MACHINE LEARNING PROJECT

HELLO!

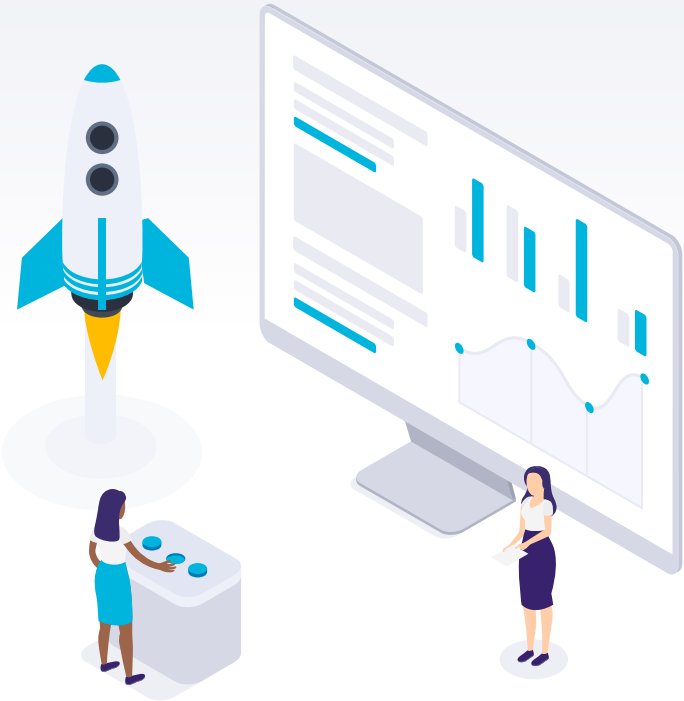
Done By: Zeinelabdin Salih

Instructor: Giti. Saikia

January 25, 2021

About Project

- Analyze and comparing models results to predict the net hourly electrical energy output (PE) of Combined Cycle Power Plant (MLRegression) and Actions for **Internet Firewall** (Classification)
- Applying machine learning technicians in both datasets to understand dataset, preprocessing, fitting and measurement prediction models using python
- Project visualize accuracy scores for models and discuss all procedures and outcomes in two main parts:
  - Multi Linear Regression Model – CCPP Dataset
  - Classification Model – Internet Firewall datasets
- Source of dataset is UCI web site



Part I: Combined Cycle Power Plant Dataset – MLR

- Dataset for CCPP data
- Contains 9568 datapoints (rows) and 4 variables (columns)
- Features consist of hourly average ambient variables:
  - Temperature (AT) in the range 1.81°C and 37.11°C
  - Ambient Pressure (AP) in the range 992.89-1033.30 milibar
  - Relative Humidity (RH) in the range 25.56% to 100.16%
  - Exhaust Vacuum (V) in teh range 25.36-81.56 cm Hg
  - Net hourly electrical energy output (PE) 420.26-495.76 MW

```
(9568, 5)
```

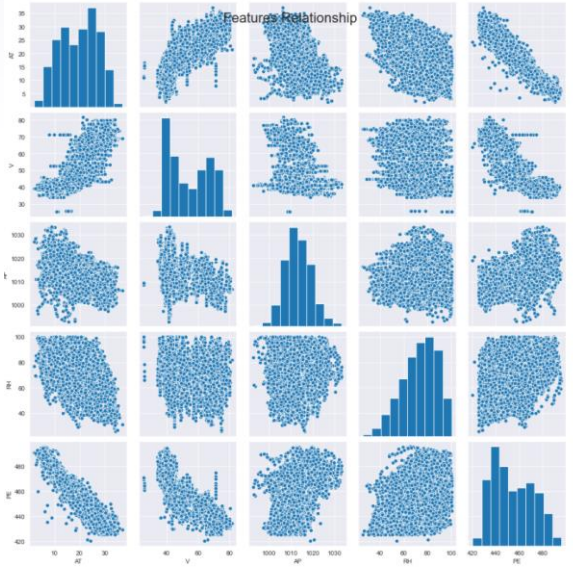
	AT	V	AP	RH	PE
0	14.96	41.76	1024.07	73.17	463.26
1	25.18	62.96	1020.04	59.08	444.37
2	5.11	39.40	1012.16	92.14	488.56
3	20.86	57.32	1010.24	76.64	446.48
4	10.82	37.50	1009.23	96.62	473.90

