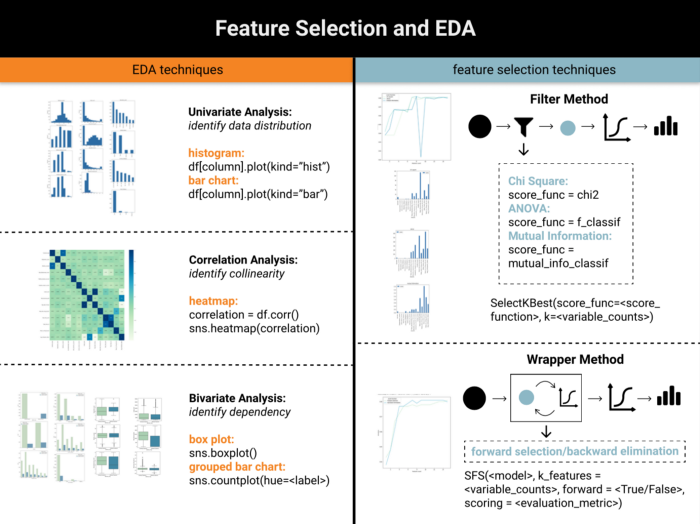
**Feature Selection and EDA in Machine Learning**

***How to Use Data Visualization to Guide Feature Selection***



Feature Selection and EDA Cheatsheet (image by author, from [website](https://www.visual-design.net/post/feature-selection-and-eda-in-machine-learning))

In Machine Learning Lifecycle, feature selection is a critical process that selects a subset of input features that would be relevant to the prediction. Including irrelevant variables, especially those with bad data quality, can often contaminate the model output.

Additionally, feature selection has following advantages:

1) **avoid the curse of dimensionality**, as some algorithms perform badly when high in dimensionality, e.g. general linear models, decision tree

2) **reduce computational cost**and the complexity that comes along with a large amount of data

3) **reduce overfitting** and the model is more likely to be generalized to new data

4) **increase the explainability** of models

In this article, we will discuss two main feature selection techniques: **Filter Methods**and **Wrapper Methods**, as well as how to take advantage of data visualization to guide decision making.

**Data Preprocessing**

Before jumping into the feature selection, we should always load the dataset, perform data preprocessing and data transformation:

**1. Load dataset and import libraries**

I am using [Credit Card Customer](https://www.kaggle.com/sakshigoyal7/credit-card-customers) Dataset from Kaggle to predict who is more likely to get churned.

import pandas as pd   
import numpy as np   
import seaborn as sns   
import matplotlib.pyplot as plt   
from pandas.api.types import is\_string\_dtype, is\_numeric\_dtype df = pd.read\_csv("../input/credit-card-customers/BankChurners.csv")

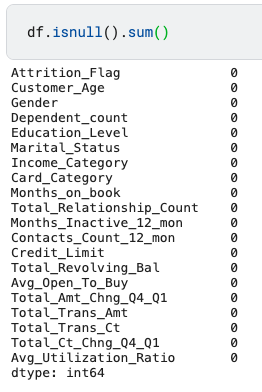
Now let’s have a glimpse of the raw data.



Credit Card Customer Dataset (image by author)

**2. Define the prediction:** this exercise is a classification problem, from which we predict the binary variable “Attrition\_Flag” which can either be existing customers or attrited customers.

**3. Examine missing data:**



missing data (image by author)

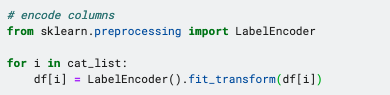
Luckily, this dataset does not contain any missing values, but it is not always the case. If you would like to know more about how to address missing data, you may find this article helpful.

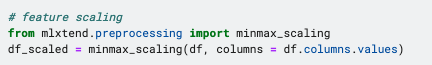
**[How to Address Missing Data](https://medium.com/analytics-vidhya/how-to-address-missing-data-531ed964e68" \t "_blank)**

[Solutions to Three Types of Missing Data](https://medium.com/analytics-vidhya/how-to-address-missing-data-531ed964e68" \t "_blank)

[medium.com](https://medium.com/analytics-vidhya/how-to-address-missing-data-531ed964e68" \t "_blank)

**4. Variable transformation:** This process consists of encoding categorical variables and transforming all variables into the same scale. I chose label encoder and min-max scaling respectively.





