Using K-Nearest Neighbors to Classify Lending Club Loans Issued

Matthew Delsman & Ian Poblete

OMIS 3490

March 12, 2020

For the final project, we were assigned the K Nearest Neighbors (KNN) formula. KNN is a non parametric instanced-based learning machine learning function used for classification[1]. Non parametric means that no assumptions are made about the overall shape of the data (i.e. there is no assumption that the data fits a normal bell shaped curve). Instance-based learning means that the formula is derived as new data enters the model. Practically these two attributes mean that running the formula is incredibly computationally intensive as data points are “manually” compared each time the formula is run. In addition, the formula suffers from the curse of dimensionality[2]. Essentially this means that model’s predictive power decreases as the number of evaluated features increases due to the way proximity is calculated.

From a high level view, KNN assumes that similar things exist in close proximity to each other. Diving deeper, proximity in this case is determined mathematically as the distance from point to point. This distance can be determined using a series of different mathematical functions including Euclidean, Hamming, or Manhattan, all of which are specific cases of the Minkowski distance.[3,4] Once distance is established, a generalized region is created based on areas where similar data points exist.

---

[1] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[2] https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e
[3] https://en.wikipedia.org/wiki/Minkowski_distance
[4] https://www.youtube.com/watch?v=_EEcjn0Uirw

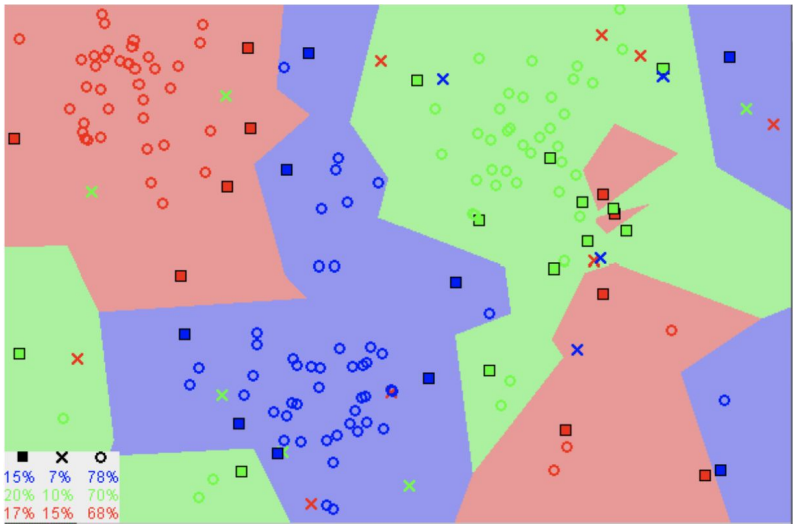

Exhibit 1: KNN Visualization with 3 Classifications[5]

Once the regions have been defined, the learning model can be used to predict the class of new data points. This prediction begins by plotting the point in the field and comparing it to its K nearest neighbors (where the formula gets its name). Essentially, a number of comparative points are chosen, the new data point is plotted, and the K amount of data points nearest to the plotted point is evaluated to determine its classification.[6] As an example, if K is set to 5 and the new point is plotted, if we find that of the 5 nearest points, 4 are classified as X, 1 is classified as square, and 0 are classified and circle. Based on this evaluation, the formula will determine that the new point is in the X category because the largest number of neighbors are classified the same way. It is important to note that the K number chosen should be an odd number in order to avoid any “ties” in cases when equal amounts of classes are obtained.