Analysis and Results:

- Dataset exploration and description
- No missing value detected

```
AT      0
V       0
AP      0
RH      0
PE      0
dtype: int64
```

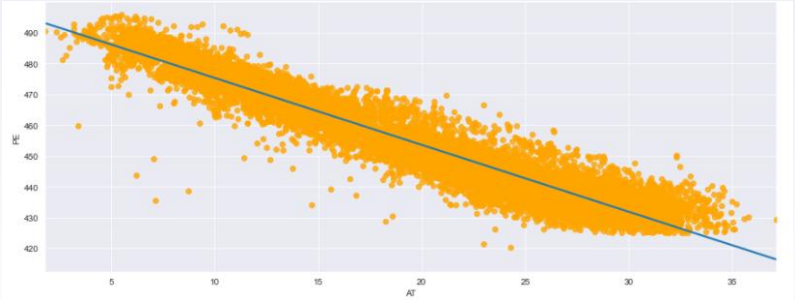
	AT	V	AP	RH	PE
count	9568.000000	9568.000000	9568.000000	9568.000000	9568.000000
mean	19.651231	54.305804	1013.259078	73.308978	454.365009
std	7.452473	12.707893	5.938784	14.600269	17.066995
min	1.810000	25.360000	992.890000	25.560000	420.260000
25%	13.510000	41.740000	1009.100000	63.327500	439.750000
50%	20.345000	52.080000	1012.940000	74.975000	451.550000
75%	25.720000	66.540000	1017.260000	84.830000	468.430000
max	37.110000	81.560000	1033.300000	100.160000	495.760000



- Visualized Features Relationship

Analysis and Results:

- The graph shows relations between attributes and target variable energy output "PE" - (Example: PE VS. AT)



- Correlation: Figured out the values and linear correlation between

	AT	V	AP	RH	PE
AT	1.000000	0.844107	-0.507549	-0.542535	-0.948128
V	0.844107	1.000000	-0.413502	-0.312187	-0.869780
AP	-0.507549	-0.413502	1.000000	0.099574	0.518429
RH	-0.542535	-0.312187	0.099574	1.000000	0.389794
PE	-0.948128	-0.869780	0.518429	0.389794	1.000000

- Visualized Correlation by Heatmap

Analysis and Results:

❖ P-values of independent variables

```
Variable Name : AP
Variable =  5.5071088524993335e-11
```

```
const     0.000000e+00
AT        0.000000e+00
V        4.375305e-215
AP        5.507109e-11
RH       3.104584e-293
dtype: float64
```

OLS Regression Results

Dep. Variable:	PE	R-squared:	0.929
Model:	OLS	Adj. R-squared:	0.929
Method:	Least Squares	F-statistic:	3.114e+04
Date:	Mon, 25 Jan 2021	Prob (F-statistic):	0.00
Time:	09:19:01	Log-Likelihood:	-28088.
No. Observations:	9568	AIC:	5.619e+04
Df Residuals:	9563	BIC:	5.622e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	454.6093	9.749	46.634	0.000	435.500	473.718
AT	-1.9775	0.015	-129.342	0.000	-2.007	-1.948
V	-0.2339	0.007	-32.122	0.000	-0.248	-0.220
AP	0.0621	0.009	6.564	0.000	0.044	0.081
RH	-0.1581	0.004	-37.918	0.000	-0.166	-0.150

❖ Find R2 and RMSE

```
R Squared(R^2):  0.9325315554761302
Mean Squared Error(MSE):  19.73369930349765
Root Mean Squared Error (RMSE):  4.442262858442491
```

