# Accuracy Analysis of Continuance by using Classification and Regression Algorithms in Python

Swayanshu Shanti Pragnya
Department of Computer Science and Engineering,
Centurion University of Technology and Management,
Bhubaneswar-752050
Hyderabad, India

## ABSTRACT
Reinforcement rate of technics and appositeness towards the convenience of the human being is a perennial mechanism. Mathematics has always been in the root towards the implementation of an algorithm or analysis regarding statistics or language. Extracting more about the data and analyzing them to solve a particular problem is the reason behind any analysis. Scrutiny itself has the different number of outcome which can be predictive or descriptive. Now prediction is how far accurate is tested by using various techniques. The enhancement in problem-solving capability leads to come up with a new aptitude concerning machine learning algorithms. But before prediction of data set collection, exploration, feature extraction, model building, accuracy testing are primarily required to invent. So for explaining all these processes, concept learning is essential. In this paper different algorithms like SVM, Linear and Logistic Regression, Decision tree, and Random forest algorithms will be used to demonstrate the accuracy in titanic data from Kaggle Website with all the required steps by using Python language.

## General Terms
Inferential Statistical analysis, Titanic disaster analysis, Machine learning algorithms

## Keywords
Data analysis, Machine learning, Linear regression, Logistic regression, Random-Forest, SVM, Pandas and Seaborn Library, Confusion matrix, ROC, Precision-Recall Curve

## 1. INTRODUCTION
Now a day's statistical analysis in any data is performed just to analyze the data little bit more by using mathematical terms. But only resolving a data is not sufficient when it comes to analysis that too by using statistics. So at this point, predictive audit comes which is nothing but a part of inferential statistics. Here we try to infer any outcome based on analyzing patterns from previous data to predict for the next dataset when it comes to prediction first buzzword came, i.e., machine learning. So machine learning combine's statistical analysis and computer science for the prediction purpose. Machine learning also introduced to self-learning process from particular data. This learning reduces the gap between computer and statistics. A large amount of data prediction can be possible by human interaction as a human brain can analyze the situation with various aspects. Here the partition of algorithms occur, i.e., Supervised (used for labeled data) and unsupervised (data with no tag for learning) algorithm. As the name itself says that machine will learn, but the question arises how that is by using data. In general, by performing mistakes, we learn anything so in Machine learning these mistakes are the data which will be given to the machine to learn. But only learning is not sufficient for a model as again we need to test whatever that machine learned is it accurate or not. Here accuracy testing is required which we are going to measure by creating confusion matrix.

Before building any model in machine learning first, we need to collect the data then few pre-processing is required. Feature extraction is essential to know which features are vital in our model building. After getting the features we can build our model by using different algorithms, depending on our problem statement. Once the model is built, now we need to check its accuracy. By using Logistic regression technique [9] the prediction accuracy increased to 80.756%. The actual Titanic disaster which was a ship voyage sunk in the Northern Atlantic on 15th Apr 1912 where 1502 passengers crewed out of 2224 [1]. The reason behind sinking, which data impacted more upon the analysis of survival is continuing [2], [3]. For analyzing the data set more effectively is already available in the Kaggle website [4]. Kaggle has given the platform for data analysis and machine learning [4]. The persons who are able to predict to the most accurate Kaggle provides cash prize for encouragement. [1]. Here we will know all the process carried out in model building. Different algorithms used like SVM, K- means, Decision tree, Random Forest, Linear and Logistic regression, from statistics standard deviation, variance analysis, Mean usability, displacement calculation and so on. All the concepts will execute by Python language and code will implement by using Jupyter Notebook.

## 2. THE RELEVANCE OF CLASSIFICATION AND REGRESSION
A dataset is an amalgamation of both categorical and continuous data. Dealing with such different type of data can only be possible if we can identify and segregate them effectively. Here feature engineering comes into the picture for knowing the dataset keenly.

Both classification and regression are frequently used in Data mining techniques. Regression comes into eye view when we need to predict dependent (Rely upon other attributes) variable which has relation with other data.

Example- In our given Titanic data the number of survived passenger is somehow dependent upon which class the passenger is traveling as well as which cabin they were sitting. So for predicting which person survived is relative upon all these attributes so here we will use regression technique to predict.

As the name itself defines Classification is all about the categorization of data based on condition.