Variable Transformation (image by author)

Check out this article if you would like to know more data transformation technique:

**[Data Transformation and Feature Engineering](https://towardsdatascience.com/data-transformation-and-feature-engineering-e5181ef274b5" \t "_blank)**

[How to Choose the Appropriate Technique for Your Data](https://towardsdatascience.com/data-transformation-and-feature-engineering-e5181ef274b5" \t "_blank)

[towardsdatascience.com](https://towardsdatascience.com/data-transformation-and-feature-engineering-e5181ef274b5" \t "_blank)

**Exploratory Data Analysis (EDA)**

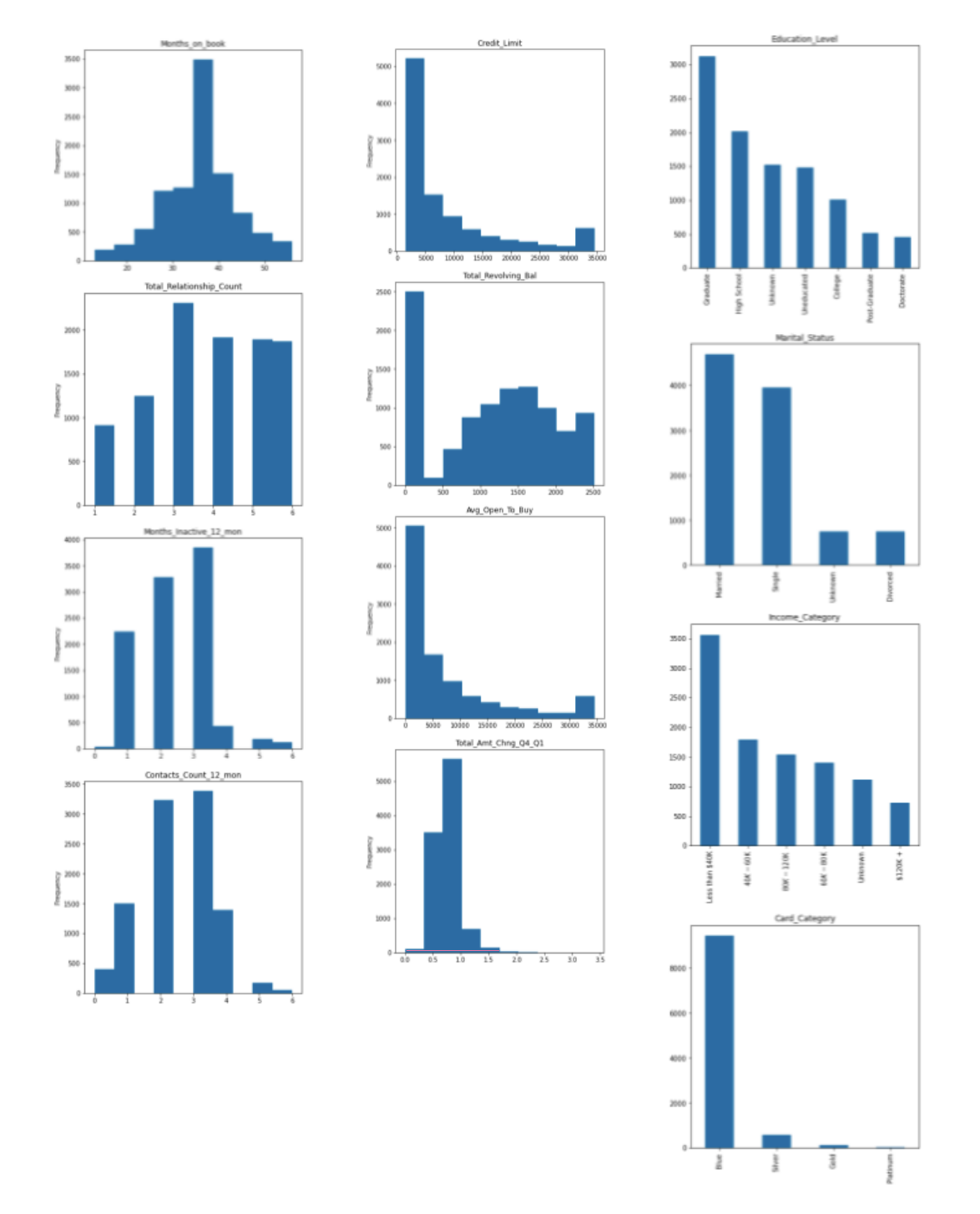
Data visualization and EDA can be great complementary tools to feature selection process, and they can be applied in the following ways:

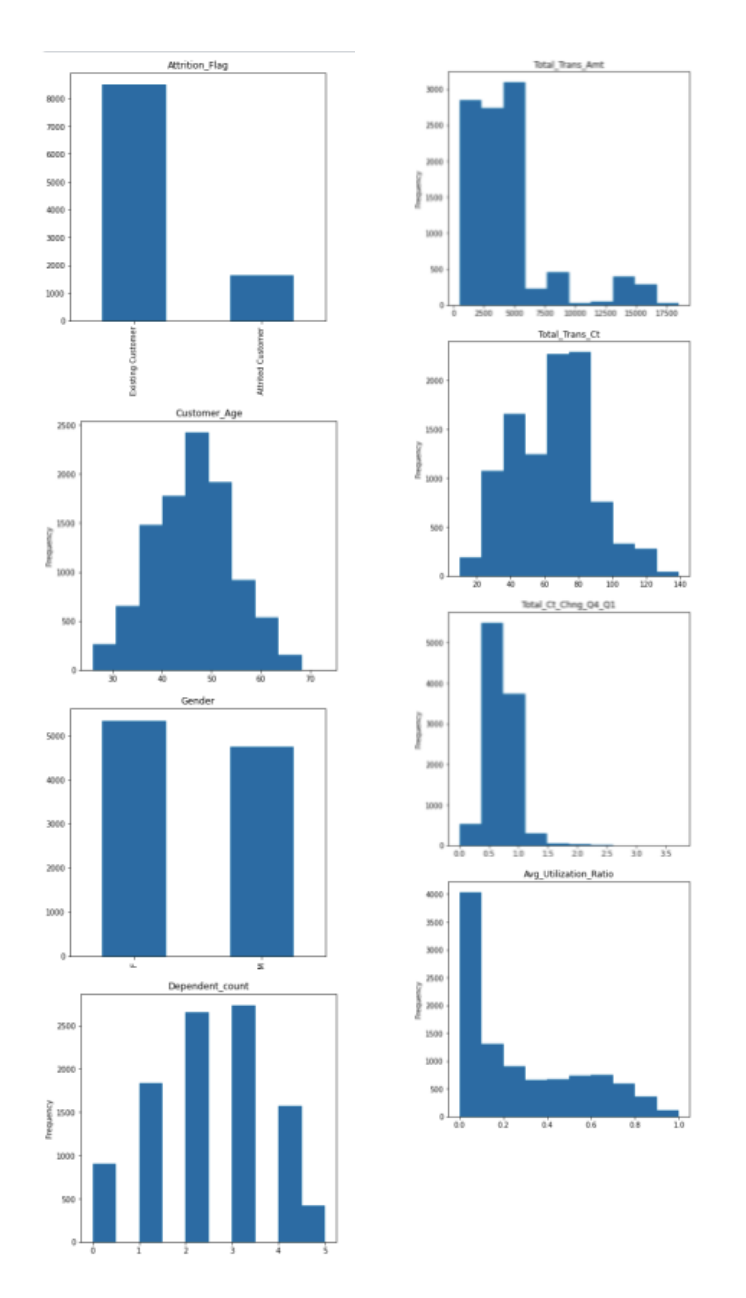
1. **Univariate Analysis: Histogram**and**Bar Chart** help to visualize the distribution and variance of each variable
2. **Correlation Analysis: Heatmap** facilitates the identification of highly correlated explanatory variables and reduce collinearity
3. **Bivariate Analysis: Box plot**and **Grouped bar chart**help to spot the dependency and relationship between explanatory variables and response variable

*(to have a better understanding of the data, exploratory data analysis can be performed before data transformation)*

**Univariate Analysis — Histogram and Bar Chart**

To obtain an overview of distribution, firstly let’s classify features in to categorical and numerical variables, then visualize categorical features using bar chart and numerical features using histogram. Visualizing the distribution gives suggestions on whether the data points are more dense or more spread out, hence whether the variance is low or high. Low variance features tend to contribute less to the prediction of outcome variable.

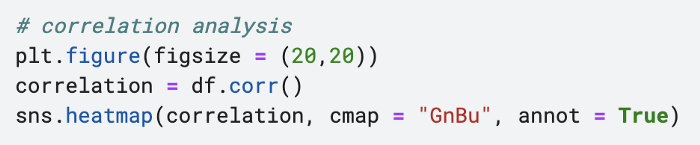


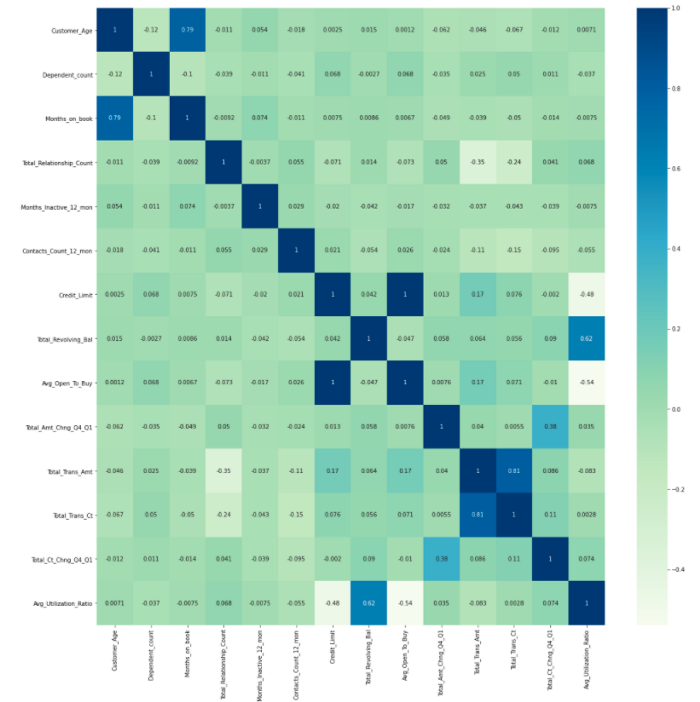


Univariate Analysis (image by author)

**Correlation Analysis — Heatmap**

Some algorithms demands the absence of collinearity in explanatory variables, including logistic regression which will be used for this exercise. Therefore, eliminating highly correlated features is an essential step to avoid this potential pitfall. Correlation analysis with heatmap visualization highlights pairs of features with high correlation coefficient.