❖ Adjusted R2

```
Adjusted R2:  0.9323901862890713
```

❖ Linear R Model scores

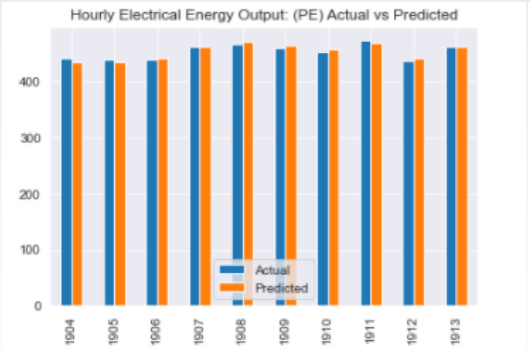
```
Model Test Accuracy Score:  0.9325315554761302
Model Train Accuracy Score:  0.9277253998587902
```

❖ K-fold cross validation for Linear Regression model

```
K-fold cross validation : 0.9286022473864548
```

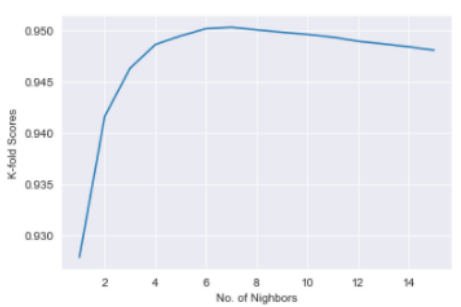
Analysis and Results:

- Prediction of Linear Regression
- Par graph visualized the tail of predicted PE compering with actual PE



	Actual	Predicted
0	431.23	431.427616
1	460.01	458.561246
2	461.14	462.752647
3	445.90	448.595962
4	451.29	457.870777
5	432.68	429.693839
6	477.50	473.041853
7	459.68	456.508363
8	477.50	474.340491
9	444.99	446.343029
10	444.37	441.939224

- KNN Model Results




```
The Optimal K : 7

[0.9278570994010462,
 0.9415964487881785,
 0.9462799950457154,
 0.9485892847930293,
 0.9494328692798186,
 0.9501293765220572,
 0.9502690089715831,
 0.9500072763083849,
 0.9497800555809164,
 0.9495737674007427,
 0.9493010348669653,
 0.9489040394780105,
 0.948644178754568,
 0.9483670422687198,
 0.948028250440295]
```

```
KNN Train Score : 1.0
KNN Test Score : 0.952585973183326
```

```
KNN K-fold cross validation: 0.948028250440295
```

Analysis and Results:

❖ Random Forest Model

```
GridSearchCV(cv=4, estimator=RandomForestRegressor(),
             param_grid={'n_estimators': [80, 100, 120, 140, 160, 180]})
```

```
{'n_estimators': 160}
```

```
Random Forest Best Score: 0.9577770879560168
Random Forest test Score: 0.9647282858380863
```

```
Random forest K-fold cross validation: 0.957838922107113
```

❖ AdaBoost Model

```
GridSearchCV(cv=4, estimator=AdaBoostRegressor(),
             param_grid={'base_estimator': [DecisionTreeRegressor(max_depth=10),
                                            DecisionTreeRegressor(max_depth=12),
                                            DecisionTreeRegressor(max_depth=14)],
                         'n_estimators': [120, 140, 160, 180, 200]})
```

```
{'base_estimator': DecisionTreeRegressor(max_depth=12), 'n_estimators': 160}
```

```
AdaBoost Best Score: 0.9604046155915316
ABaBoost test Score: 0.9683430877291839
```

```
AdaBoost K-fold cross validation: 0.9612818979351534
```

❖ SVR Model

```
GridSearchCV(cv=4, estimator=SVR(),
             param_grid={'C': [500, 1000, 1500], 'degree': [2, 3, 4],
                         'kernel': ['linear', 'poly', 'rbf', 'sigmoid']})
```

```
{'C': 1500, 'degree': 4, 'kernel': 'poly'}
```

```
SVR(C=1500, degree=4, kernel='poly')
```

```
SVR Best Score:  0.9320940716055818
SVR test Score: 0.9368176216620794
```

```
SVR K-fold cross validation: 0.9332599537923512
```

Regression Part- Conclusion

- According to all previous analysis and models scores, the best model is AdaBoost Model .
- Overall, all models are doing great and scored more than (Avg 95%), so that means the prediction here is perfect, and we can use it in the future.