[5] https://towardsdatascience.com/knn-visualization-in-just-13-lines-of-code-32820d72c6b6

[6] https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

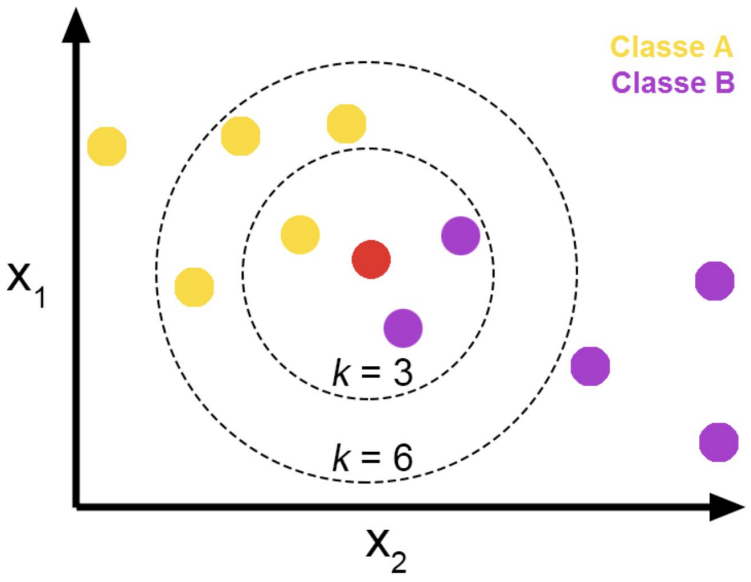

Exhibit 2: KNN Visualization with Different K Values[7]

This formula naturally lends itself to graded classification systems. When searching for data sets, we found a large data set based on loans given out from a peer-to-peer lending company called Lending Club. Due to the fact that Lending Club was peer-to-peer, a proxy score was created to help give investors a summarized score in order to evaluate the inherent risk of the loan. This proxy score would then inform the interest rate charged on the loan with riskier loans receiving a higher interest rate. Based on this data, our question became how do they determine the loan grade.

When joining Lending Club, a prospective client must fill out a series of questions about who they are and what the loan is for. In addition to this, Lending Club itself pays various agencies to get information about the prospective client such as number of credit lines, derogatory credit marks, and number of open accounts. The overall idea is that each of these data points is taken into an overall model and out comes a loan grade. This grade is incredibly important to everyone involved, with interest rate jumping up to 10% between grades[8,9]. Therefore we should expect to see significant differences

[7] https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d
[8] https://www.lendingclub.com/foliofn/rateDetail.action
[9] https://www.moneycrashers.com/lending-club-review-peer-to-peer-lending/

in the application data between grades. In actuality, we saw a very low level of difference between grades that are close, but this difference grows the further away the grades are.

Exhibit 3: Lending Club Average Interest Rates[10]

While we used KNN to solve this question, other methods could be used to answer similar questions. The most obvious that comes to mind is the linear regression model. Linear regression deals with the explanatory power of individual variables within the equation. This is in comparison to the KNN model which evaluates the overall power of the attributes. For linear regressions, the first step in this process would be to create individual subsets based on each loan grade. Then the linear regression would be run and each feature would be evaluated based on its statistical significance. This would inform which features are most important when determining the grade. Regressions would then be run on the rest of the grades using the same feature parameters. One could then compare the statistical significance of the features across grades to see how grades are affected differently by different features.

The data set we used was a csv file featuring 2.26 millions records and 145 features.[11] Because KNN is a computationally intensive method, smaller files of one

[10] https://www.moneycrashers.com/lending-club-review-peer-to-peer-lending/
[11] https://www.kaggle.com/wendykan/lending-club-loan-data

thousand, ten thousand, and one hundred thousand randomly sampled rows were created with R in order to test our code. The number of features also needed to be massively reduced. We initially used our intuition to select the following features: annual income, loan amount, total number of credit lines, loan purpose, DTI (debt-to-income ratio), expected loan default rate, home status, and credit inquiries in the last 12 months. In order to ensure that the selected features were poorly correlated with each other, a correlation heat map was created with all numeric features (Figure 1).[12] Based upon this figure, the feature expected default rate was changed to the feature derogatory public records, and the feature credit inquiries in the last 12 months was substituted for the feature credit inquiries in the last 6 months. We used the accuracy score to evaluate our model from the sklearn metrics package. This is just a simple measure of the percent of the test data points that were classified correctly.

