ENTIRE MACHINE LEARNING PROCESS AND HOW TO GO ABOUT IT:

NOTE: The **bias** of an estimator is its average error for different training sets. The **variance** of an estimator indicates how sensitive it is to varying training sets. **Noise** is a property of the data.

1. IMPORT THE LIBRARIES:

* Initial libraries

Import numpy as np

Import pandas as pd

Import warnings

Import matplotlib.pyplot as plt

Import seaborn as sns

* Libraries we use for the project

e.g

Import scipy

Import stats.models.api as sm

From scipy import stats

1. DEAL WITH WARNINGS

Warnings.filterwarnings(‘ignore’)

1. IMPORT THE DATASET

Pd.read\_csv(‘File\_Name’)

1. EDA

* Initial analysis

Dataset.info()

Dataset.head()

Dataset.tail()

Dataset.describe()

Dataset.describe(include = ‘all’)

Dataset.nunique()

Dataset.isnull().sum()

Dataset.isnull().sum().sum()

Dataset.corr()

* Visualization

Dataset.hist()

Sns.heatmap()

1. DATA CLEANING

* Search and deal with missing columns. Use MEAN, MEDIAN, MODE dependent on how the distribution looks
* Remove redundant columns

1. DATA TRANSFORMATION

* Turn categorical data to numerical data.

From sklearn.preprocessing import LabelEncoder, OneHotEncoder

From sklearn.compose import ColumnTransformer

OR

USE GET DUMMIES from pandas

Pd.get\_dummies()

1. DATA SCALING

* Scale the data to equal units. Remember that if you scale the INDEPENDENT VARIABLES (x), then there is a must need to scale the DEPENDENT VARIABLE(Y) as well. This is only applicable in REGRESSION ANALYSIS and it is compulsory, to avoid bias in the model

From sklearn.preprocessing import StandardScaler

1. PANDAS PROFILING

* Get a profile report on your cleaned data

From ydata\_profiling import ProfileReport

1. SUMMARIZE PANDAS PROFILE

* Pay attention to conclusions drawn from pandas profile while summarizing findings

1. SPLIT THE DATA INTO DEPENDENT AND INDEPENDENT VARIABLES

Dataset.loc[]

Dataset.iloc[]

Dataset[[]]

Datset[]

1. DEALING WITH BALANCED AND UNBALANCED DATASETS

* This problem is peculiar to CLASSIFICATION PROBLEMS only. If you’re doing regression, then you can skip this step. A balanced dataset signifies that the DEPENDENT VARIABLES or the VECTOR Y doesn’t have equal classes to predict. To what extent do we classify a dataset as UNBALANCED?

Well, if any class is less than 0.6 of the other class, the dataset is considered to be an UNBALANCED dataset and we will need to fix this in other to avoid bias in our prediction and modelling

HOW DO WE FIX THIS

1. Using Algorithms

These include:

- ***Ensemble Methods/Algorithms:***

This is a technique in machine learning that is used to combine multiple base models to improve accuracy and robustness of predictions. It is generally divided into two types:

1. Bagging:

In this process, multiple independent models are trained on random subsets of the data and the predictions are combined together for the final prediction. Bagging can be used on any type of base model however, it is mostly used with decision tress to create the RANDOM FOREST ALGORITHM.

TYPES

- Random Forest Algorithm

1. Boosting:

In this process, multiple independent models are trained SEQUENTIALLY with each model correcting the errors of the previous model. Boosting can be used with any type of base model but it is commonly used with decision trees to create GRADIENT BOOSTING and with linear regression to create ADA BOOST

TYPES

- Gradient Boosting Algorithm

- Ada Boosting Algorithm

***- Deep Learning***

1. Sampling Techniques

These include:

- **SMOTE (Synthetic Minority Over-Sampling Technique):**

SMOTE generates new examples by interpolating between existing minority class examples.

- OverSampling

Oversampling involves adding copies of minority class examples to the dataset

- UnderSampling

Undersampling involves removing some examples from the majority class. The most common resampling techniques are random oversampling and random undersampling.

\*Check the PDF titled Supervised learning guide and codes on how to FIX UNBALANCED DATA SET USING SAMPLING code

1. FEATURE SELECTION
2. Use feature selection to select the best features for the machine learning process

- You could use SelectKBest & f\_classif/f\_regression

- You could use SelectFromModel and the ESTIMATOR you want

- You could use RFE(Recursive Feature Elimination) and the ESTIMATOR you want

- You could use LASSO regression that gives a penalty or assigns weights to features based on their importance

1. Use Stats.models.api as sm to:

- run your linear regression

- get the model summary

- use backward propagation to get the best desired features while sticking with a specified p-value of maybe 5%

12)