


# Campus Placement Prediction Using Various Machine Learning Algorithms





By :

D Gokul Kanna  
St. Joseph's College, Bengaluru  
M.Sc. Statistics (20MST08)




# CONTENTS



- ✓ Study On Data
  - ✓ Variable Description
  - ✓ Objective
  - ✓ Evaluation Metrics
  - ✓ Decision On which Performance metrics to be used
  - ✓ Types Of Algorithms Used
  - ✓ Result
  - ✓ Conclusion
- 
- 

# STUDY ON DATA

The Data is a primary data, which contains information of 986 students of a University in Bengaluru (India), From the year 2019-2020, who passed within the Stipulated Time period.



	Stream	Current Degree	10th%	12th%	UG%	Backlog	Work Experience	Placement Status
0	Science	BSc	83.0	70.0	71.00	No	No	Placed
1	Arts	BA	87.0	86.0	72.00	No	No	Placed
2	Arts	BA	95.0	89.4	77.33	No	No	Placed
3	Science	BSc	79.5	93.0	66.67	No	Yes	Placed
4	Arts	BA	93.0	94.0	75.00	No	No	Placed
...	...	...	...	...	...	...	...	...
981	Professional	BVoc	72.5	61.0	70.00	Yes	Yes	Not Placed
982	Professional	BVoc	53.0	51.0	55.00	Yes	No	Not Placed
983	Commerce	BCom	83.0	84.0	65.00	Yes	No	Not Placed
984	Professional	BVoc	72.0	61.0	70.00	Yes	No	Not Placed
985	Arts	BA	63.0	55.0	75.00	No	No	Placed

# VARIABLE DESCRIPTION



- ☐ **Stream** (includes Arts, Commerce, Professional, Science)
- ☐ **Current Degree** (includes Bvc, Bvoc, Bsw, Bsc, Bcom, Bca, Bba, BA)
- ☐ **Backlogs** (If the Student has backlog or not, throughout all 6 semesters)
- ☐ **Work Experience** (If the Student has any prior work experience)
- ☐ **Placement Status** (If the Student is placed through Campus Selection or not)
- ☐ **10<sup>th</sup>, 12<sup>th</sup> & UG Marks** (in percentage)

# OBJECTIVE



The Main Objective of this project is to Predict Campus Placement using Machine Learning Algorithms and to compare their efficiencies.



# Evaluation Metrics







---

- Precision
- Recall
- F1 Score
- Accuracy

Determining which one to use is an important step in the data science process.

Confusion Matrix is extremely useful for measuring Recall, Precision, Accuracy.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN


		Actual Values	
		1	0
Predicted Values	1	<b>TRUE POSITIVE</b> 	<b>FALSE POSITIVE</b> 
	0	<b>FALSE NEGATIVE</b> 	<b>TRUE NEGATIVE</b> 

# Decision on Which Performance Metrics to be used ?

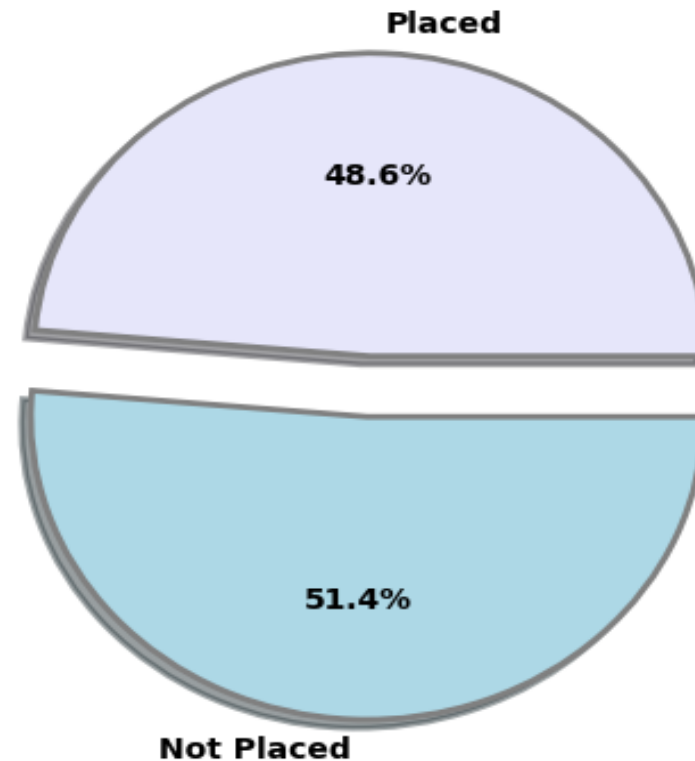


**Balanced Datasets :** If the Target variable have positive values which are approximately same as negative values. Then we can say our dataset in balanced.

**Imbalanced Datasets :** If there is the very high different between the positive values and negative values. Then we can say our dataset is Imbalanced Dataset.



The Target variable in the dataset is “ Placement Status” .



And it can be observed that it can fall under **Balanced Dataset**.