	Test Score	Train Score	Cross Validation	Average
Linear Regression	0.932532	0.927725	0.928602	0.930342
KNN	0.952586	1.000000	0.948028	0.975087
Random Forest	0.964979	0.957711	0.961044	0.962171
AdaBoost	0.968453	0.960198	0.961981	0.964761
SVR	0.936818	0.932094	0.933260	0.934742

```
Model Score Maximum Average : KNN  =  0.9750872987216542

The Model Who Scored the Maximum:

Test Score            AdaBoost
Train Score              KNN
Cross Validation      AdaBoost

Models Average Score: 0.953420627262316
```

Part II: Internet Firewall–Classification

- Dataset for Internet Firewall data
- Contains 65532 datapoints (rows) and 12 variables (columns)
- Dataset Attributes :
  - Source Port
  - Destination Port
  - NAT Source Port
  - NAT Destination Port
  - Action
  - Bytes,
  - Bytes Sent,
  - Bytes Received
  - Packets
  - Elapsed Time (sec)
  - pkts_sent,
  - pkts_received

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Action	Bytes	Bytes Sent	Bytes Received	Packets	Elapsed Time (sec)	pkts_sent	pkts_received
0	57222	53	54587	53	allow	177	94	83	2	30	1	1
1	56258	3389	56258	3389	allow	4768	1600	3168	19	17	10	9
2	6881	50321	43265	50321	allow	238	118	120	2	1199	1	1
3	50553	3389	50553	3389	allow	3327	1438	1889	15	17	8	7
4	50002	443	45848	443	allow	25358	6778	18580	31	16	13	18

Analysis and Results:

- Dataset exploration and description

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Action	Bytes	Bytes Sent	Bytes Received	Packets	Elapsed Time (sec)	pkts
count	65532.000000	65532.000000	65532.000000	65532.000000	65532	6.553200e+04	6.553200e+04	6.553200e+04	6.553200e+04	65532.000000	65532.0
unique	NaN	NaN	NaN	NaN	4	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	allow	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	37640	NaN	NaN	NaN	NaN	NaN	
mean	49391.969343	10577.385812	19282.972761	2671.049930	NaN	9.712395e+04	2.238580e+04	7.473815e+04	1.028660e+02	65.833577	41.3
std	15255.712537	18466.027039	21970.689669	9739.162278	NaN	5.618439e+06	3.828139e+06	2.463208e+06	5.133002e+03	302.461762	3218.8
min	0.000000	0.000000	0.000000	0.000000	NaN	6.000000e+01	6.000000e+01	0.000000e+00	1.000000e+00	0.000000	1.0
25%	49183.000000	80.000000	0.000000	0.000000	NaN	6.600000e+01	6.600000e+01	0.000000e+00	1.000000e+00	0.000000	1.0
50%	53776.500000	445.000000	8820.500000	53.000000	NaN	1.680000e+02	9.000000e+01	7.900000e+01	2.000000e+00	15.000000	1.0
75%	58638.000000	15000.000000	38366.250000	443.000000	NaN	7.522500e+02	2.100000e+02	4.490000e+02	6.000000e+00	30.000000	3.0
max	65534.000000	65535.000000	65535.000000	65535.000000	NaN	1.269359e+09	9.484772e+08	3.208818e+08	1.036116e+06	10824.000000	747520.0