[12] https://towardsdatascience.com/better-heatmaps-and-correlation-matrix-plots-in-python-41445d0f2bec

Figure 1: Correlation Heat Map of All Numerical Features

In order to get our code to run, some light data cleaning was performed. Rows containing NAs were removed, as were rows where annual income = 0. An initial model was created and we explored a variety of k-values (Figure 2) . Because our initial model resulted in poor accuracy scores (~28%), it was clear the data needed additional refinement.

```
Accuracy_3: 0.2660372952596991
Accuracy_4: 0.2710411315341762
Accuracy_5: 0.2757780965406812
Accuracy_6: 0.2774794008740034
Accuracy_7: 0.27928078193281514
Accuracy_8: 0.28041498482169663
Accuracy_9: 0.28171598225306066
Accuracy_10: 0.2841845414818027
Accuracy_15: 0.29088968208960203
Accuracy_20: 0.2928912165993929
```

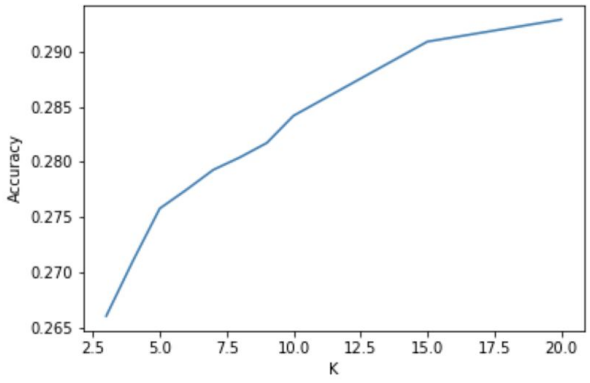

Figure 2: K Accuracy, Initial Run

To test our model a second time, dummy variables were created for the purpose and home status features, and all features were normalized to the same scale in order for each to have similar predictive power. Figure 3 shows a new correlation heatmap of the cleaned and normalized final features.

