Amex AI/ML Hackathon- Geek Goddess 2021 Submission and Analysis

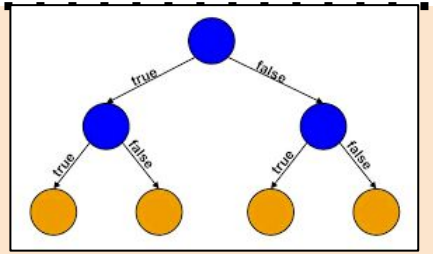
Shreya Pawaskar

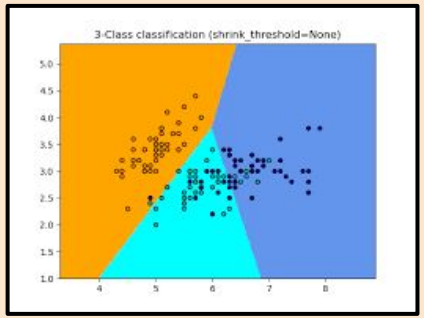
shreya.pawaskar@cummincollege.in

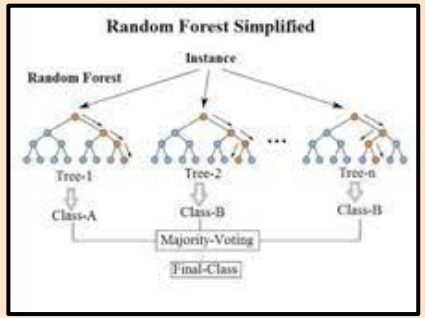
+91 9527468625

Approach taken to create the model.

- First Import the data and analyse the columns.
- Visualize your data.
- Do basic EDA
- Check for correlations between the different characteristics.
- Choose the model after trying on different algorithms
- Train your machine learning model with algorithms like
  - Decision trees
  - Gaussian NB
  - KNN
  - Random Forest
  - XGBoost
  - Logistic Regression
  - Multilayer Perceptron
  - Neural network using Tensorflow



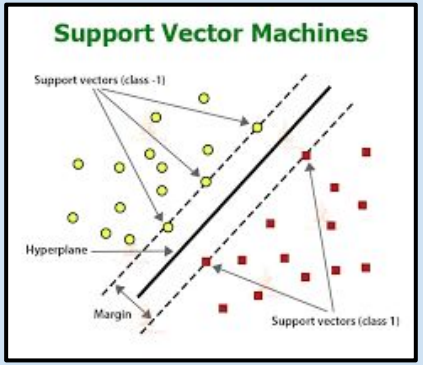


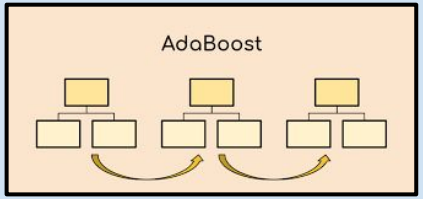


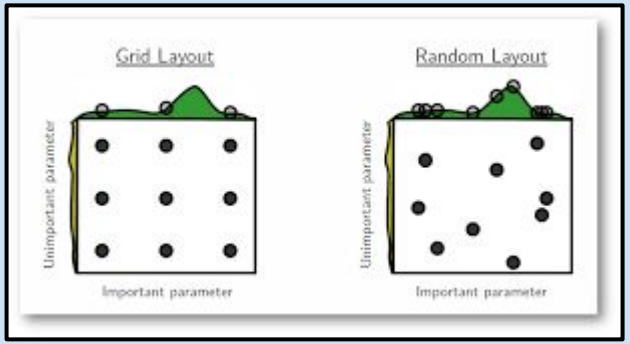
Approach taken to create the model.

- AdaBoost
- XGBoost
- SVM

- Check the model created against your evaluation
- Do Parameter Tuning
  - Grid Search CV
  - Random Search CV
- Test the machine learning model







Feature Engineering

Feature engineering is the process of using domain knowledge to extract features characteristics, properties, attributes) from raw data. An Exploratory Data Analysis is followed by a feature engineering/data augmentation step where you work on the initial data to bring them additional value.