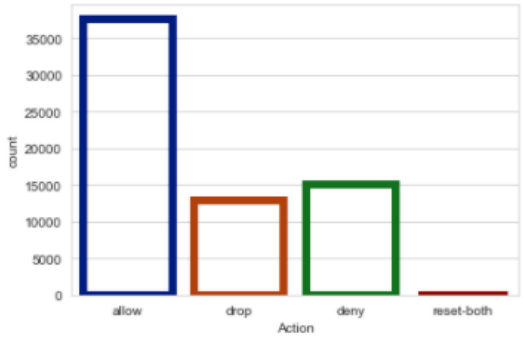
```
array(['allow', 'drop', 'deny', 'reset-both'], dtype=object)
```

- No missing value detected

```
Source Port             0
Destination Port        0
NAT Source Port         0
NAT Destination Port    0
Action                  0
Bytes                   0
Bytes Sent              0
Bytes Received          0
Packets                 0
Elapsed Time (sec)      0
pkts_sent               0
pkts_received           0
dtype: int64
```

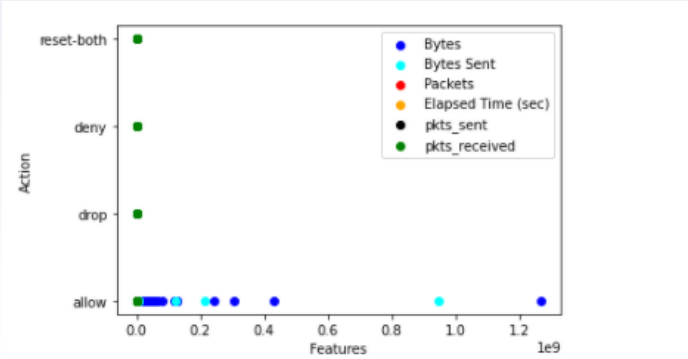
- Unique values for Target Variable “Actions”

- Visualized classes of “Actions”

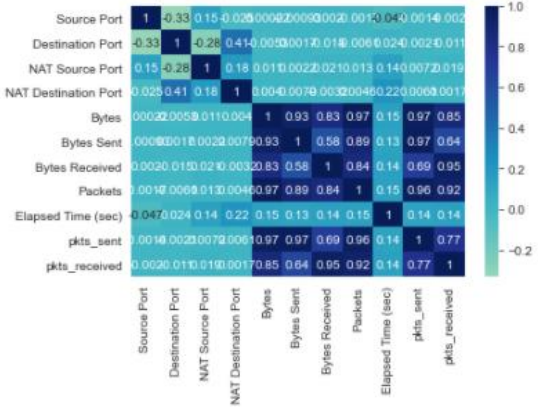


Analysis and Results:

❖ Visualised the relationship between "Action" classes and some Features



❖ Visualized dataset variables Correlation by Heatmap graph



Analysis and Results:

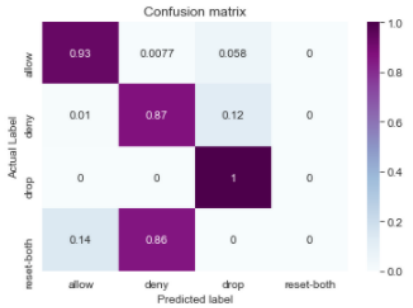
- Logistic Regions Model scores
- K-fold cross validation for Logistic Regression model
- Classification Report
- Confusion Matrix
- Heatmap visualized the Confusion Matrix

```
The Testing Accuracy is :  0.9317158770122835
The Training Accuracy is :  0.928621840724845
```

```
K-fold cross validation: 0.9294801409814402
```

	precision	recall	f1-score	support
allow	1.00	0.93	0.96	7522
deny	0.98	0.87	0.92	2989
drop	0.76	1.00	0.87	2589
reset-both	0.00	0.00	0.00	7
accuracy			0.93	13107
macro avg	0.68	0.70	0.69	13107
weighted avg	0.94	0.93	0.93	13107

```
[[7028   58  436    0]
 [  31 2595  363    0]
 [   0    0 2589    0]
 [   1    6    0    0]]
```