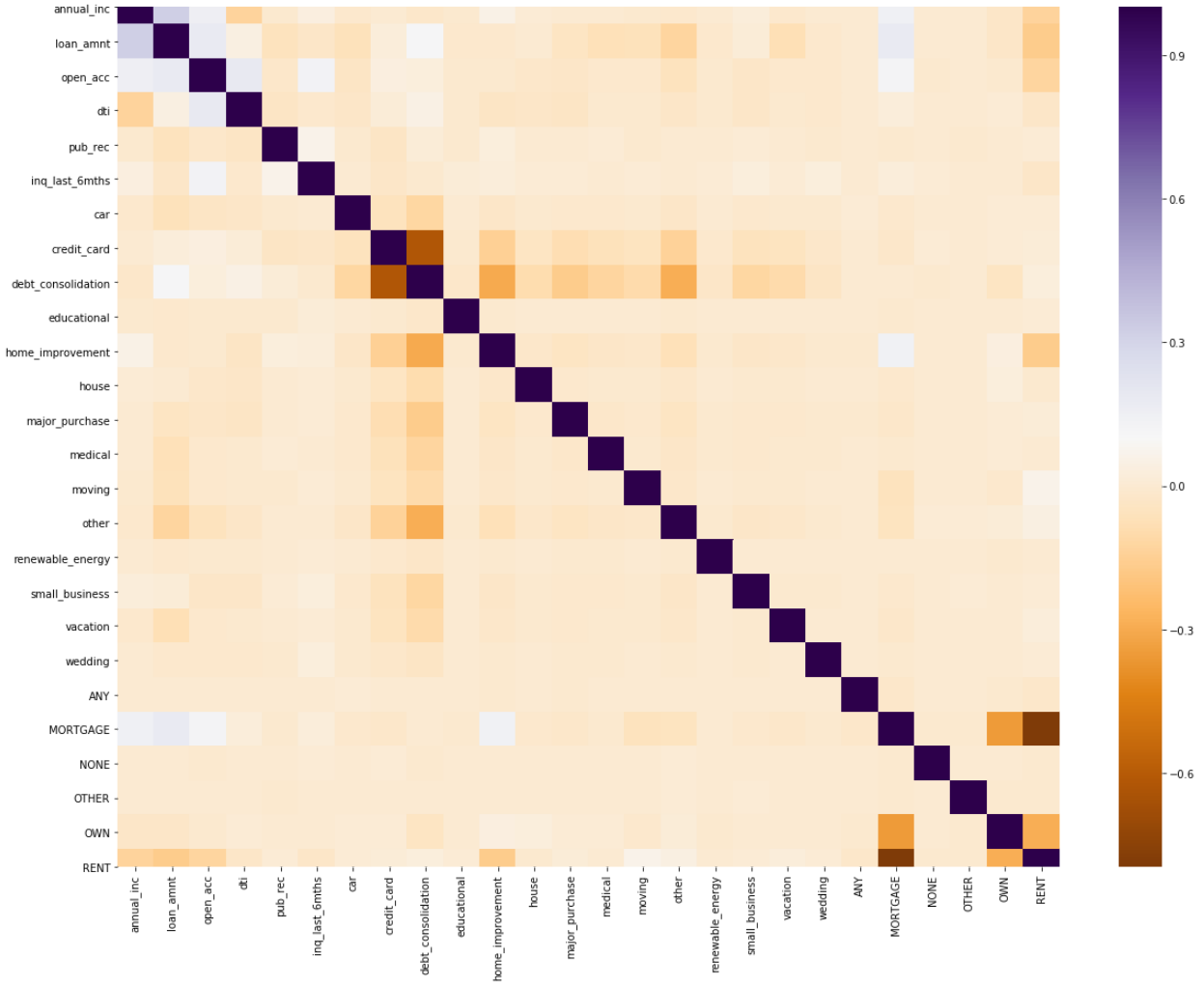

Figure 3: Correlation Heat Map of Final Features

Our model was again evaluated over a variety of different k-values (Figure 4). Accuracy had improved only by ~1% over our first run, and we decided to refine our data even further.

```
Accuracy_3: 0.27621176235113587
Accuracy_4: 0.275444507455716
Accuracy_5: 0.2852853854621877
Accuracy_6: 0.2876538679654402
Accuracy_7: 0.28972212029222405
Accuracy_8: 0.2941588551222604
Accuracy_9: 0.2946592387497081
Accuracy_10: 0.29455916202421856
Accuracy_15: 0.30239850552089936
Accuracy_20: 0.3082696734162858
```

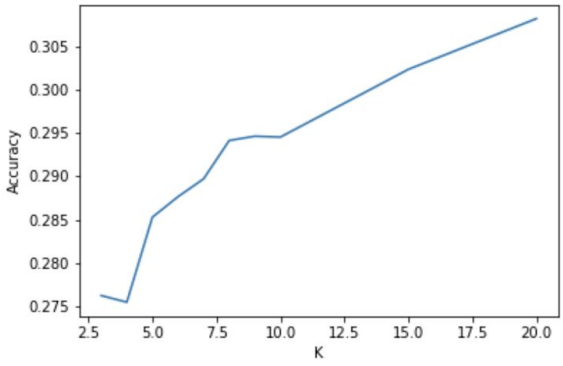

Figure 4: K Accuracy, Second Run

For our third run, outliers were removed across all numeric features, and categorical responses with low counts were consolidated. This involved deleting records with DTI > 60, annual income > $500,000, total number of credit lines > 60, derogatory public records > 5, or credit inquiries in the last 6 months > 5, and consolidating the purpose categorical responses of “house”, “car”, “vacation”, “renewable energy”, “wedding”, and “educational” with “other” and the home status categorical responses of “ANY” and “NONE” with “OTHER” (Figures 5 - 11).

