes from one another.

A common trade-off in machine learning is between the recall and one minus specificity. This alternate trade-off is also known as a **receiver operator characteristic curve (ROC curve)**. In the context of a ROC curve, the recall is often called the **true positive rate**, while one minus specificity is called the **true negative rate**.



Regression

In scikit-learn I will use the sklearn.neighbors.KNeighborsRegressor to solve regression problems, Consider this plot, which is used to predict debt from income on the credit card default dataset.



When using the KNeighborsRegressor, it is important to note the:

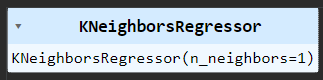
▪ Name of the class

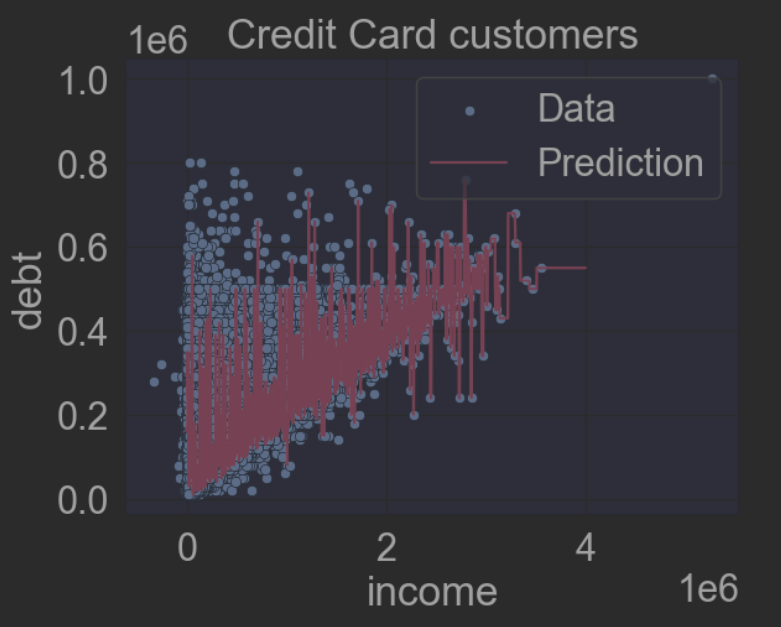
▪ Names of the input and the output columns

The regressor outputs a numerical value rather than a label.

Suppose we have a new customer whose income is $150,800. What debt level will your model predict?

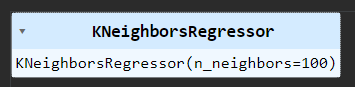
The nearest-neighbors model identifies the single nearest neighbor where k = 1.

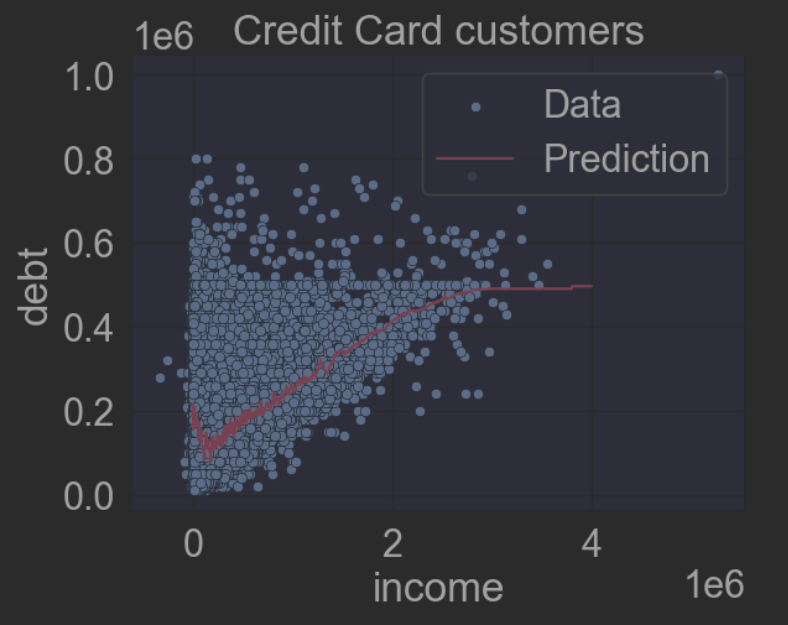




In this plot, the nearest neighbour is the rightmost point in the original dataset, which results in a predicted value of $ 210000.0.

If we increase complexity of the model, k neighbors to 100, we see a decrease in the variance and predicted value.





If we have a new customer whose income is $150,800, according the above model the predicted value debt allow is 106800.0

Hi Jim,

You can use K-Nearest Neighbours for classification and regression.

For predicting a value ( the price of a vehicle) use KNeighborsRegressor.

For predicting the type of the vehicle or the manufacture of the vehicle which it belongs, that is classification you use KNeighborsClassifier.

In a regression problem the goal is to predict a real-valued outcome, given a set of features. In a classification problem, the goal is to predict the class to which a sample belongs, given a set of features.

Hi Vani,

Very good analysis, great work!

Another alternative of accuracy metric, one could explore trade-offs among these metrics (precision and recall).

Hi Vincent,

Very innovative plot and a good analysis, a great work!

FYI

The accuracy is usually used to assess the quality of a classifier. But accuracy is not always the right metric. This is an example of a situation with imbalanced classes.

Another alternative is to focus on the trade-offs between metrics instead. One of the most commonly explored trade-offs among these metrics, is between precision and recall.