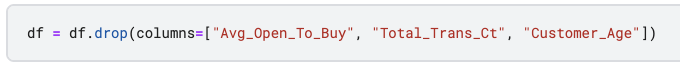


Correlation Analysis and Heatmap (image by author)

As shown, it is not hard to find following pairs of highly correlated features:

* *Customer\_Age & Month\_on\_Book (0.79)*
* *Total\_Trans\_Amt & Total\_Trans\_Ct (0.82)*
* *Avg\_Open\_To\_Buy & Credit\_Limit (1)*

Based on this result, I dropped following variables:

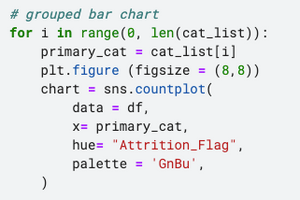


drop highly correlated features

**Bivariate Analysis — Box Plot and Grouped Bar Chart**

Bivariate EDA investigates the relationship between each explanatory variable and the target variable. Categorical features and numerical variables are addressed using **grouped bar chart** and **box plot** respectively, and this exploration can further facilitate the statistical tests used in the filter methods, e.g. Chi-Square and ANOVA test.

*G***rouped bar chart is used as the visual representation of Chi-Square Analysis**

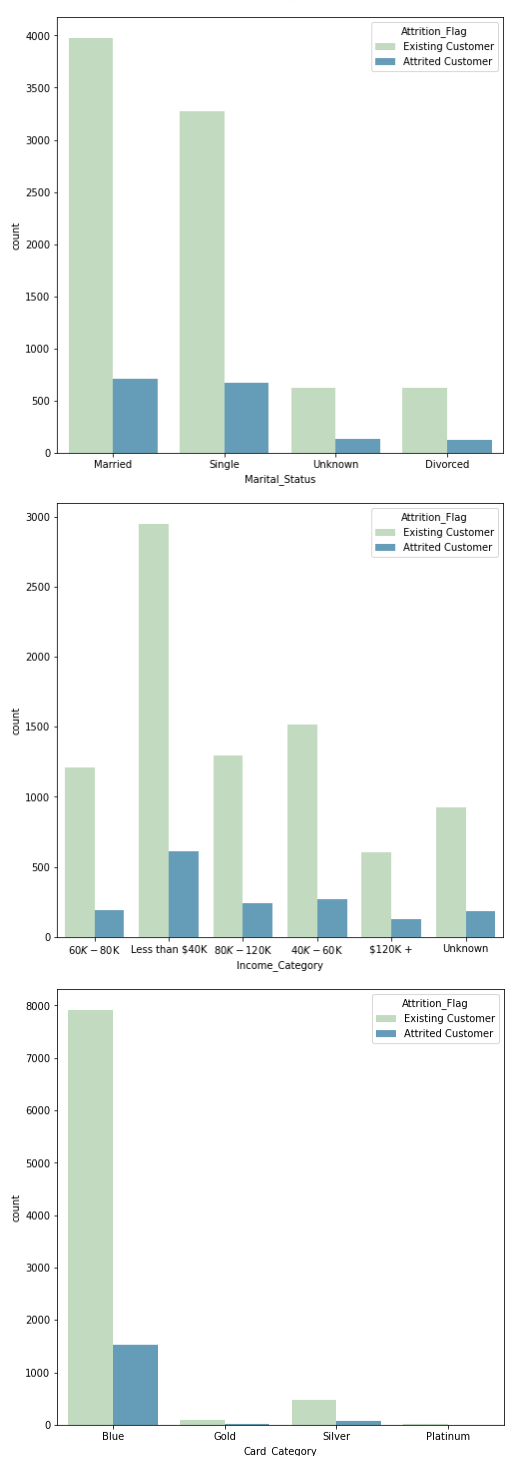


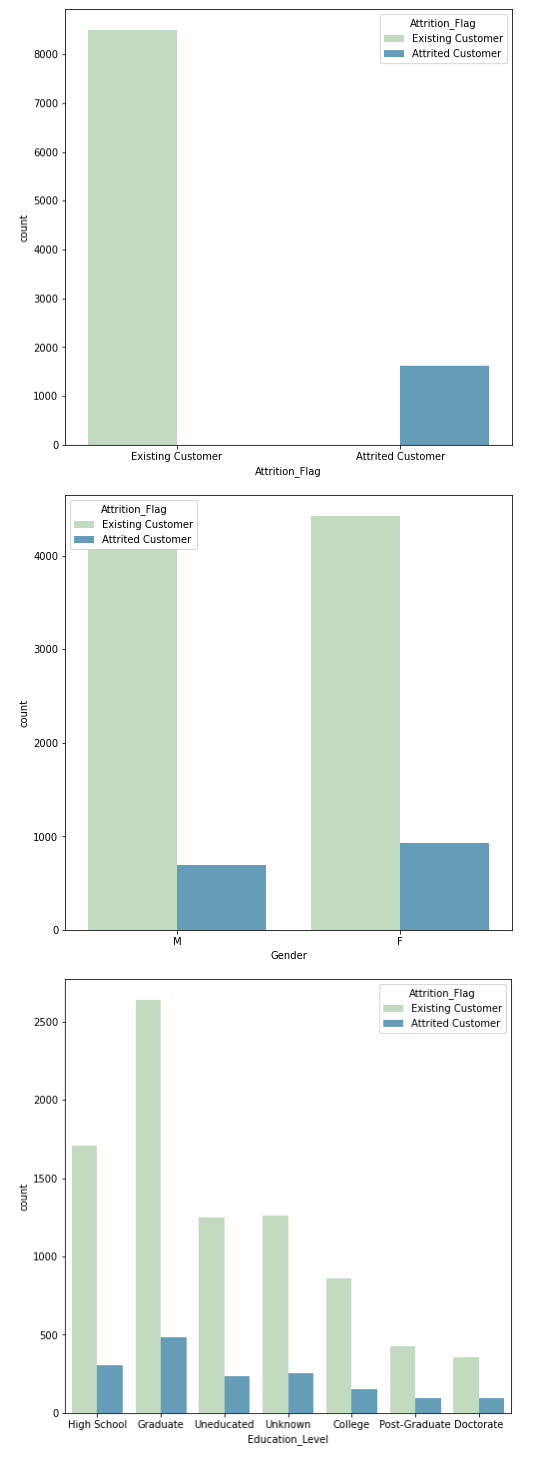
grouped bar chart code