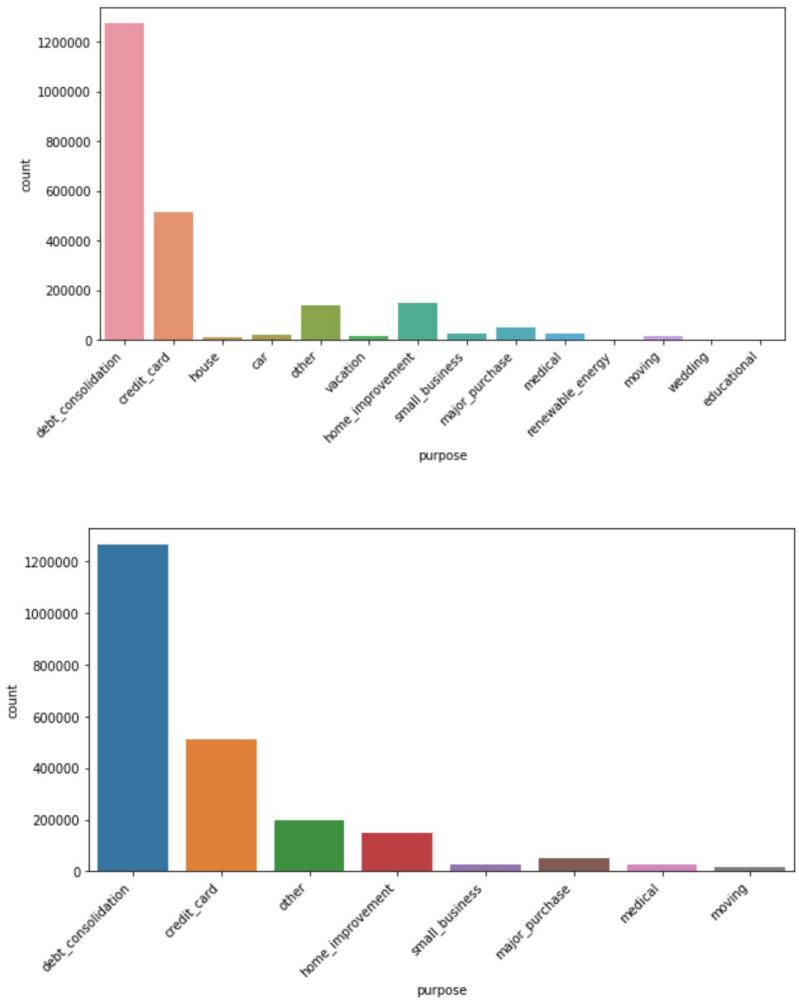

Figure 5: Loan Purpose, before and after data cleaning

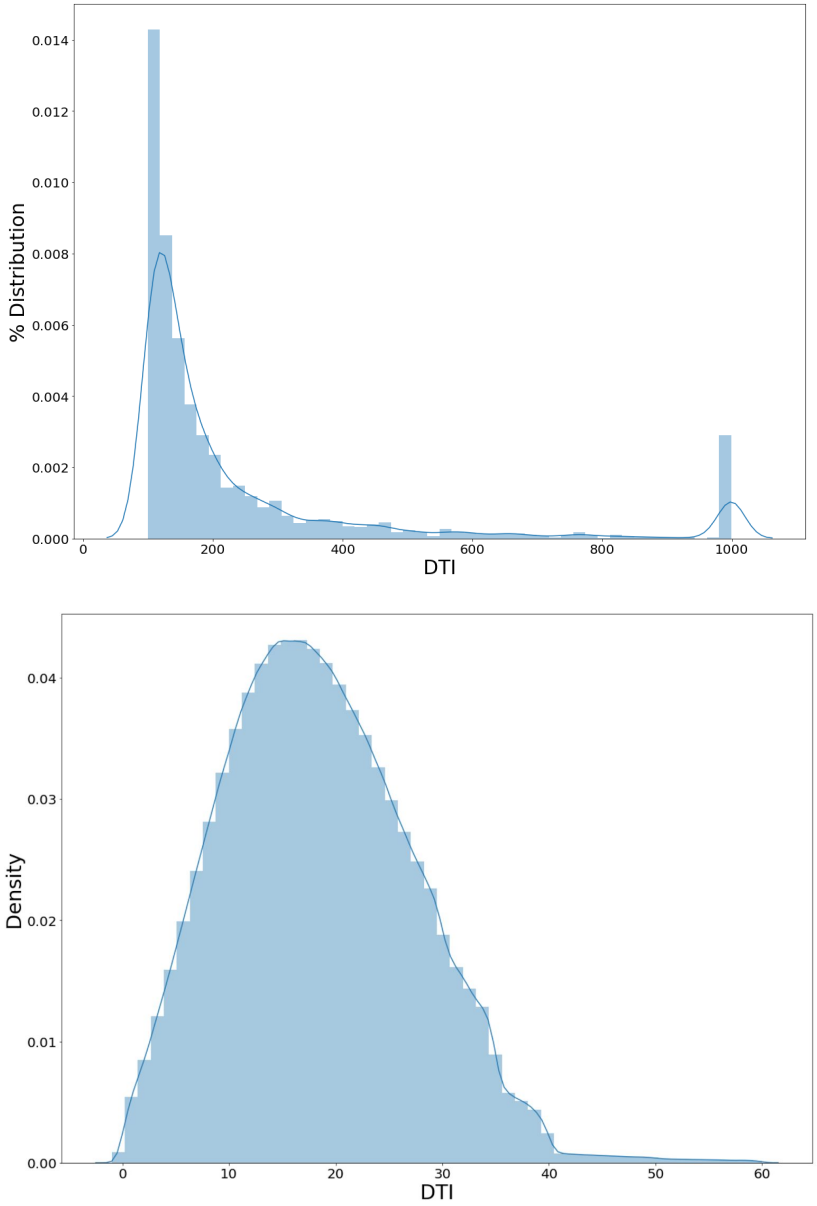

