**Model Implementation**

Since this is a prediction problem where we need to predict the rating of an application based on various factors (or predictors), a regression model would be an ideal approach.

We plan to split the data in the ratio of ***80:20*** for training and testing the model respectively. Some of the regression models that we plan to use are:

1. **Linear Regression:** Our baseline model would be linear regression. LR compares input and output variables based on linear/labeled data. Since our data are mostly numeric (we can convert categories into binary using one-hot encoding), LR models fit well. We can find the association between two variables using the correlation coefficient the value of which ranges from -1 to 1.

Some advantages of the LR model are:

* + The model is easy to use and understand, it is of low complexity
  + Training time and efficiency is great
  + If the data is truly linear, then fitting the data is perfect

Some disadvantages of the LR model are:

* + It assumes that all predictors are linear - It usually tends to under-fit the data
  + If outliers exist, it’s performance dips

**Implementation:** This is our baseline prediction model, and we are using the default parameters and settings to find the score and accuracy. Default parameters include fit\_intercept = True, normalize = False and copy\_x = True. We got the following results:

* Score – 0.016
* MSE – 0.28
* RMSE – 0.53

Plot for actual vs predicted in Linear Regression:

Chart, histogram

Description automatically generated

1. **KNN Regressor:** The K-Nearest-Neighbor algorithm predicts the value based on how close it is to the nearest neighbor values. This distance can be measured using the following methods:a. Euclidean
   1. Manhattan
   2. Statistical

The ‘K’ is the number of neighbors the training model considers. KNN generally is used for classification problems but adding a regressor to it makes it predict numerical values. The model then predicts the value based on the mean of predictor neighbor values.

Advantages of using the KNN Regressor:

* + Training time is very less
  + It can be used for both classification and regression
  + Does not require parametric assumptions

Disadvantages of using the KNN Regressor:

* + If the number of predictor variables is large, it tends to under-perform
  + Prediction will be skewed if range of one predictor dominates the other when using raw data. Standardization is recommended.

**Implementation:** We used GridSearchCV to find the best optimal parameters to fine tune this model. We also used Standard Scaler to scale the data because it is recommended for KNN to have all parameters under one scale for better scores. The optimal parameters for this model are:

* Number of Neighbors: 19
* Metric: Manhattan

We got the following results:

* Score: -0.0019
* MSE: 0.28
* RMSE: 0.53

Plot for actual vs predicted in KNN:

Chart, line chart

Description automatically generated

1. **Random Forest Regressor:** Large number of decision trees work as estimators that predict values based on rules and parameters. The outcome of all these trees is then combined and averaged to make an accurate prediction.

Advantages of RF Regressor:

* + Handles missing data automatically
  + Scaling data is not needed, it is a rule-based approach
  + Since many estimator trees make the prediction, it increases accuracy and reduces variance Disadvantages of RF Regressor:
  + Training time is generally high
  + Tends to over-fit the data

**Implementation**: We used GridSearchCV to find the best optimal parameters to fine tune this model. The optimal parameters for this model are:

* n\_estimators=44
* max\_features='auto'
* max\_depth=9
* random\_state=300

We got the following results:

* Score: 0.146
* MSE: 0.24
* RMSE: 0.49

Plot for actual vs predicted in RF:

Chart, line chart

Description automatically generated

1. **Support Vector Machine Regressor:** It sorts the data into two categories that it most likely belongs to. The SVM model then predicts which class the new data point for validation/testing belongs to based on predictors. The idea is to fit the error within a certain threshold – this approximates the best value without a given margin.

Advantages of SVM Regressor:

* + Can be used for both classification and regression
  + Robust to outliers
  + Data with high dimensions can be fitted

Disadvantages of SVM Regressor:

* + Training time is high
  + Data needs to be scaled
  + Tends to over-fit the data

**Implementation**: We used GridSearchCV to find the best optimal parameters to fine tune this model. We also used Standard Scaler to scale the data because it is recommended for SVM to have all parameters under one scale for better scores. The optimal parameters for this model are:

* 'C': 10
* 'gamma': 1
* 'kernel': 'rbf'

We got the following results:

* Score: 0.0032
* MSE: 0.28
* RMSE: 0.53

Plot for actual vs predicted in SVM:

Chart

Description automatically generated