Each independent variable is set as the primary category. The target variable is set be be the secondary category using hue = “Attrition\_Flag”. As the result, it depicts whether the “Attrition\_Flag” would vary in distribution across different level of the primary category. If two variables are independent, then we would expect the distribution to be the same across all levels.

This is the same logic as Chi-Square test which calculates the difference between the observed value and expected value based on the assumption of independency. If there are no or little dependencies existing, we expect the ratio of each group of bars to be proportional to the ratio of attrited customers vs. existing customers. If the ratio is significantly different, then it suggests a high disparity between the observed value and expected value, which means high Chi-Square value, hence rejects the hypothesis that two variables are independent.

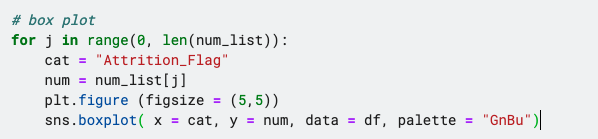
After plotting all category variables against the target label, I found that *“Card\_Category”*seems to display a variation in ratio across*Blue, Gold, Silver and Platinum*. In the following section, we will find out if this is true according to the quantitative scoring based on filter method.





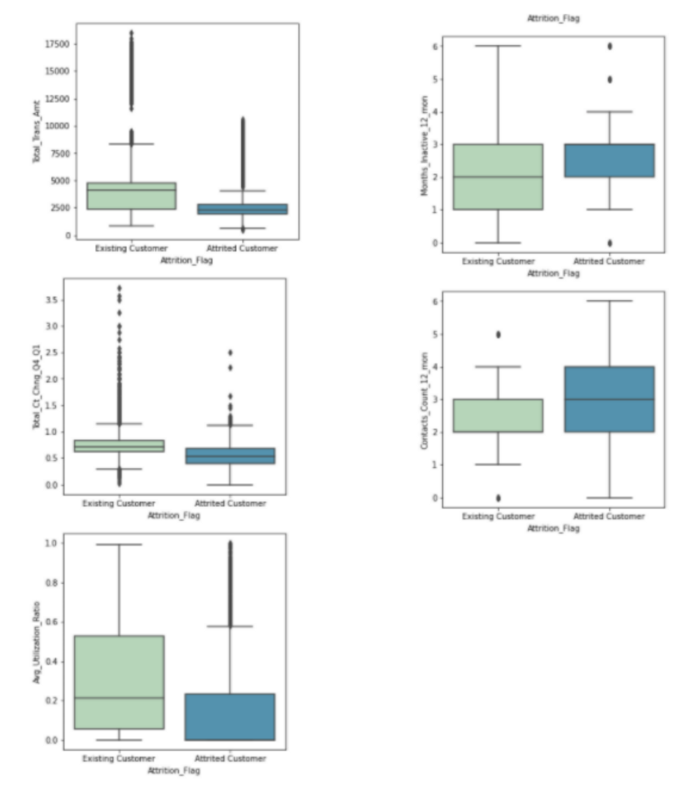
grouped bar chart results (image by author)