Figure 6: DTI, before and after data cleaning

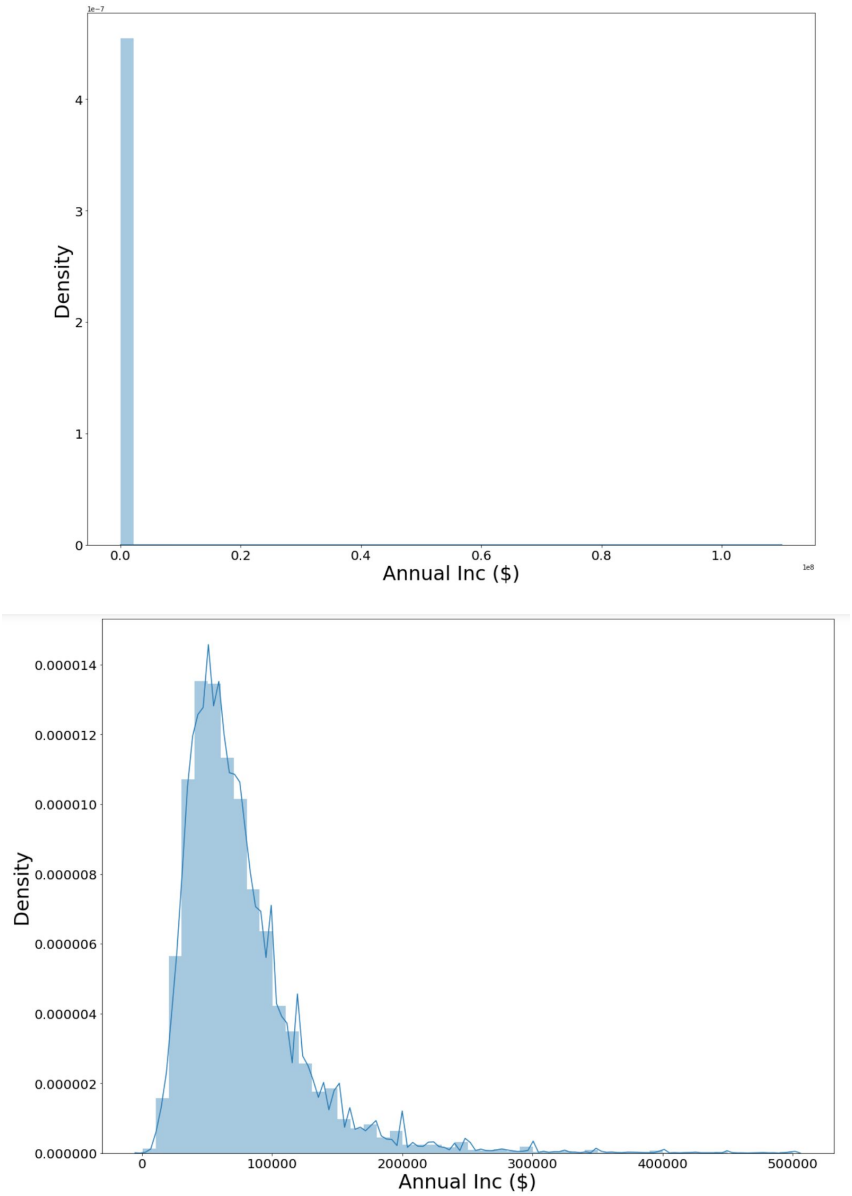

Figure 7: Annual