Analysis and Results:

- Tested Prediction of L Regression by showing the comparison between actual and predicted "Action"

	Actual	Predicted
0	allow	allow
1	drop	drop
2	allow	allow
3	deny	deny
4	drop	drop
5	allow	deny
6	deny	drop
7	allow	allow
8	drop	drop
9	deny	deny

- KNN Model Results

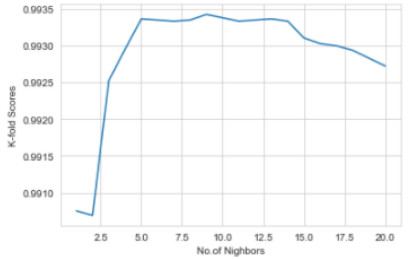
KNN Confusion Matrix

```
[[7470   46    6    0]
 [  19 2960    9    1]
 [   0    1 2588    0]
 [   2    5    0    0]]
```

KNN Classification Report

	precision	recall	f1-score	support
allow	1.00	0.99	1.00	7522
deny	0.98	0.99	0.99	2989
drop	0.99	1.00	1.00	2589
reset-both	0.00	0.00	0.00	7
accuracy			0.99	13107
macro avg	0.74	0.75	0.74	13107
weighted avg	0.99	0.99	0.99	13107

Text(0, 0.5, 'K-fold Scores')



The Optimal K : 9

```
[0.9907526094121956,
 0.9906915705304279,
 0.9925227369834585,
 0.9929500091558322,
 0.9933620216077641,
 0.9933467618873222,
 0.9933315021668803,
 0.9933467618873222,
 0.9934230604895318,
 0.9933772813282061,
 0.9933315021668804,
 0.9933467618873223,
 0.9933620216077642,
 0.9933315021668804,
 0.9931026063602515,
 0.9930263077580419,
 0.9929957883171581,
 0.9929347494353903,
 0.9928279313922969,
 0.9927211133492034]
```

KNN Train Score : 0.9997329518359561
KNN Test Score : 0.9932097352559701

KNN K-fold cross validation: 0.9927211133492034

Analysis and Results:

- Random Forest Model
- AdaBoost Model
- SVM Model

Random Forest Confusion Matrix

```
[[7519    3    0    0]
 [   0 2984    5    0]
 [   0    4 2585    0]
 [   0    5    0    2]]
```

Random Forest Classification Report

	precision	recall	f1-score	support
allow	1.00	1.00	1.00	7522
deny	1.00	1.00	1.00	2989
drop	1.00	1.00	1.00	2589
reset-both	1.00	0.29	0.44	7
accuracy			1.00	13107
macro avg	1.00	0.82	0.86	13107
weighted avg	1.00	1.00	1.00	13107

```
GridSearchCV(cv=4, estimator=RandomForestClassifier(),
             param_grid={'n_estimators': [40, 60, 80, 100, 120]})
```

```
{'n_estimators': 40}
```

```
Random Forest Best Score: 0.9978636132299701
Random Forest test Score: 0.9987029831387808
```

```
Random Forest K-fold cross validation: 0.9829243728254898
```

AdaBoost Confusion Matrix

```
[[7519    3    0    0]
 [   0 2981    5    3]
 [   0   19 2570    0]
 [   0    4    0    3]]
```

AdaBoost Classification Report

	precision	recall	f1-score	support
allow	1.00	1.00	1.00	7522
deny	0.99	1.00	0.99	2989
drop	1.00	0.99	1.00	2589
reset-both	0.50	0.43	0.46	7
accuracy			1.00	13107
macro avg	0.87	0.85	0.86	13107
weighted avg	1.00	1.00	1.00	13107

```
GridSearchCV(cv=4, estimator=AdaBoostClassifier(),
             param_grid={'base_estimator': [DecisionTreeClassifier(max_depth=1),
                                            DecisionTreeClassifier(max_depth=5),
                                            DecisionTreeClassifier(max_depth=10)],
                         'n_estimators': [20, 40, 60, 80, 100]})
```

```
{'base_estimator': DecisionTreeClassifier(max_depth=5), 'n_estimators': 100}
```

```
Adaboost Best Score: 0.9980925189332781
Adaboost test Score: 0.9974059662775616
```

