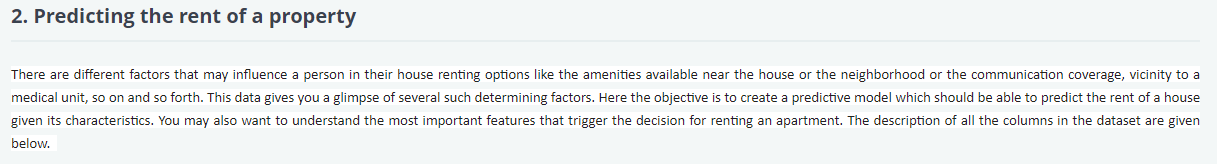
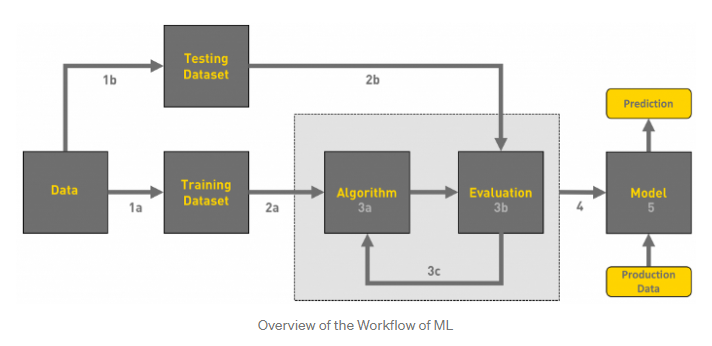
**SOLUTION FLOW DOCUMENT**

Problem Statement:



Python Libraries used to achieve the task:  
1. Numpy  
2. Pandas  
3. Sci-kit Learn  
4. Matplotlib



**Understanding the machine learning workflow:**

We can define the machine learning workflow in the below stages:

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

**Gathering data**

As per the problem statement, we were provided three datasets as train.csv, test.csv and submission.csv

**Data pre-processing**

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set.

The steps involved were:

1. Handling missing value
2. Encoding categorical data
3. Checking correlation
4. Scaling/Normalization of data

**Researching the model that will be best for the type of data**

Our main goal is to train the best performing model possible, using the pre-processed data.

The problem statement is for Supervised Learning as system is presented with data which is labelled.

Further, problem is where the target variable is **continuous**(i.e. the output is numeric), so it is a **Regression** problem.

**Training and testing the model on data**

I have divided my dataset into two parts – train and test data. I have considered test data as 30% of total data.

In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built.

Since this is a regression problem, I have used following regression algorithms:

* Linear Regression
* KNN
* Decision Tress
* Bagging algorithm like Random Forest
* Boosting algorithm like Gradient Boosting, AdaBoost

**Model Evaluation**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

To improve the model we can also tune the hyper-parameters of the model.

I have used RMSE(Root mean square error) as the evaluation metrics for the model.

As RMSE is clear by the name itself, that it is a simple square root of mean squared error(squared difference between actual and predicted value). The advantage of using this is The output value you get is in the same unit as the required output variable which makes interpretation of loss easy.

There are other metrics also like :

* Mean Absolute Error(MAE)
* Mean Squared Error(MSE)
* R squared
* Adjusted R Squared

From observation, it can be seen that *Random Forest* gave the best results for the data. So, I have chosen it as a final model.