Support Vector Machine algorithm can give high accuracy when the data set is small and as well as less missing values in the given dataset.

## 2.1 Programming Languages

**Pandas:** Highly used library for data analysis. Easy to understand. Open source as well as easy to use in data manipulation.

**Python:** Easy to use predefined libraries**.** Open source as well as easy to understand, the syntax is easy for beginners and used for statistical data analysis.

**Numpy:** Used for scientific computing with python.

**Seaborn:** It is used to plot different graphs for the visualization purpose.

**Matplotlib**: It is a mathematical extension from Numpy (Library for mathematical calculation) as well as primarily used for plotting graphs.

# 3. METHOD

Linear and logistic regression [3] both used for prediction purpose. But what's the difference is much more important to know. These are the following attributes to perceive the difference between these two regression algorithms.

**Outcome after regression:** In linear regression, the result we got is continuous whereas logistic regression has limited number of possible values**.**

**Dependent variable:** Logistic regression-used for the instance of true/false, yes/no, 0/1 which are categorical in nature but linear regression used in case of a continuous variable like a number, weight, height, etc.[4]

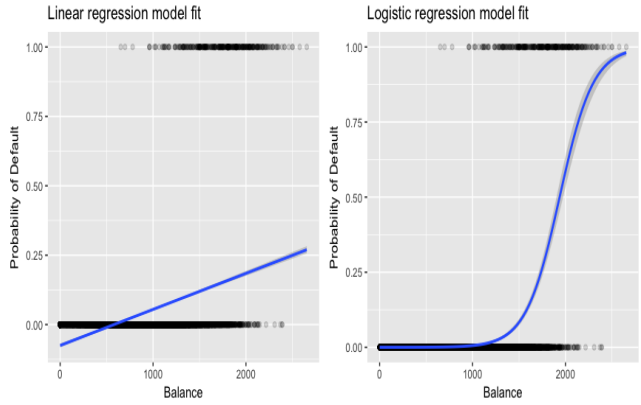


**Fig 1: Linear and Logistic Regression [14]**

## Equation

Linear regression gives a linear equation in the form of$Y = aX + C$, means degree 1 equation but, logistic regression gives curved association which is in the form of $Y = \frac{e^X}{1} + e^{-X}$ or can be simplified as $\frac{p}{1-p} = e^y$ where p is positive and less than 1.

## 3.1 Minimization of Error

Linear regression **(LR)** uses ordinary least squares method which minimizes the error and, Logistic Regression [5] use the least square method which reduces the error quadratic-ally.

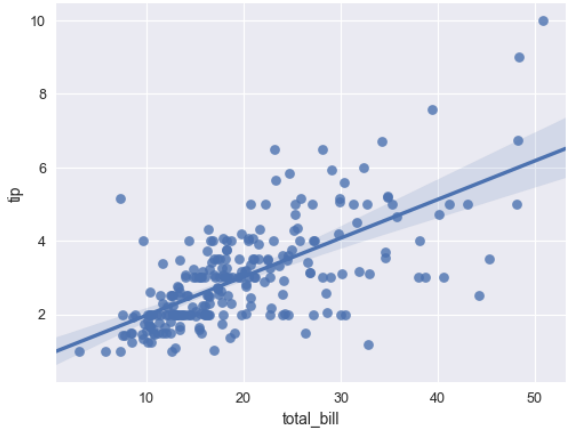


**Fig 2: Linear regression with nearest data [13]**

# 4. SUPPORT VECTOR MACHINES

These are the vectors (magnitude and direction) which take support for classification purpose near to the hyperplane. [2] Creating chasm between attributes is the process of classification where support vectors are used to classify attributes by using endorse points.

**Kernels**

Making a presage by the input as X and individual support vectors as iX can be calculated as $f(X) = b0 + sum(ia * (X, iX))$ where b0 is the coefficient and ia can be determined from the training dataset.