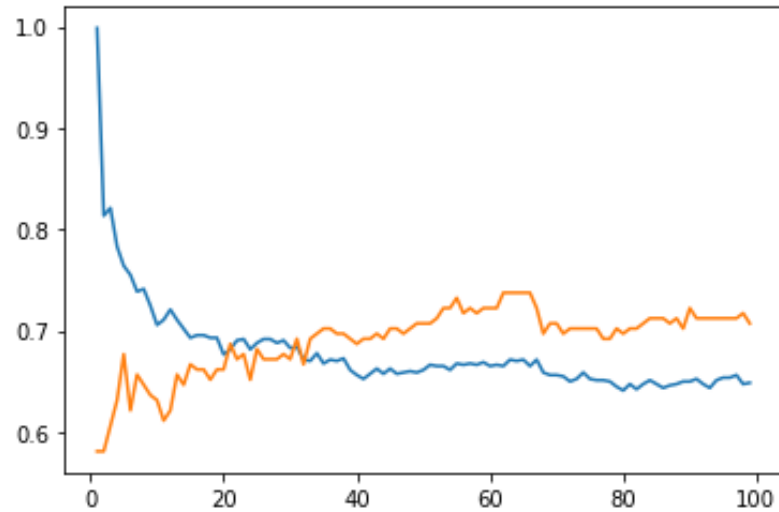
**Precision, Recall and F1 Score** has to be used in case of **Imbalanced Dataset**.

**Accuracy** is used when the Dataset is **Balanced**.



	KNN Classifier	Naïve Bayes Algorithm	Random Forest	Logistic Regression	SVM Classifier
Definition	This algorithm revolves around the concept that similar things are always in close proximity within each other.	Naive Bayes uses the Bayes' Theorem and assumes that all predictors are independent.	The <b>random forest</b> is a supervised learning algorithm that randomly creates and merges multiple decision trees into one "forest."	Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable	<b>SVM algorithm</b> is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.
Advantages	<ul style="list-style-type: none"> <li>Simple and easy-to-implement.</li> <li>Assumptions not required.</li> <li>Versatile algorithm.</li> </ul>	<ul style="list-style-type: none"> <li>This algorithm works quickly.</li> <li>If assumption holds true, it can perform better than other models &amp; requires less training data</li> </ul>	<ul style="list-style-type: none"> <li>Data with higher dimensionality can be handled.</li> <li>handles 1000's of input variables &amp; can identify most significant.</li> <li>it is a good dimensionality reduction method.</li> </ul>	<ul style="list-style-type: none"> <li>Its good for linearly separable dataset.</li> <li>It is efficient to train &amp; easy to interpret &amp; implement.</li> <li>Less prone to overfitting.</li> </ul>	<ul style="list-style-type: none"> <li>Better during clear margin of separation.</li> <li>Effective in high dimensional spaces.</li> <li>If, no of dimensions &gt; no of samples, algorithm performs better.</li> <li>Memory efficient.</li> </ul>
Disadvantages	Algorithm becomes significantly slower as the number of independent variables increases.	<ul style="list-style-type: none"> <li>Assumes that all features are independent.</li> <li>Estimations can be wrong in some cases.</li> </ul>	Suites more for classification problems rather than regression problems as it finds it harder to produce continuous values.	<ul style="list-style-type: none"> <li>Predicts discrete functions only.</li> <li>If, No. of obs in the dataset &lt; No of features, cant be used.</li> <li><b>Assumptions</b> need to be satisfied.</li> </ul>	Performance is affected when large data sets are used & when the data set has too much noise.
Outcomes	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>

## Plot for Training Accuracy vs Test Accuracy To obtain the number of neighbors (k-value)



The Following Outputs were Observed on Using KNN Classifier :

```
# display confusion matrix
print(confusion_matrix(Y_test, Y_pred))

# display accuracy
print(accuracy_score(Y_test, Y_pred))
```

```
[[76 24]
 [28 70]]
0.7373737373737373
```

Therefore the **Accuracy** of the model under KNN Algorithm is **73.73%**

[Back](#)



The Following Outputs were Observed on Using Naïve Bayes Classifier :

The Confusion Matrix under Naive Bayes Classifier is:

```
[[73 32]
```

```
 [ 4 89]]
```

Accuracy of the model under Naive Bayes Classifier is: 81.81818181818183 %

[Back](#)



	Decision Tree	Random Forest
Interpretability	Easy to interpret	Hard to interpret
Accuracy	Accuracy can vary	Highly accurate
Overfitting	Likely to overfit data	Unlikely to overfit data
Outliers	Can be highly affected by outliers	Robust against outliers
Computation	Quick to build	Slow to build (computationally intensive)

The Following Outputs were Observed on Random Forest Classifier :

```
# display confusion matrix
print(confusion_matrix(Y_test, y_pred))

# display accuracy
print(accuracy_score(Y_test, y_pred))
```