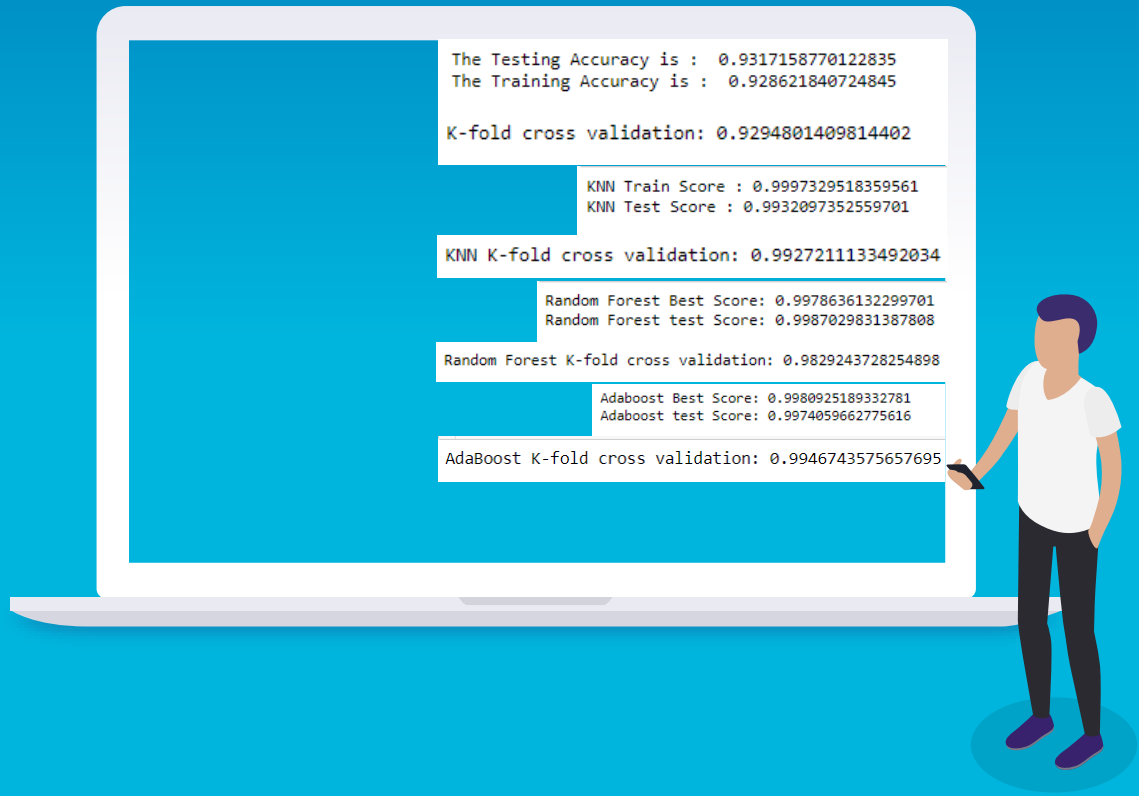
```
AdaBoost K-fold cross validation: 0.9946743575657695
```

```
SVC test Score: 0.9249785407725322
SVC test Score: 0.9268329900053407
```

```
SVC K-fold cross validation: 0.920084844045657
```

Classification Part- Conclusion

- According to all previous analysis and models scores, the best model in this part compering with others is KNN (according to current results)
- Generally, most of models are doing great and scores exceeding 98% so that means the prediction it is perfect, and also, we can use it in the future
- On the last but not least, we can say, classification model is the best vs Multi linear regression model



THANKS!

Any questions?