income, before and after data cleaning

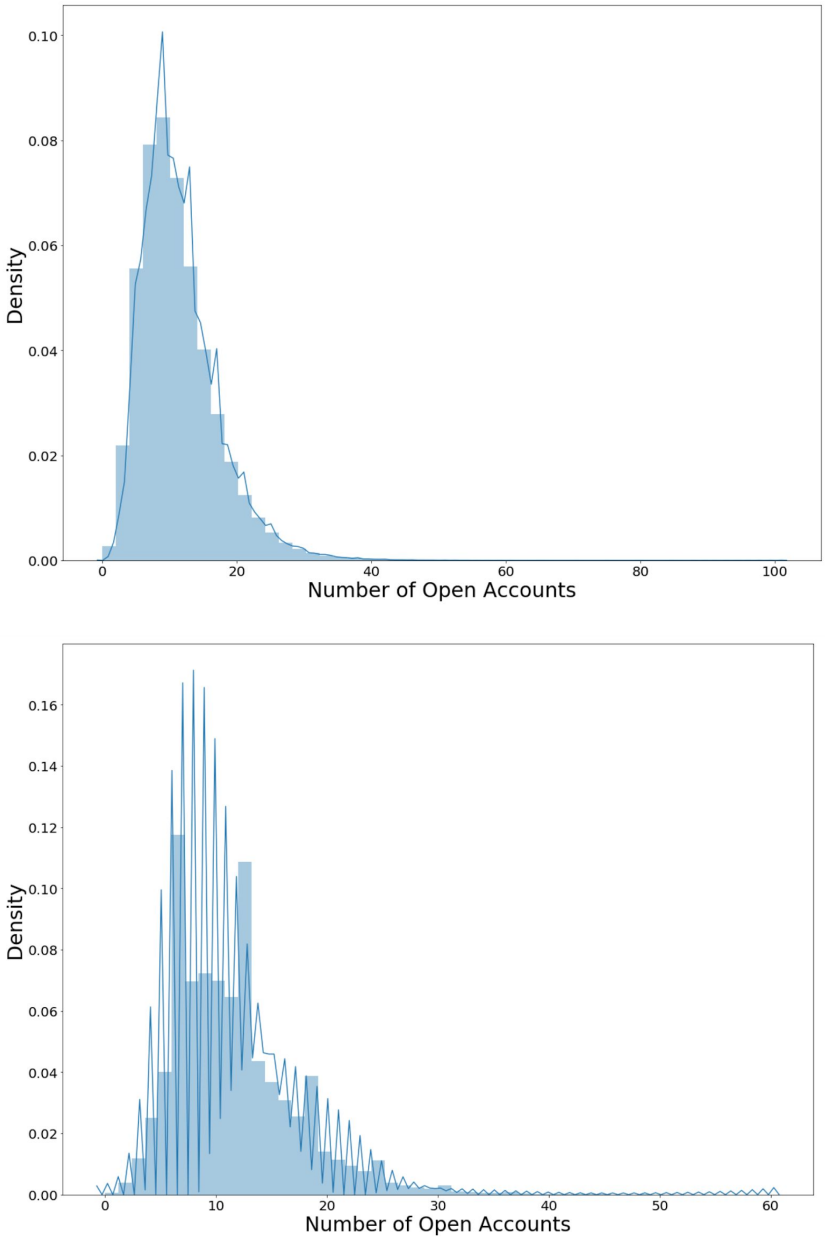

Figure 8: Total number of credit lines, before