**Box plot is used as the visual representation of ANOVA analysis**

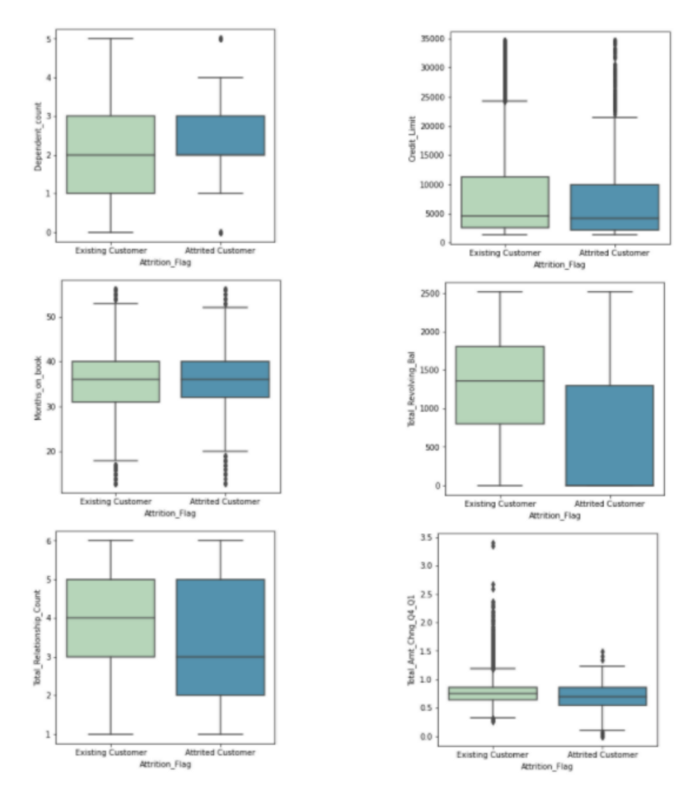


box plot code

Box plot displays the distributions of groups of numerical data through their quantiles. Each box shows how spread out the data is within group and putting boxes side by side indicates the difference among groups. It is aligned with ANOVA test which also analyze the degree of variance between-group compared to within-group. If the relative variability is large, for instance *“Total\_Revolving\_Bal”*and *“Total\_Cnt\_Chng\_Q4\_Q1”*shown below, then it may be an indication of that these features could contribute to predicting the labels. Let’s find out whether this can be quantified by the ANOVA test in filter methods.



box plot results (image by author)



box plot results (image by author)

*If you would like to have a more thorough understanding of EDA, feel free to read my article on “*[*Semi-Automated Exploratory Data Analysis (EDA) in Python*](https://towardsdatascience.com/semi-automated-exploratory-data-analysis-eda-in-python-7f96042c9809?source=your_stories_page-------------------------------------)*”.*

**Feature Selection**

If you would like to dive deeper into the importance of feature selection and different techniques to implement feature selection, the blog below from [*neptune.ai*](https://neptune.ai/home) provides a comprehensive guide.

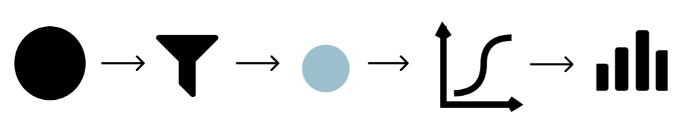
**[Feature Selection Methods and How to Choose Them - neptune.ai](https://neptune.ai/blog/feature-selection-methods" \t "_blank)**

[Have you ever found yourself sitting in front of the screen wondering what kind of features will help your machine…](https://neptune.ai/blog/feature-selection-methods" \t "_blank)

[neptune.ai](https://neptune.ai/blog/feature-selection-methods" \t "_blank)

In this article, we will mainly introduce two types of feature selection methods: **filter method and wrapper method.**The fundamental difference is that filter method evaluates the feature importance based on statistical tests such as Chi-Square, ANOVA etc, whereas wrapper method iteratively assessed the performance of subsets of features based the performance of models generated by these features.

**Filter Methods**



filter method illustration (image by author)

Filter methods give a score to each feature by evaluating its relationship with the dependent variable. For classification problems with categorical response variables, I am using these three major scoring functions: **Chi-Square (score\_func = chi2), ANOVA (score\_func = f\_classif), and Mutual Information (score\_func = mutual\_info\_classif).**To create a feature selection model, we need the *SelectKBest()*function, then specify which scoring functions to utilize and the how many variables to select.

selection\_model = SelectKBest(score\_func=score\_function, k=variable\_counts)

I would like to know how these two parameters, scoring function and the number of variables, would affect the accuracy of the model trained on the selected features.

Firstly, to create the carry out the feature selection and examine the performance of the model built upon it, I define a *feature\_selection* function with following steps:

* import required libraries
* create a feature selection model based on two parameters: score\_function (e.g. chi square) and variable counts (e.g. ranging from 1 to all features)
* train a logistic regression model based on selected features only
* calculates the accuracy score



import libraries



define feature\_selection function

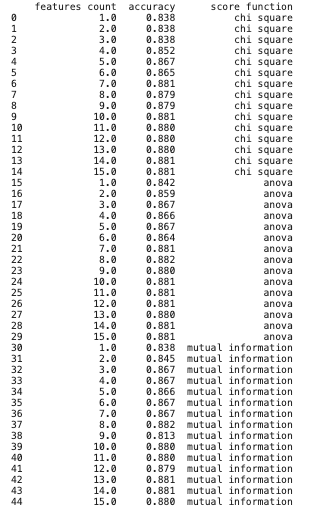
Secondly, to test how score functions and variable counts would affect the model performance, I iteratively passing different combinations of two parameters, *“variable\_counts”* and *“score\_function”,* using the following code.

