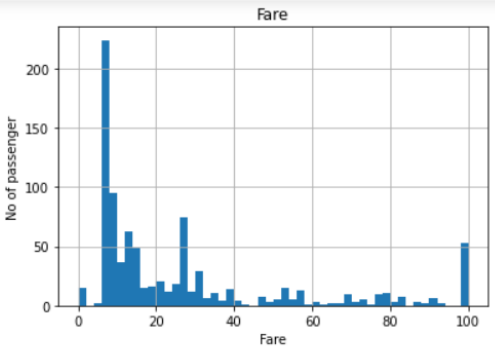
Project Name: Analyzing and Handling Outliers

Table of Contents

Demo
Overview
Motivation
Technical Aspect
Installation
Run/How to Use/Steps
Directory Tree/Structure of Project
To Do/Future Scope
Technologies Used/System Requirements/Tech Stack
Credits

Demo



In [22]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(data[['Age','Fare']].fillna(0),data['Survived'],test_size=0.3)
```

In [23]:
```
### Logistic Regression
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(X_train,y_train)
y_pred=classifier.predict(X_test)
y_pred1=classifier.predict_proba(X_test)

from sklearn.metrics import accuracy_score,roc_auc_score
print("Accuracy_score: {}".format(accuracy_score(y_test,y_pred)))
print("roc_auc_score: {}".format(roc_auc_score(y_test,y_pred1[:,1])))
```