Hyper-plane: Generally plane forms in 2 dimensions but more than 2D it is called the hyperplane. Though support vectors drawn in more than two extent that's why it splits data through hyper-plane [2]

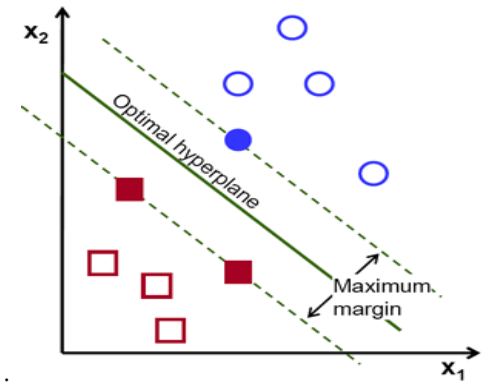


**Fig 3: SVM [12]**

In the above example we saw the set of blue and red dots separated, but in the next picture, the splitting is done via hyper-plane to segregate data set in two different clusters.

**SVM (Polynomial Kernel)**

During the evaluation of kernel using the equation leads to the dot product of the values. So for simplifying the equation we can implement polynomial kernel as $k(X, iX) = 1 + sum(X * iX)^d$ where the degree can be assigned during the learning stage and d=1(linear kernel)

**Way to find right hyper-plane**

Nearest data point and hyper-plane distance are known as margin. So when the margin is less the chance of correct segregation is more. [5]

**Decision Trees**: It is a decision sequence which designed in such a tree-like structure. It includes Yes or No type of answers. In our given data set the Passenger either survive or will die. It is built top to down from the root which involves in splitting data into instances. Entropy can be calculated as follows

$E(S) = \sum_{i=1}^{c} - pi\ log2\ pi$

**Random Forest** Tree will be the combination of the Decision tree. It can be used as both classification and regression task. The forest which is built is the ensemble of decision trees and trained by bagging method (combination of learning models)

## 5. CODE AND EXPLANATION

Step 1- Irrespective of any regression or classification algorithm initially need to import libraries like Pandas, Numpy, Matplotlib, Seaborn and from Scikit-learn linear, logistic regression and SVM module.

Step 2 – Loading data in CSV file format as the data has been taken from Kaggle Titanic competition. Where train and test dataset were grasped for regression. [10]

Step 3- Select required columns in X (mostly independent variable) and in Y take dependant column as per here number of passengers survived is dependent that's why clasped in Y.

Step 4- Data cleaning and fill null values to prepare data.

Step 5- For knowing which column is influenced (value related to other column in data) more on the output column, we need to plot graphs by using regression type. [9]

Step 6 – Split the data set into train and test by using Scikit-learn (free software for Machine learning libraries for Python programming).

Step 7- Fill all the null values using Mean or Dummy Values

Step 8- Finally call regression function whether it is linear, logistic or SVM, KNN, Decision tree. [7]

Step 9- Calculate accuracy of all the algorithms and print it.

Step 10- By importing confusion matrix calculate precision and Recall to Plot the graph.

### COMPLETE PYTHON CODE for ALGORITHMS

```
# linear algebra

import numpy as np

# data processing

import pandas as pd

# data visualization (Graphs are designed by this library)

import seaborn as sns

%matplotlib inline

from matplotlib import pyplot as plt

# Algorithms

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC, LinearSVC

test_df = pd.read_csv("test.csv")

train_df = pd.read_csv("train.csv")

train_df.describe()

#what data is actually missing (Making the dataset more convenient for prediction we need to check the null values so that we can fill up with mean or dummy values which will not affect the data loss)

total = train_df.isnull().sum().sort_values(ascending=False)

percent_1 =
train_df.isnull().sum()/train_df.isnull().count()*100

percent_2 = (round(percent_1,
1)).sort_values(ascending=False)

missing_data = pd.concat([total, percent_2], axis=1,
keys=['Total', '%'])

missing_data.head(5)

ax.legend()

_ = ax.set_title('Male')
```

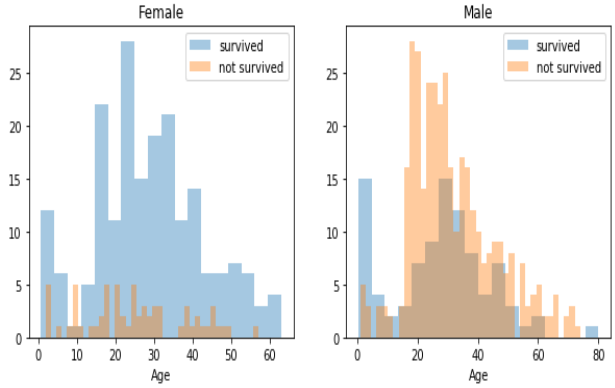


**Fig 4: Gender Wise Survival Representation**

```
#Embarked seems to be correlated with survival,

sns.barplot(x='Pclass', y='Survived', data=train_df)

for dataset in data:

    # extract titles

    dataset['Title'] = dataset.Name.str.extract(' ([A-Za-z]+)\.',
expand=False)

    dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')

    # convert titles into numbers

    dataset['Title'] = dataset['Title'].map(titles)

    # filling NaN with 0, to get safe

    dataset['Title'] = dataset['Title'].fillna(0)

# Let's take a last look at the training set, before we start
training the models.

train_df.head(5)
```