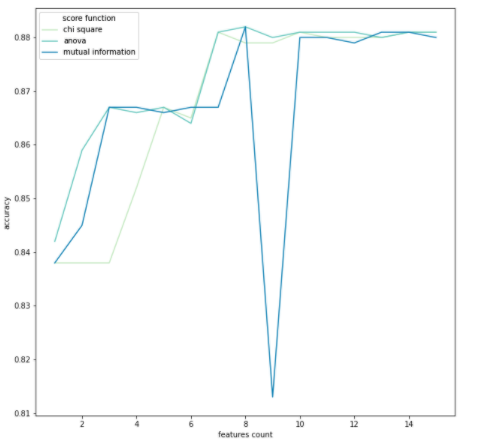


filter method accuracy chart code

**Filter Methods with Data Visualization**

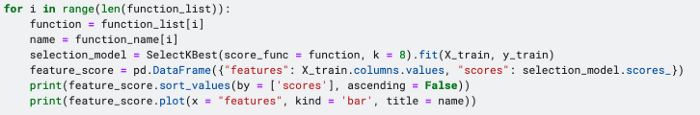
The result was generated in a data frame format and then use line chart to demonstrate how the accuracy progress as the number of selected features grows. As shown, except mutual information method, the accuracy score stabilizes around 0.88 after reaching 8 features.



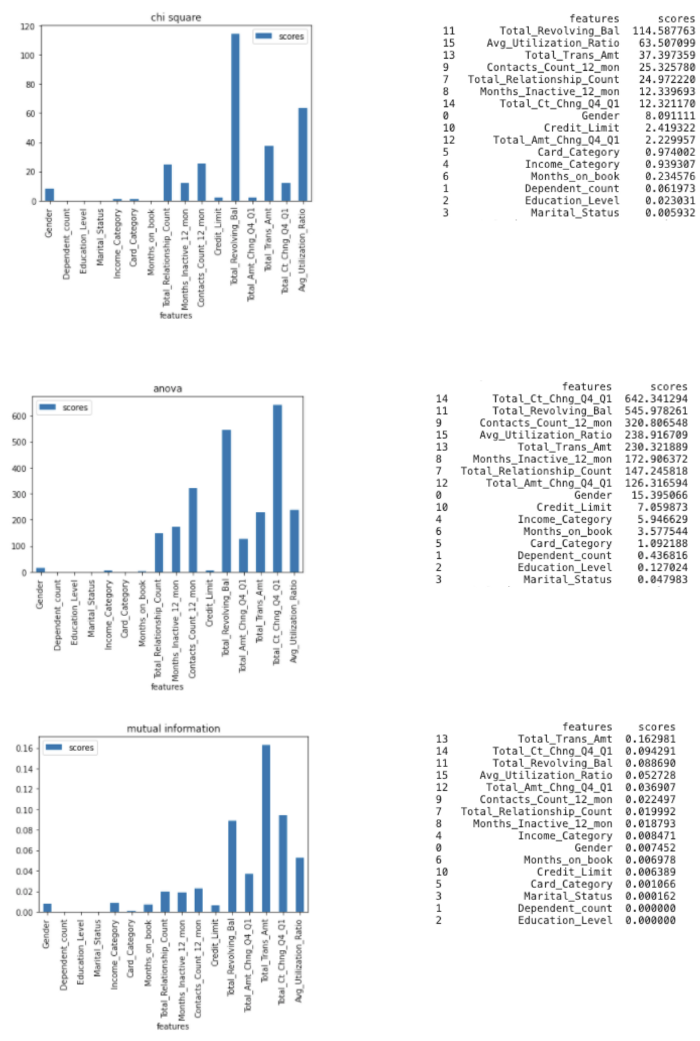


filter method accuracy (image by author)

Afterwards, let’s investigate what is the score of each feature based on various approaches. This time we will use bar charts to visualize the scores has been allocated to features according to chi-square, anova or mutual information.



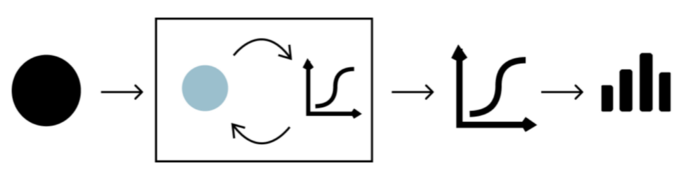
feature score code



feature score by different statistics (image by author)

As you can see, different approach scores the same feature differently, but some features always appear higher up on the list. For example, *“Total\_Revolving\_Bal”* is always among the top 3, which is aligned with the findings from the box plot in the bivariate EDA. And “Card\_Category” does have high feature importance compared to other categorical variables, which can be explained by the grouped bar chart.