The functions used:

1. The replace function in pandas dynamically replaces current values with the given values. The new values can be passed as a list, dictionary, series, str, float, and int.
2. LabelEncoder can be used to normalize labels. It can also be used to transform non-numerical labels to numerical labels.

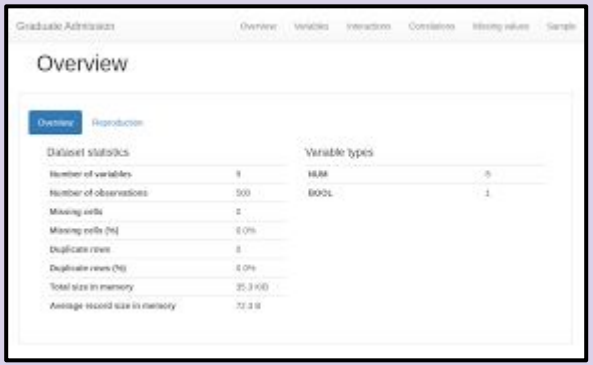
CORRELATION

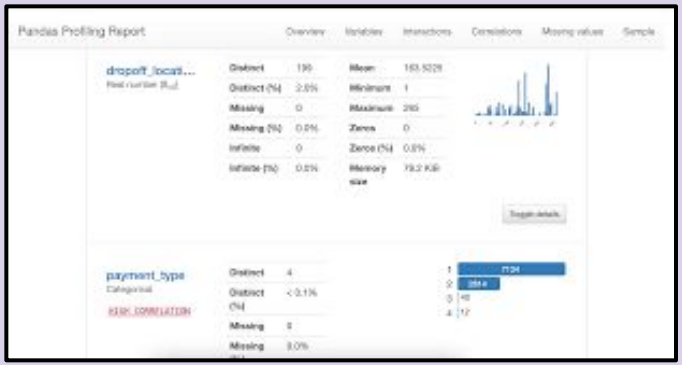
```
corr = df.corr()
corr.style.background_gradient(cmap='PuBu')
```

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
age	1.000000	-0.027648	-0.006649	-0.018031	0.027550	-0.063935	-0.024199	-0.038010	-0.069362	-0.005605
duration	-0.027648	1.000000	-0.094163	0.014566	-0.022384	0.014814	-0.023239	0.064134	0.022276	-0.033808
campaign	-0.006649	-0.094163	1.000000	0.014663	-0.051239	0.132746	0.194202	-0.119033	0.128719	0.094227
pdays	-0.018031	0.014566	0.014663	1.000000	-0.256894	0.102195	0.055753	0.024429	0.106235	0.033981
previous	0.027550	-0.022384	-0.051239	-0.256894	1.000000	-0.397809	-0.217028	-0.095092	-0.413536	-0.132276
emp.var.rate	-0.063935	0.014814	0.132746	0.102195	-0.397809	1.000000	0.553989	-0.024363	0.992830	0.635722
cons.price.idx	-0.024199	-0.023239	0.194202	0.055753	-0.217028	0.553989	1.000000	-0.644952	0.538831	0.368753
cons.conf.idx	-0.038010	0.064134	-0.119033	0.024429	-0.095092	-0.024363	-0.644952	1.000000	0.063235	-0.533119
euribor3m	-0.069362	0.022276	0.128719	0.106235	-0.413536	0.992830	0.538831	0.063235	1.000000	0.539383
nr.employed	-0.005605	-0.033808	0.094227	0.033981	-0.132276	0.635722	0.368753	-0.533119	0.539383	1.000000

Pandas-profiling

- Pandas-profiling brings all the bricks together to a complete EDA: Most frequent values, missing values, correlations, quantile and descriptive statistics, data length and more.
- In short, what pandas profiling does is save us all the work of visualizing and understanding the distribution of each variable.
- You'll quickly see the distribution and disparity of your data.





Choice of model Algorithm on the Training Dataset and why ?

- The algorithm chosen is **Random Forest Classifier.**
- Random Forest is based on the bagging algorithm and uses Ensemble Learning technique.
- It creates as many trees on the subset of the data.
- Trains each tree independently, using a random sample of the data.
- This randomness helps to make the model more robust than a single decision tree.
- It gave the best results (accuracy (10-fold): 0.96).

