Proposed Approach:

**Problem statement**: A Company wants to automate the loan eligibility process based on customer details provided while filling online application form. The details filled by the customer are **Gender, Marital Status, Education, Number of Dependents, Income of self and co applicant, Required Loan Amount, Required Loan Term, Credit History** and others. The requirements are as follows:

* Loan Eligibility criteria decided by the above given variables and we have historic data related to that to train our machines

STEPS:

1. Load Datasets: (DA)\ Data Accessing

**Load Libraries**: generally, we load basic required libraries. Which are numpy for numeric operation, pandas for file reading and data manipulation, seaborn and matplotlib for data Visualization.

**Load Dataset**: We have two files one is to train and one is to predict and test the accuracy of the model. First check 18 rules of data entry are used or not. Data is clean or not. At least 30% clean data is needed to develop the good model and machine learning. Most important rule no of rows must be 20 times the number of columns.

**Problem Statement:** In my view it is a classification problem for Loan\_status Yes or No with 2 output only and whole datasets contains some numeric and some categorical data. As a data scientist Loan status is my (y) dependent/Exploratory/Class/Source/Target variable and rest of the other columns are my independent/Factors/Explained/Features variables (x1, x2…. etc.).

We have 0-9 and A-z values and train data so our solution comes under supervised machine learning.

1. **Data Summarization: (DS)**

**Descriptive stats: By** looking at the dataset we cannot understand much about the data that’s why we gave to use descriptive stats to understand the behavior of data. First to understand central tendency of data we have to look mean, median, mode of the data. And to understand the dispersion of the data we have to look **range (MIN/MAX), Variance, S.D, Skewness, Kurtosis, Coefficient of varience.** If you want to look deep down in coefficients then we have to do inferential statistics like **correlation, testing of Hypothesis, probability, Distributions… etc.**

1. **Prepared Data**:

**Data Cleaning:** Generally real time data is not clean we have to take care of that. There are lots of methods to clean the data if dataset is bigger usually anybody preferred to delete that data in time series analysis it is again complicated to delete the data. If we are not able to delete the data then we have to replace with some value like **Mean (Numerical), Median(Numerical), Mode(Categorical) or**  just use forward and backward fill **‘Ffill’ and ‘Bfill’**

We can use some advanced technic like **association, RF** to handle the missing data.

**Feature Selection:** generally, in feature selection we get useful important significant variable that are contributing more to develop the good model. We can do feature extraction like **LDA, PCA, T-SNE** in this feature selection. In feature selection we have to convert categorical data to numeric data with the help of label encoding (sklearn Library) or one hot and target, order encoding with the help of python custom programs.

**Data Transformation:** if skewness is there generally, we need to transform the data to make data looks like normal data. Some times to run algorithms and make data even we do normalization and standardization of the data.

**After doing all this thing finally we get the Model data on which we will build our Machine learning Model.**

1. **Evaluate Algorithm:**

**Split\_out\_validation\_data:**  To do cross validation we use train test split. We train the model on train data and test the model on test data. Some times we get different datasets to train and test. We use fold or random data to spilt train and testing data. Generally, 10 folds are used split the data.

**Spot checking/Compare Algorithms: As** a data scientist we know which algorithm will work best it comes with experience. still we have to run all algorithm to conform the output accuracy. If data sets are bigger definitely RF classifier CART works well but if DATA Sets are smaller, we have to look for other algorithms. Spot checking Algorithms conform the best algorithms among the tried ones. Accuracy of classification decided from AUC and ROC Curve and what we want. For classification algorithms we to check for accuracy, recall, precision for good classification.

1. **Improve Accuracy**

**Algorithm tuning:** Hyperparametric tuning and pipelines are use to increase the accuracy of the model further. In hyperparametric tuning and pipelines we generally change the parameters in the algorithms to get most possible best output. If we take the example of xgboost and decision tree we have to look for best possible **‘Learning rate’, ’Max\_depth’, ‘Gradient Descent’ etc.** GridsearchCv used in pipeline to get best possible parameter values by doing iteration over model equation.

**Ensemble:** In ensemble technique we used bagging, boosting and bootstrap methods to increase the accuracy.

1. **Finalize Model:**

**After** doing all hyperparametric tuning and ensemble technique we have to combine all the form to form best model which will work on any dataset that the model is never seen. Generalize model (Best for any Data) gives less bias and less varience and have true relationship with data. Generalize model must give the consistent prediction on any kind of Data:

**ALL STEPS IN SHORT:**

* Load important libraries and read the datasets.
* Check the shape and datatypes of given data.
* Do Descriptive stats to understand the behavior of data.
* Check the NAN/Null value and take necessary action to eliminate them.
* Prepared model data by doing transformation of categorical data to numeric data (one hot encoding/ Label Encoding).
* Find out the correlation between the variable with .corr function.
* Plot the correlation matrix to visualize the relation between variables.
* (Preprocessing)With the help of standard scaler standardize the feature variables. Which helps for further analysis in different algorithms.
* Run different model and check the accuracy of each one by comparing with test data.
* Do pipelines to increase the accuracy of the model and again check it on test data.
* Finalize one model with all tuning and ensemble to predict the values on different test.csv file.
* Predicted output write to the output.csv file.

**Still lots of things can be done exploratory analysis will take more study and time. I hope you will like my solution.**

**Thank you,**

**SWAPNIL SANJAY CHAVAN**