```
Accuracy_score: 0.6828358208955224
roc_auc_score: 0.6874091569767441
```

Overview

This is diving into Outliers Concept.
Outlier tell us when a value is too far from the middle. An outlier is a point which falls more than 1.5 times the interquartile range above the third quartile or below the first quartile. A data point that lies outside the overall distribution of dataset.
This repository contains the code for Handling Outliers using python's various libraries.
It used pandas, seaborn and sklearn libraries.

These libraries help to perform individually one particular functionality.
Pandas objects rely heavily on Numpy objects.
Sklearn has 100 to 200 models.
Seaborn is data visualization library based on matplotlib.
The purpose of creating this repository is to gain insights into how to analyze and handle outliers.
These python libraries raised knowledge in discovering these libraries with practical use of it.
It leads to growth in my ML repository.
This above screenshot will help you to understand flow of output.

Motivation

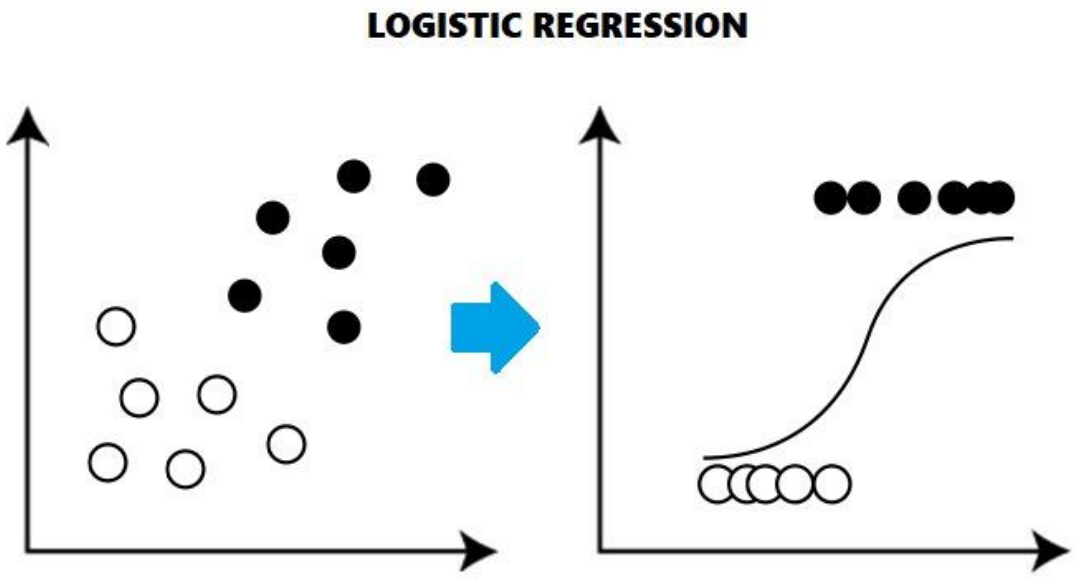
The reason behind building this is, Outliers can be very informative about the subject-area and data collection process. And excluding outliers can cause your results to become statistically significant. Outliers is tricky topic because as The Data Science project starts with collection of data and that's when outliers first introduced to the population. Though, you will not know about the outliers at all in the collection phase. The outliers can be a result of a mistake during data collection or it can be just an indication of variance in your data. it is pretty simple if they are the result of a mistake, then we can ignore them, but if it is just a variance in the data, we would need think a bit further. Before we try to understand whether to ignore the outliers or not, we need to know the ways to identify them which you can refer in credit section. An outlier can cause serious problems in statistical analyses. Outlier points can therefore indicate faulty data, erroneous procedures, or areas where a certain theory might not be valid. However, in large samples, a small number of outliers is to be expected. There is difference between outlliers and anomalies, Outlier = legitimate data point that's far away from the mean or median in a distribution. Anomaly detection refers to the problem of ending anomalies in data. While anomaly is a generally accepted term, other synonyms, such as outliers are often used in different application domains. Outlier is an extreme value in a set of data which is much higher or lower than the other numbers. Outliers affect the mean value of the data but have little effect on the median or mode of a given set of data. There are different types of outliers. During data analysis when you detect the outlier one of most difficult decision could be how one should deal with the outlier. Should they remove them or correct them? Therefore, getting hands-on with outliers. Another major aim is, as I told even in my previous projects, I try to relate IT and Life. So, in my house also my brother is kind of outlier as he is not good with academics but rather, I am comparatively better than him but because of same I had seen many partial scenarios which I would have otherwise not so he always used to be outlier in his class with other mates. These are the reasons that encourage me to take up and work on such concepts. Hence, I continue to gain knowledge while practicing the same and spread literary wings in tech-heaven.

Technical Aspect

Pandas module mainly works with the tabular data. It contains Data Frame and Series. Pandas is 18 to 20 times slower than Numpy. Pandas is seriously a game changer when it comes to cleaning, transforming, manipulating and analyzing data.
Seaborn provides a high-level interface for drawing attractive and informative statistical graphics. It provides a variety of visualization patterns and visualize random distributions.
Sklearn is known as scikit learn. It provides many ML libraries and algorithms for it. It provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.
Need to train_test_split - Using the same dataset for both training and testing leaves room for miscalculations, thus increases the chances of inaccurate predictions.
The train_test_split function allows you to break a dataset with ease while pursuing an ideal model. Also, keep in mind that your model should not be overfitting or underfitting.
Accuracy_score: Refer to below link.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

The AUC for the ROC can be calculated using the roc_auc_score() function. It returns the AUC score between 0.0 and 1.0 for no skill and perfect skill respectively.
Logistic regression is appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). I used simple logistic regression here because when you have one nominal variable and one measurement variable, and you want to know whether variation in the measurement variable causes variation in the nominal variable.



Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.
Windows 10 Environment Used.
Already Installed Anaconda Navigator for Python 3.x
The Code is written in Python 3.8.

If you don't have Python installed then please install Anaconda Navigator from its official site.
If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter*.

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.
Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

cd <PATH>
pip install pandas
pip install seaborn
pip install sklearn

You can also create requirement.txt file as, pip freeze > requirements.txt

run files.

Follow this when you want to just perform on local machine.

Download ZIP File.
Right-Click on ZIP file in download section and select Extract file option, which will unzip file.
Move unzip folder to desired folder/location be it D drive or desktop etc.
Open Anaconda Prompt, write cd <PATH> and press Enter.
eg: cd C:\Users\Monica\Desktop\Projects\Python Projects 1\15)Feature_Engineering+Selection+Importance\Outliers

In Anconda Prompt, pip install -r requirements.txt to install all packages.
Open in Jupyter Notebook, <filename>.ipynb
That is,

Open in Jupyter Notebook, 1)Analyzing_and_Handling_Outliers.ipynb
This takes titanic.csv file as input dataset.
Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.
Note: cd <PATH>
[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to cd one space <path> and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

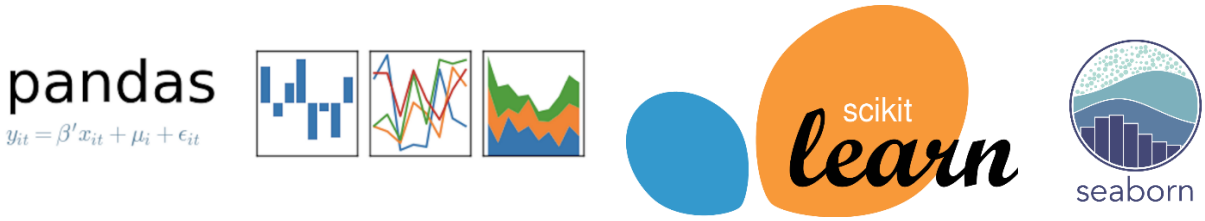
Folder: 15)Feature_Engineering+Selection+Importance > Outliers

1)Analyzing_and_Handling_Outliers.ipynb

To Do/Future Scope

Can try other dataset with same techniques to take decision and impact.

Technologies Used/System Requirements/Tech Stack



Credits

Krish Naik Channel

https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba

https://medium.com/analytics-vidhya/outlier-treatment-9bbe87384d02

https://www.analyticsvidhya.com/blog/2019/02/outlier-detection-python-pyod/

https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/