Qn. 5a) Data Preprocessing:

Since the given dataset is raw, there has to be lot of preprocessing done. The workflow is given in steps below:

1. As mentioned in the question, the records with "**loan_status**" values as "**Current**" are simply removed.
2. Dropped all the columns in the entire data frame in which all the values are null (i.e NaN)
3. As I'm not familiar with feature extraction, I tried to eyeball all the remaining columns and weed them out manually.
    - "**id**" and "**member_id**" are just tag number of people. So, does not contribute to any useful information. Hence, removed.
    - "**emp_title**" is also moreover like tag which consists of values such as 'PPDG', 'PPG Industries' etc. Hence, removed.
    - "**sub_grade**" is not considered for the sake of simplicity as we already consider "grade" column
    - "**issue_d**" , "**last_pymt_d**" and "**last_credit_pull_d**" are date information that doesn't contribute much to the learning. Hence, removed.
    - "**pymt_plan**" has just one unique value 'n'. Hence it is removed.
    - "**url**" contains various links which is of no use. Hence, removed.
    - "**desc**" and "**title**" consists of reasonings and general description (in large texts) why the person is taking loan. Hence, removed.
    - "**earliest_cr_line**" consists of month and year information which I felt is not of great use. Hence, removed.
    - "**zip_code**" is not considered as I considered another column which contains state information (ex: Texas, New Jersey etc.)
    - "**mths_since_last_delinq**" and "**mths_since_last_record**" columns are removed as they contain very less number of values.
    - "**collections_12_mths_ex_med**" , "**chargeoff_within_12_mths**" and "**tax_liens**" consists of all zeros and some NaN. Hence removed.
    - "**policy_code**" column consists of all "1"s. Hence, removed.
    - "**application_type**" consists of all values which are named "individual". Hence, removed.
    - "**acc_now_delinq**" , "**delinq_amnt**" ,"**out_prncp**" and "**out_prncp_inv**" columns are not considered as all values are 0.
    - "**initial_list_status**" consists of all values as 'f'. Hence, removed.
4. Now, after removing all the irrelevant columns, there are three columns namely "**emp_length**"," **revol_util**" and "**pub_rec_bankruptcies**" which have missing values. Their missing values are filled using "**Forward fill**" option of the pandas. Some more processing was done to remove certain string elements (such as "%" "<") and to convert them into 'int' or 'float' data types.

5. The columns that contain the categorical attributes include:
    - term
    - grade
    - home_ownership
    - verification_status
    - loan_status (label column)
    - purpose
    - addr_state

    All these categorical columns are converted to numerical values using "LabelEncoder" method in "sklearn.preprocessing" library.

6. Hence, after performing all the above steps, the data set is checked for any null values and then train data is finalized with 24301 rows and 31 columns.

Qn.5b) Tabulation of accuracy, precision and recall.

No. of trees (n_estimators)	Learning rate	Max features	Max depth	Accuracy	Precision	Recall
10	1	5	5	0.9932053796581676	0.9940026289845547	0.998020292006929
20	1	5	5	0.9936957130848977	0.9947372748951566	0.9978553163408397
30	1	5	5	0.9947464275707482	0.9954762296430334	0.9983502433391075
40	1	5	5	0.994536284673578	0.9950678175092479	0.9985152190051967
50	1	5	5	0.9949565704679182	0.9953144266337854	0.9987626825043306
30	0.05	5	5	0.9789857102829924	0.9758512436609514	1.0
30	0.075	5	5	0.9835388063883441	0.9809839779899661	1.0
30	0.1	5	5	0.9869010927430653	0.9850455136540962	0.9997525365008662
30	0.25	5	5	0.9934855701877277	0.9929519750860515	0.9994225851686876
30	0.5	5	5	0.9949565704679182	0.9948263118994827	0.9992576095025983
30	0.75	5	5	0.9942560941440179	0.9947407346536281	0.9985152190051967
30	1	5	5	0.9947464275707482	0.9954762296430334	0.9983502433391075

30	0.5	1	1	0.9501961333706921	0.9446002805049089	1.0
30	0.5	2	2	0.9820678061081536	0.9797849114579121	0.9995050730017322
30	0.5	3	3	0.991454188848417	0.9910794664047794	0.9989276581704198
30	0.5	4	4	0.9939058559820678	0.9934404722859954	0.9994225851686876
30	0.5	5	5	0.9949565704679182	0.9948263118994827	0.9992576095025983

Conclusion: Hence the best test accuracy is obtained when there are 30 estimators, learning rate=0.5, max_features=5 and max_depth=5 which is 0.9949565704679182.

Effect of increase in the number of trees:

In the above table we could observe that when the no. of trees (n_estimators) increases, the accuracy also increases.

Comparison with simple decision Tree:

Model	Accuracy	Precision	Recall
Gradient Boosting	0.9949565704679182	0.9948263118994827	0.9992576095025983
Decision Tree	0.9618240403474363	0.961421967160848	0.9949682421842778

Report by,

Prasanna Kumar R

sm21mtech14001