## Building Machine Learning Models

```
X_train = train_df.drop("Survived", axis=1)

Y_train = train_df["Survived"]

X_test  = test_df.drop("PassengerId", axis=1).copy()

# Random Forest
```

```
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, Y_train)
Y_prediction = random_forest.predict(X_test)
random_forest.score(X_train, Y_train)
acc_random_forest = round(random_forest.score(X_train, Y_train) * 100, 2)
print(round(acc_random_forest,2,), "%")
```

92.82 %

```
# Logistic Regression
logreg = LogisticRegression()
logreg.fit(X_train, Y_train)
Y_pred = logreg.predict(X_test)
acc_log = round(logreg.score(X_train, Y_train) * 100, 2)
print(round(acc_log,2,), "%")
```

82.04 %

```
# KNN
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)
acc_knn = round(knn.score(X_train, Y_train) * 100, 2)
print(round(acc_knn,2,), "%")
```

85.75 %

```
# Decision Tree
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, Y_train)
Y_pred = decision_tree.predict(X_test)
acc_decision_tree = round(decision_tree.score(X_train, Y_train) * 100, 2)
print(round(acc_decision_tree,2,), "%")
```

92.82 %

```
results = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression',
              'Random Forest', 'Naive Bayes',
              'Decision Tree'],
    'Score': [acc_linear_svc, acc_knn, acc_log,
              acc_random_forest, acc_gaussian, acc_perceptron,
              acc_sgd, acc_decision_tree]})
result_df = results.sort_values(by='Score', ascending=False)
result_df = result_df.set_index('Score')
```

O/P-
Model Score:

Random Forest 92.82

Decision Tree 92.82

KNN85.75

Logistic Regression82.04

Support Vector Machines77.89

**Confusion Matrix**

```
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
predictions = cross_val_predict(random_forest, X_train, Y_train, cv=3)
confusion_matrix(Y_train, predictions)
```

O/P-

```
array([[490,  59],
       [ 87, 255]])
print("Precision:", precision_score(Y_train, predictions))
print("Recall:",recall_score(Y_train, predictions))
```

Precision: 0.812101910828

Recall: 0.745614035088

## Precision Recall Curve

```
from sklearn.metrics import precision_recall_curve
# getting the probabilities of our predictions
y_scores = random_forest.predict_proba(X_train)
```

