**Step by Step procedure to solve the binary classification problem**

1. Import and study the dataset:
   1. Import the dataset and observe the data points.
   2. Use scatter plots to plot the data points to observe the distribution.
   3. Understand the different features, dependent and independent features.
2. Feature Selection:
   1. Plot the correlation between all the features with the label. Use corr() function to check the correlation coefficient. Also use SelectKBest method to check which features are affecting majorly on the target variable.
   2. Only keep those K Best features in the dataset.
3. Pre-Processing:
   1. Check if the dataset has any missing/null values. If yes, remove the missing values if they are very less as compared to the total number of rows or fix them if there are too many missing values
   2. Check for outliers. If there are very few outliers, remove them using different outlier detection and removal methods like IQR technique.
   3. Check if the dataset is balanced among the 2 classes or not.
      1. For example if we have a dataset for determining whether the patient has diabetes or not. If we have 30000 rows and out of them, only 250-300 rows have “Yes” and the remaining has “No” in the label column then this is called imbalance data. There are various methods to solve such scenarios. For example, we can use SMOTE for imbalanced data.
   4. For categorical columns we need to apply label encoding or one hot encoding:
      1. We apply One-Hot Encoding when
         1. The categorical feature is not ordinal
         2. The number of categorical features is less so one-hot encoding can be effectively applied.
      2. We apply label encoding when:
         1. The categorical feature is ordinal (like Jr. kg, Sr. kg, Primary school, high school)
         2. The number of categories is quite large as one-hot encoding can lead to high memory consumption
   5. For numeric columns, we need to perform feature scaling (normalization or standardization):
      1. Based on the skewness value of the column, we select which columns we should apply normalization and standardization. There are helper functions to apply both of these scaling.
4. Split the dataset:
   1. After all the preprocessing, we should split the dataset into train and test. Generally this splitting is done in 80-20% of the data means, 80% of the data we use to train the model and 20% we use for testing.
5. Refer to the <https://docs.google.com/spreadsheets/d/1ULUiAfU3kcWUDsAptA7CfwqeZ18GDC7uMxbtq8xkkE8/edit#gid=0> sheet to compare different models and select the best suitable ML algorithm for your dataset. For binary classification, SVM, K Nearest Neighbors, Random forest are the best suitable algorithms.
6. Fit the model. Use hyperparameter tuning techniques like GridSearch, RandomSearch to select best suitable hyper parameter values (learning rate, no. of epochs, max-trial etc) for the model
7. Train the model by passing training data.
8. Once the model is trained, make the predictions using the test data and compare the predicted values with the actual values in the testing dataset.
9. Use different model evaluation metrics like r2\_score, accuracy, ROC AUC curve, RMSE etc to check the performance of the model.

Further addititons:

Evaluate generalised linear model regression results using smf library from statsmodels.formula.api :

1. Look at the (P>|Z|) column. If the p-value is less than 0.05 it means that the feature affects the target variable in a significant way.

P-value test the null hypothesis that the target variable has no correlation with the feature. If there is no correlation there is no association between the changes in the dependent variable and the independent variable.

1. Sign of coefficient indicates positive or negative correlation.

What are the most important features to train a model with high performance? This question can be answered by looking at the exponential coefficient values. The exponential coefficient estimates the expected change in target variable through a given feature by a change of one unit.

Create a baseline model with a Logistic Regression algorithm, then predict with other machine learning models like Support Vector Classifier (SVC), Random Forest Classifier, Decision-tree classifier, and Naive-Bayes Classifier.

Next, do feature selection to enable the machine learning algorithm to train faster, reduce model complexity, increase interpretability, and improve model accuracy if the right features subset is chosen. For this, use RFECV with stratified k-fold cv.

Recursive Feature Elimination (RFE) works by training the model, evaluating it, then removing the step least significant features, and repeating.

\_grid\_scores is the score the estimator produced when trained with the i-th subset of features. So, \_grid\_score[-1] will be the score of the estimator trained on all features. \_grid\_score[-2] will be the score of the estimator with step features removed. \_grid\_score[-3] will be the score of the estimator with 2\*step features removed.

As such, the grid scores do not reflect the scoring of individual features. In fact, if step is greater than 1, there will be fewer grid scores than features.

After this step, update the x with only optimal features that you got from above.

Use RandomSearch to further improve metrics