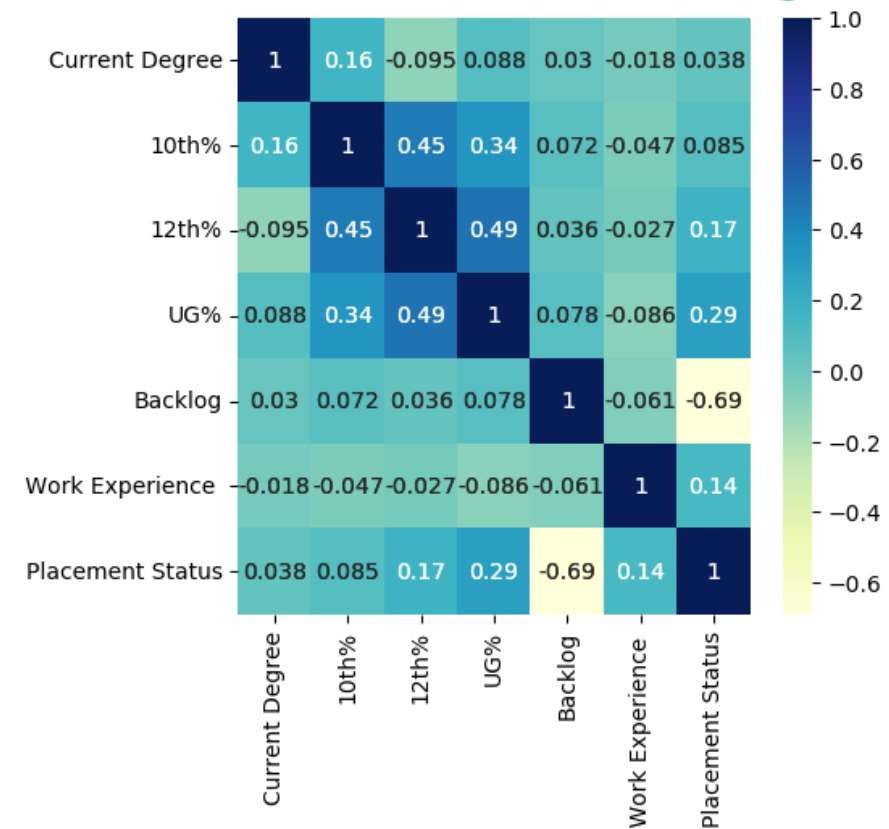
```
[[84  8]
 [18 88]]
0.8686868686868687
```

Therefore the **Accuracy** of the model  
Under Random Forest Classifier is **86.86%**

[Back](#)

# Assumptions of Logistic Regression

- First, binary logistic regression requires the dependent variable to be binary .
- Second, logistic regression requires the observations to be independent of each other.
- Third, logistic regression requires there to be little or no **multicollinearity** among the independent variables.
- Fourth, the Dataset should not contain any Outliers.
- Finally, logistic regression typically requires a large sample size.



The Following Outputs were Observed on **Logistic Regression** :

```
# display confusion matrix
print(confusion_matrix(Y_test, Y_pred))

# display accuracy
print(accuracy_score(Y_test, Y_pred))
```

```
[[93 11]
 [ 9 85]]
0.8989898989898989
```

Therefore the **Accuracy** of the model Under **Logistic Regression** is **89.89%**

[Back](#)

## The Following Outputs were Observed on SVM Classifier :

```
# Model Accuracy: how often is the classifier correct?
print("Accuracy for linear kernel:",metrics.accuracy_score(Y_test, y_pred))
print("Accuracy for polynomial kernel:",metrics.accuracy_score(Y_test, y_pred1))
print("Accuracy for rbf kernel:",metrics.accuracy_score(Y_test, y_pred2))

print(" ")
```

```
Accuracy for linear kernel: 0.9040404040404041
Accuracy for polynomial kernel: 0.6212121212121212
Accuracy for rbf kernel: 0.5555555555555556
```

Therefore the **Accuracy** of the model Under **SVM Classifier** is **90.40%**

```
# display confusion matrix
print(confusion_matrix(Y_test, y_pred))

# display accuracy
print(accuracy_score(Y_test, y_pred))
```

```
[[90  9]
 [10 89]]
0.9040404040404041
```

# RESULT

The final result of performing various machine learning algorithms are mentioned in the table. We considered KNN, Logistic Regression, Random Forest, Naïve Bayes and SVM for the analysis. We trained and predicted the placement status of students based on the same dataset and found the True Positive, False Positive, False Negative, True Negative and accuracy of each algorithm. And it is tabulated in the table.

Machine Learning Algorithms	True Positive	False Positive	False Negative	True Negative	Accuracy
SVM	90	9	10	89	90.40%
Logistic Regression	93	11	9	85	89.89%
Random Forest	84	8	18	88	86.86%
Naïve Bayes Classifier	73	32	4	89	81.81%
KNN	76	24	28	70	73.73%



# CONCLUSION

**Placement prediction system** is a system which predicts the placement status of final year students. Different machine learning algorithms are used in the python environment. We analyze the **Accuracy** of different algorithms and it is shown in the table. It is clear that **SVM** gives an **accuracy** of over **90%**. **Logistic Regression** is also good which gives an **accuracy** of **89%** based on the given dataset. The accuracy of Machine learning algorithms may differ according to the dataset. From the result from our analysis it is clear that **SVM, Logistic Regression, Random Forest** are good for binary classification problems since they all give **accuracy** of above **85%**.