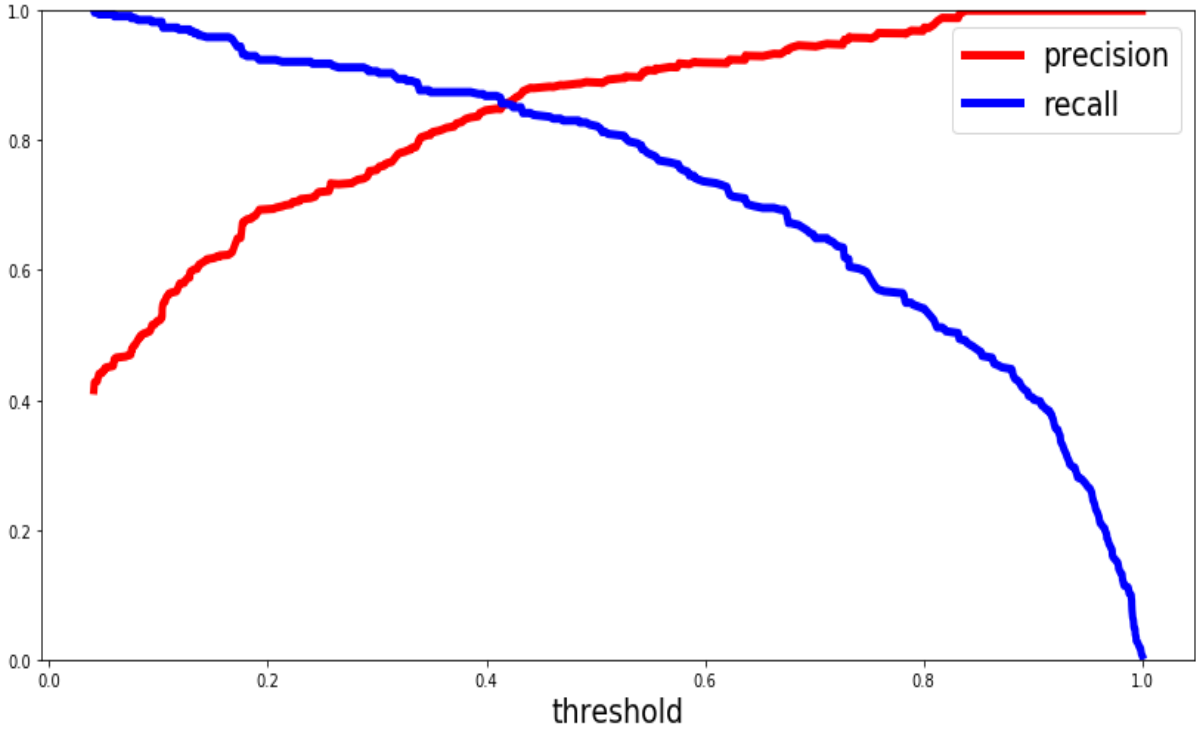


**Fig 5: Precision and Recall Graph**

The above graph clearly shows that the recall value is falling at the precision value nearly 86% and got the threshold value as 0.4. So for the desired accuracy we can train our data with the threshold exactly at 0.4.

```
def plot_precision_vs_recall(precision, recall):

plt.ylabel("recall", fontsize=19)

plot_precision_vs_recall(precision, recall

plt.show()
```

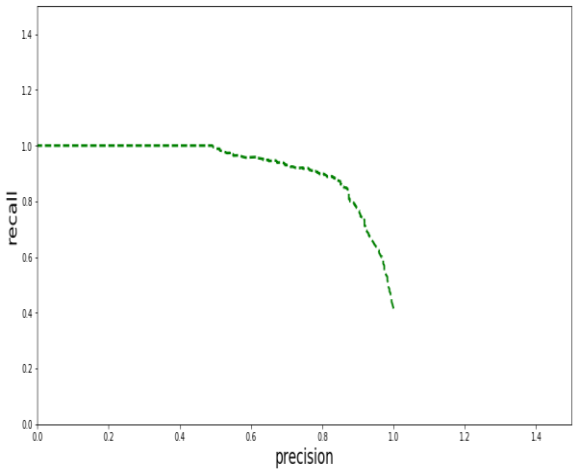


**Fig 6: ROC**

**ROC AUC Score**

```
from sklearn.metrics import roc_auc_score

r_a_score = roc_auc_score(Y_train, y_scores)

print("ROC-AUC-Score:", r_a_score)
```

**O/P: ROC-AUC-Score: 0.944713407**

## RESULT ANALYSIS

All the model which were executed in pandas, accuracy is evaluated by using confusion matrix, ROC and ROC-AUC-Score.

By using the above code, we have already calculated the accuracy of each algorithm. Now by using confusion matrix, we will reckon how many numbers are correct.

| n=192 | **Predicted: 0** | **Predicted: 1** | |
|---|---|---|---|
| **Actual: 0** | TN = 118 | FP = 12 | 130 |
| **Actual: 1** | FN = 47 | TP = 15 | 62 |
| | 165 | 27 | |

**Fig 7: Confusion Matrix Example**

**Confusion matrix terminology**

**True positive (**Positively predicted values TP**), False negative (**Opposite of TP**)**

Predicted i.e. Precision will be = TP / (TP + FP) which will evaluate how often predicted true values are correct. Recall will be TP / (TP + FN) which answers the negative prediction over correct prediction.

Where, TP = Total positive prediction, FP = False positive and FN = False negative.

As per our result, we got Precision as 0.812101910828 and Recall as 0.745614035088. So our models have predicted

81% accurately. From the results, we got both random forest, and the decision tree is giving high accuracy.