**Wrapper Methods**



wrapper method illustration (image by author)

Wrapper Methods find the optimal subsets of features by evaluating the performance of machine learning models trained upon these features. Since it incorporates model into the feature selection process, it requires more computational power. This article covers two main wrapper methods,**forward selection and backward elimination.** To perform forward selection and backward elimination, we need *SequentialFeatureSelector()*function which primarily requires four parameters:

1. model: for classification problem, we can use Logistic Regression, KNN etc, and for regression problem, we can use linear regression etc
2. k\_features: the number of features to be selected
3. forward: determine whether it is forward selection or backward elimination
4. scoring: evaluation metrics to determine the model performance, e.g. classification problem — accuracy, precision, recall etc; regression problem — p-value, R-squared etc

**Forward Selection**

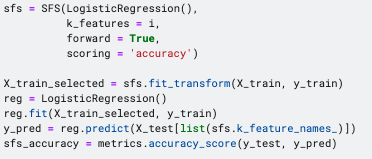
Forward Selection starts with no features in the model and incrementally adds one feature to the feature subset at a time. During each iteration, the new feature is chosen based on the evaluation of the model trained by the feature subset.

Since the machine learning model is wrapped within the feature selection algorithm, we need to specify a model as one of the input parameters. I choose Logistic Regression for this classification problem and accuracy as the evaluation metrics. There is a slight difference in calculating the accuracy in the wrapper method compared to the filter method. Since we only fit the training set to the wrapper model, the accuracy score returned by the wrapper method itself is purely based on the training dataset. Therefore, it is necessary to train an additional model on the selected features and further evaluated based on the test set.

To achieve this, I used the code below to import the required libraries, as well as create and evaluate the logistic regression model built upon the wrapper method.

https://miro.medium.com/max/460/1*e48ehp67cQfDsARYx_wnZQ.png

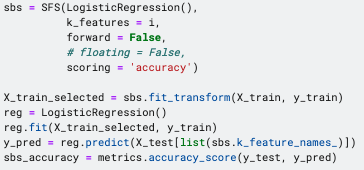
import libraries



forward selection code

**Backward Elimination**

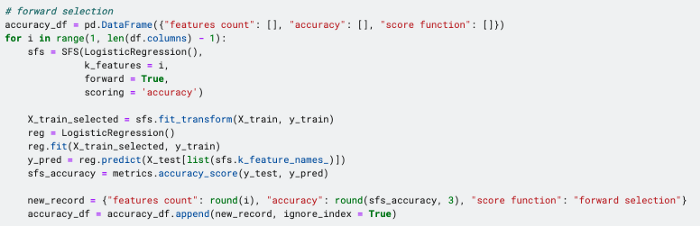
Simply put, it is just the opposite of the forward selection, starting with including all features to train the model. Then, features are iteratively removed from the feature subset based on whether they contribute to the model performance. Similarly, logistic regression and accuracy are used as the model and evaluation metrics correspondingly.



backward selection code

**Wrapper Methods and Data Visualization**

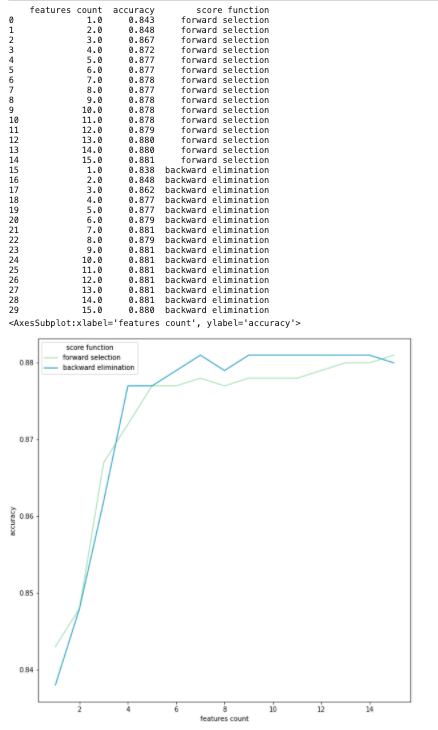
Similar to the filter method, I enveloped both forward selection and backward elimination into a “for loop”, in order to examine whether the variable counts would matter to the accuracy score.





wrapper method accuracy code

As shown in the line chart, the accuracy grows rapidly when the feature counts is less than 4 and then remains stable around 0.88 afterwards.



wrapper method accuracy chart (image by author)

In this dataset, since there are only around 20 features in total, it may be hard for feature selection to yield any significant impacts on model performance. However, it is undeniable that data visualization can help us to decide which features and how many features are suitable for the dataset or the objectives. This principle can definitely be extended to other dataset with more variables.

**Take Home Message**

This article covers two fundamental techniques of feature selection:

* **Filter Methods:** baed on chi-square, ANVOA and mutual information
* **Wrapper Methods:** based on forward selection and backward elimination

We also look at how to use data visualization to better understand the feature properties and additionally to select an appropriate number of features.