Benefits:

- May change considerably by a small change in the data
- Impressive in Versatility
- Robust to outliers
- Lower risk of overfitting
- It is flexible to both classification and regression problems.
- It works well with both categorical and continuous values.
- It automates missing values present in the data.

COMPARISON WITH OTHER CLASSIFIERS

GaussianNB accuracy %: 17.737296260786195

GaussianNB roc_auc_score : 0.56841046277666

GaussianNB f1_score : 0.10251046025104604

KNN accuracy % : 95.78139980824545

KNN roc_auc_score : 0.6965363610232825

KNN f1_score : 0.47619047619047616

DecisionTreeClassifier accuracy %: 95.49376797

DecisionTreeClassifier roc_auc_score : 0.7435326243

DecisionTreeClassifier f1_score : 0.5154639175

SVM accuracy % : 96.35666347075743

SVM roc_auc_score : 0.631647025007186

SVM f1_score : 0.40625

KMEANS accuracy % : 94.24736337488015

KMEANS roc_auc_score : 0.9019114688128773

KMEANS f1_score : 0.5833333333333334

MLP accuracy % : 92.90508149568552

MLP roc_auc_score : 0.9433745329117562

MLP f1_score : 0.5595238095238095

Performance and Accuracy

RANDOM FOREST

accuracy % : 95.78139980824545

roc_auc_score : 0.5704225352112675

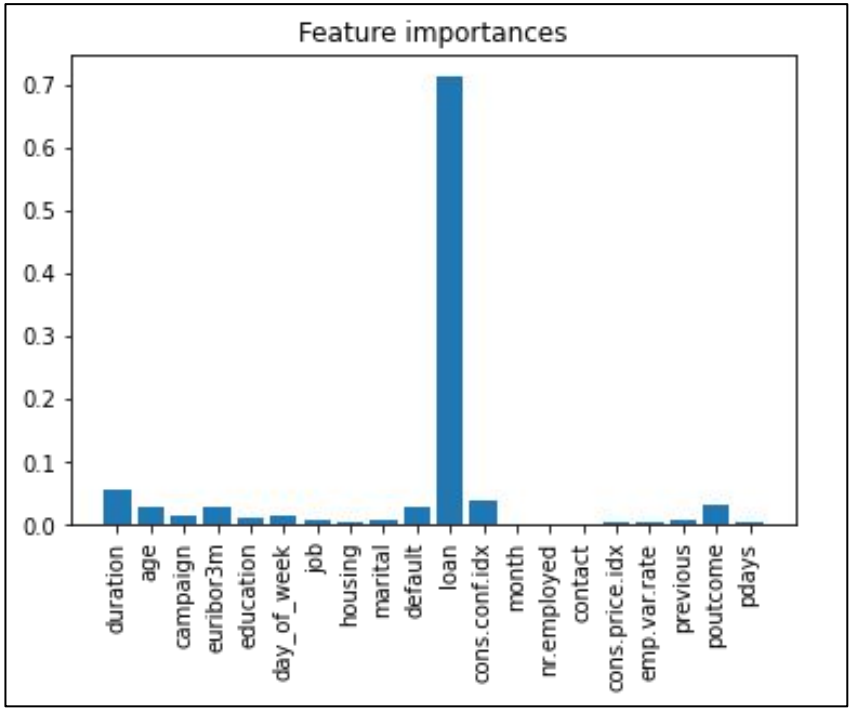
f1_score : 0.24137931034482757

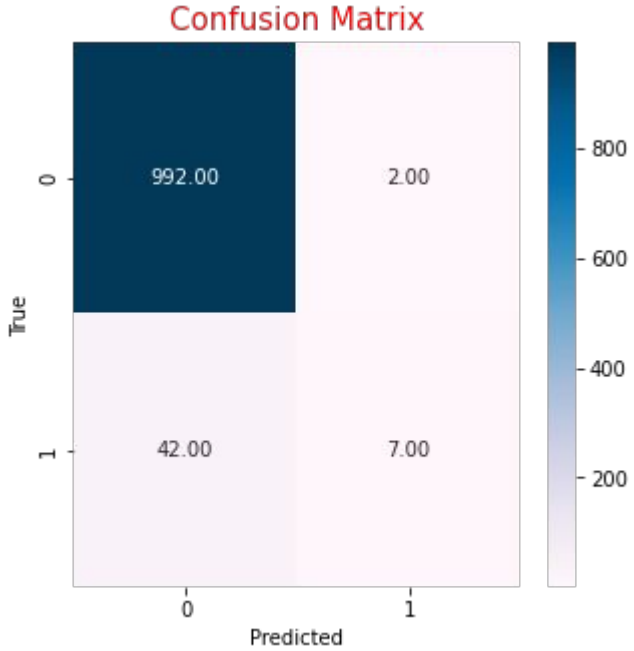
accuracy (10-fold): 0.961563

Reports

	precision	recall	f1-score	support
0	0.96	1.00	0.98	994
1	0.78	0.14	0.24	49
accuracy			0.96	1043
macro avg	0.87	0.57	0.61	1043
weighted avg	0.95	0.96	0.94	1043

FEATURE IMPORTANCE AND CONFUSION MATRIX





Feature ranking

1. feature 10 (0.712292) name: loan
2. feature 0 (0.056743) name: duration
3. feature 11 (0.037814) name: cons.conf.idx
4. feature 18 (0.030277) name: poutcome
5. feature 3 (0.029526) name: euribor3m
6. feature 9 (0.027602) name: default
7. feature 1 (0.027086) name: age
8. feature 5 (0.013801) name: day_of_week
9. feature 2 (0.013513) name: campaign
10. feature 4 (0.010851) name: education
11. feature 6 (0.007787) name: job
12. feature 17 (0.006786) name: previous
13. feature 8 (0.005955) name: marital
14. feature 19 (0.005728) name: pdays
15. feature 7 (0.005197) name: housing
