**Data Exploration**

The dataset is related to banking sector. The classification goal is to predict whether the client is eligible for a **loan** or **not**. The y variable in the dataset states if that person is eligible for a loan or not. If the person is eligible for a loan, he/she is denoted as 1. If not, then 0.

**Brief of the dataset**

1. LoanID: An identification number assigned to an applicant. Unique alpha numeric value.
2. Gender: Categorical variable. Denotes the applicant’s gender. Male/Female.
3. Married: Categorical variable. Denotes the applicant’s marital status. Yes/No.
4. Dependents: Numeric value.Number of persons dependent on the applicant.
5. Education: Categorical variable. Denotes the applicant’s education. Graduate/Not Graduate.
6. Self\_Employed: Categorical variable. Denotes if the applicant is self-employed or not.
7. ApplicantIncome: Numeric Value. Income of the applicant.
8. CoApplicantIncome: Numeric Value. Income of the co applicant.
9. Loan Amount: Numeric Value. Applicant’s desired loan amount.
10. Loan\_Amount\_Term: Numeric value. Applicant’s desired time for the loan.
11. Credit\_History:
12. Property\_Area: Categorical variable. Applicant’s residence area. Rural/Urban/Semiurban
13. Loan\_Status: Categorical variable. Applicant is given a loan or not. Y/N

**Tools Used**

* Spyder IDE
* Microsoft Excel
* Jupyter Notebook

**Feature Engineering Brief**

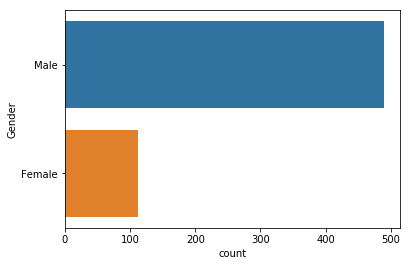
First, Duplicate data are removed. Then we removed the NA values. Removed LoanID column. Transformed Gender, Married, Education, Self-employed columns to binary variables i.e. 0/1. We converted the PropertyArea column into dummy variables and removed first column of those dummy variables to avoid dummy variable trap.

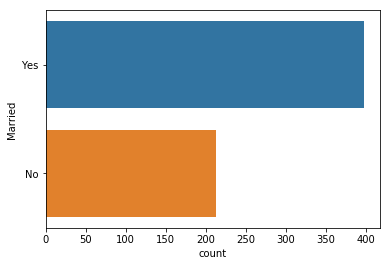
**Algorithms Used**

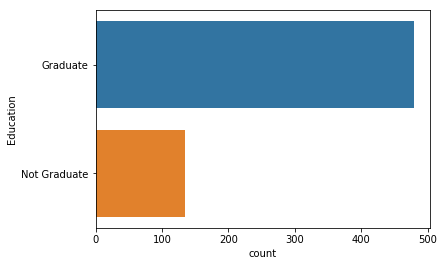
1. Standard Scaler: Used this to scale the variables in standard normal scale.
2. PCA: Used PCA to extract features from the dataset.
3. Logistic Regression: Used this binary classifier to classify if an applicant is eligible for a loan or not.
4. Decision Tree: Used this algorithm as a binary classifier to determine if an applicant is eligible for a loan or not.
5. Naïve Bayes: Used this binary classifier to classify if an applicant is eligible for a loan or not.
6. SVM: We used liner SVM, RBF kernel SVM, Poly kernel SVM to determine if an applicant is eligible for a loan or not.
7. Random Forest Classifier: Used this classifier also.
8. Grid Search CV: Used this for hyperparameter tuning.