**Receiver Operating Characteristic (ROC)**
This is a graph plots in between true positive rate against false positive rate. The more closely the graph will be towards the axis the more is the predicted accuracy.

**ROC-AUC-Score** It is the area under ROC curve. The classifier is 100% correct have the score as 1 and random classifier would be having the score as 0.5.

In our given dataset we got ROC-AUC-Score as 0.944713407 which is nearly equal to 1 by 0.05 valueless. It means the models have given nearly 100% correct score.

**Table 1: Algorithm and Percentage of accuracy**

| **Algorithms name** | **Accuracy in %** |
|---|---|
| Random Forest | 92.82 |
| Decision Tree | 92.82 |
| KNN | 85.75 |
| Logistic Regression | 82.04 |
| Support Vector Machines | 77.89 |

# 6. CONCLUSION

Here we have studied the basic about machine learning, linear regression, logistic regression, SVM, KNN, Decision tree and Random forest tree algorithm. We have executed the code by using python language and got the output successfully by using Confusion matrix, Precision-recall curve and ROC-AUC-Score. In the end, we have calculated Random forest, and decision tree model are giving a higher accuracy of 92.82 % of data by using modules from sci-kit learn. From ROC-AUC-Score we got the score as 0.9447 which is nearly giving cent percent correct prediction for the chosen model. The objective was for knowing all these five algorithms and code execution which is computed with accuracy. We have also performed confusion matrix, Area under curve score for result analysis and got the result by getting the Precision and Recall value. For the future work I will work on the significance of feature engineering while building a model using machine learning algorithms.

# 7. ACKNOWLEDGMENTS

The author is thankful to the reviewers for their scarce suggestions which augmented the constitution of the paper.

# 8. REFERENCES

[1] THE TRAGEDY OF TITANIC: A LOGISTIC REGRESSION ANALYSIS. Dina Ahmed Mohamed Ghandour1 and May Alawi Mohamed Abdalla2.

[2] A Comparative Analysis on Linear Regression and Support Vector Regression Kavitha S Assistant Professor Computer Science and Engineering Bannari Amman Institute of Technolgy Sathyamangalamkvth.sgm@gmail.com

[3] An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain Park, Hyeon-Ae College of Nursing and System Biomedical Informatics National Core Research Center, Seoul National University, Seoul, Korea

[4] Bagley, S. C., White, H., & Golomb, B. A. (2001). Logistic regression in the medical literature: Standards for use and reporting, with particular attention to one medical domain. Journal of Clinical Epidemiology, 54(10), 979-985. Bewick, V., Cheek, L., & Ball, J. (2004).

[5] Statistics review 13: Receiver operating characteristic curves. Critical Care (London, England), 8(6), 508512. http://dx.doi.org/10.1186/cc3000

[6] Austin, J. T., Yaffee, R. A., & Hinkle, D. E. (1992). Logistic regression for research in higher education. Higher Education: Handbook of Theory and Research, 8, 379–410. 2. Bagley, S. C., White, H., & Golomb, B. A. (2001).

[7] Logistic regression in the medical literature: Standards for use and reporting, with particular attention to one medical domain. Journal of Clinical Epidemiology, 54(10), 979-9853

[8] Prediction of Survivors in Titanic Dataset: A Comparative Study using Machine Learning Algorithms Tryambak Chatterjee* Department of Management Studies, NIT Trichy, Tiruchirappalli, Tamilnadu, India

[9] GE, "Flight Quest Challenge," Kaggle.com. [Online]. Available: https://www.kaggle.com/c/flight2-final. [Accessed: 2-Jun-2017].

[10] "Titanic: Machine Learning from Disaster," Kaggle.com. [Online]. Available: https://www.kaggle.com/c/titanic-gettingStarted. [Accessed: 2-Jun-2017]. [3] Wiki, "Titanic." [Online]. Available: http://en.wikipedia.org/wiki/Titanic. [Accessed: 2-Jun-2017].

[11] Kaggle, Data Science Community, [Online]. Available: http://www.kaggle.com/ [Accessed: 2-Jun-2017

[12] Svmhttps://aitrends.com/ai-insider/support-vector-machines-svm-ai-self-driving-cars/

[13] https://seaborn.pydata.org/tutorial/regression.html.

[14] https://afit-r.github.io/logistic_regression