16. feature 16 (0.004973) name: emp.var.rate
17. feature 15 (0.003733) name: cons.price.idx
18. feature 13 (0.000202) name: nr.employed
19. feature 14 (0.000135) name: contact
20. feature 12 (0.000000) name: month

Hyperparameter tuning

- GridSearchCV is a library function that is a member of sklearn's model_selection package.
- It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set.
- So, in the end, you can select the best parameters from the listed hyperparameters

- Random search is the best parameter search technique when there are fewer dimensions.
- While less common in machine learning practice than grid search, random search has been shown to find equal or better values than grid search within fewer function evaluations for certain types of problems.

CONCLUSION

- **RANDOM FOREST GIVES THE BEST RESULT ON THE TESTING SET**
- **KNN GIVES THE BEST RESULT ON THE TRAINING SET**
- **ACCURACY ACHIEVED WITH THE TEST SET : 94.5%**
- **THE SUBMISSION CSV FILE HAS :**
  - **NO - 32777**
  - **YES - 4241**

REFERENCES

https://help.sap.com/saphelp_nw73/helpdata/de/99/02f1afe99c46cda61f136375101e9/content.htm?no_cache=true

https://en.wikipedia.org/wiki/Random_forest

https://en.wikipedia.org/wiki/TensorFlow

https://towardsdatascience.com/a-mathematical-explanation-of-adaboost-4b0c20ce4382

https://medium.com/@senapati.dipak97/grid-search-vs-random-search-d34c92946318

https://www.thermofisher.com/blog/connectedlab/machine-learning-a-primer-to-laboratory-applications/

https://techvidvan.com/tutorials/svm-in-r/

https://scikit-learn.org/

https://www.tensorflow.org/

REFERENCES

https://github.com/pandas-profiling/pandas-profiling

https://www.knowledgehut.com/blog/data-science/bagging-and-random-forest-in-machine-learning

https://towardsdatascience.com/exploratory-data-analysis-with-pandas-profiling-de3aae2ddff3#:~:text=Pandas%20profiling%20is%20an%20open,a%20few%20lines%20of%20code.&text=In%20short%2C%20what%20pandas%20profiling,the%20distribution%20of%20each%20variable.

https://greatexpectations.io/blog/pandas-profiling-integration/

https://www.datacourses.com/pandas-1150/