**Metrices Used**

1. Confusion Matrix:
2. Classification Report:
3. Roc\_curve:
4. Roc\_auc\_score:

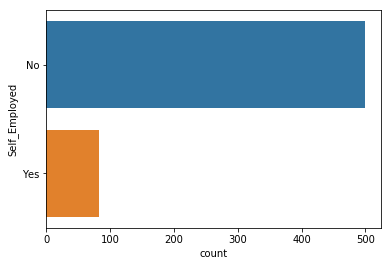
**Visualized Data**

1. Gender count:
2. Married count:

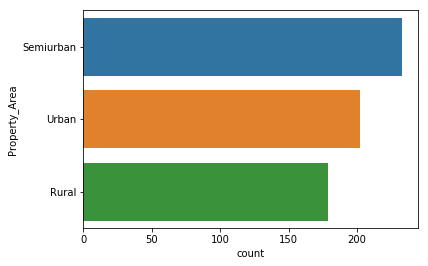


1. Education

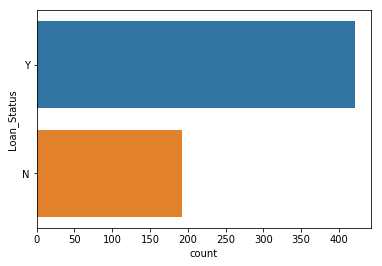
count:

1. Self\_Employed

count:

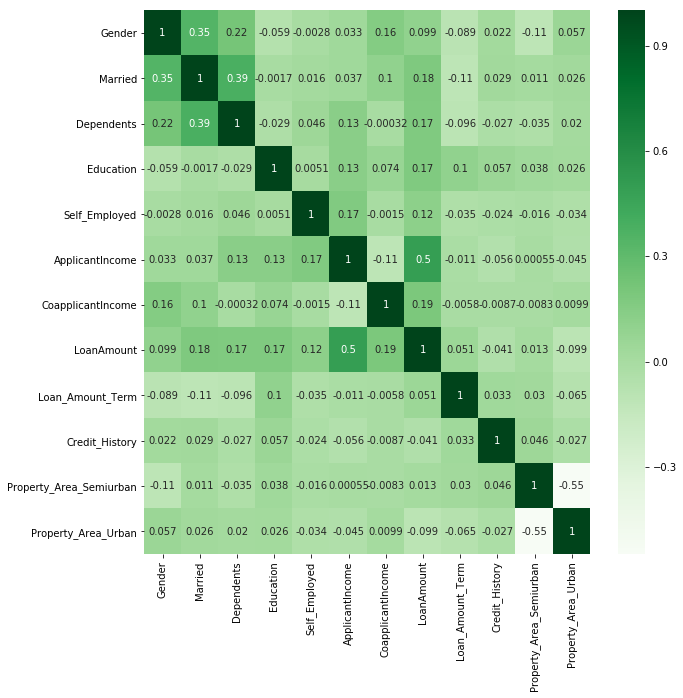
1. Property\_area

Count:

1. Loan\_status

Count:

**Correlation Heatmap between variables:**



**Process Brief**

Used all the above-mentioned algorithms one by one and got following scores.

1. Logistic Regression: First we used PCA to extract required features. Then we used Logistic Regression and got the following score:

Confusion matrix: 20 22

10 68

F1 score: 0.72

After parameter tuning with Grid Search CV:

Confusion matrix: 16 26

1. 78

F1 score: 0.75

1. Decision Tree: We don’t need to rescale the data to use decision tree classifier.

After fitting, we got the following score.

Confusion matrix: 21 21

1. 57

F1 score: 0.65

1. Random Forest Classifier: After fitting the data, we got,

Confusion matrix: 17 25

11 67

F1 score: 0.68

After hyperparameter tuning,

Confusion matrix: 17 25

1 77

F1 score: 0.75

1. Naïve Bayes: After fitting to naïve bayes,

Confusion matrix: 17 25

4 74

F1 score: 0.73

1. SVC (Linear): After fitting to linear support vector classifier,

Confusion matrix: 12 30

1. 78

F1 score: 0.70

1. SVC (Poly): After fitting to polynomial kernel support vector classifier,

Confusion matrix: 12 30

1. 78

F1 score: 0.70

1. SVC (RBF): After fitting to RBF kernel support vector classifier,

Confusion matrix: 0 42

0 78

F1 score: 0.51

**Conclusion:** After trying all the algorithms and tuning hyperparameters. We found that ***Random Forest Classifier*** produces the best result with 75 percent f1 score. So it can be concluded that random forest classifier can be used for this dataset to classify if a new applicant is eligible for a loan or not.