and after data cleaning

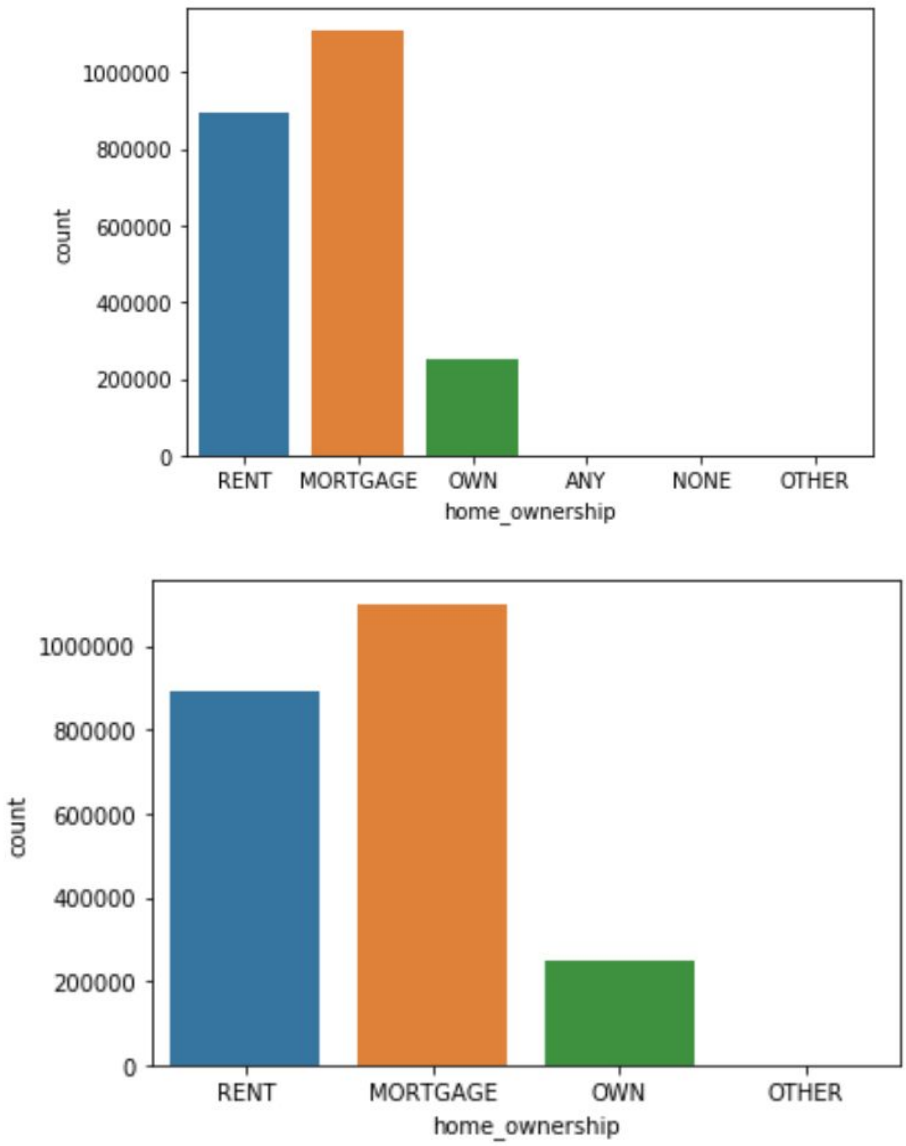

Figure 9: Home status, before and after data cleaning

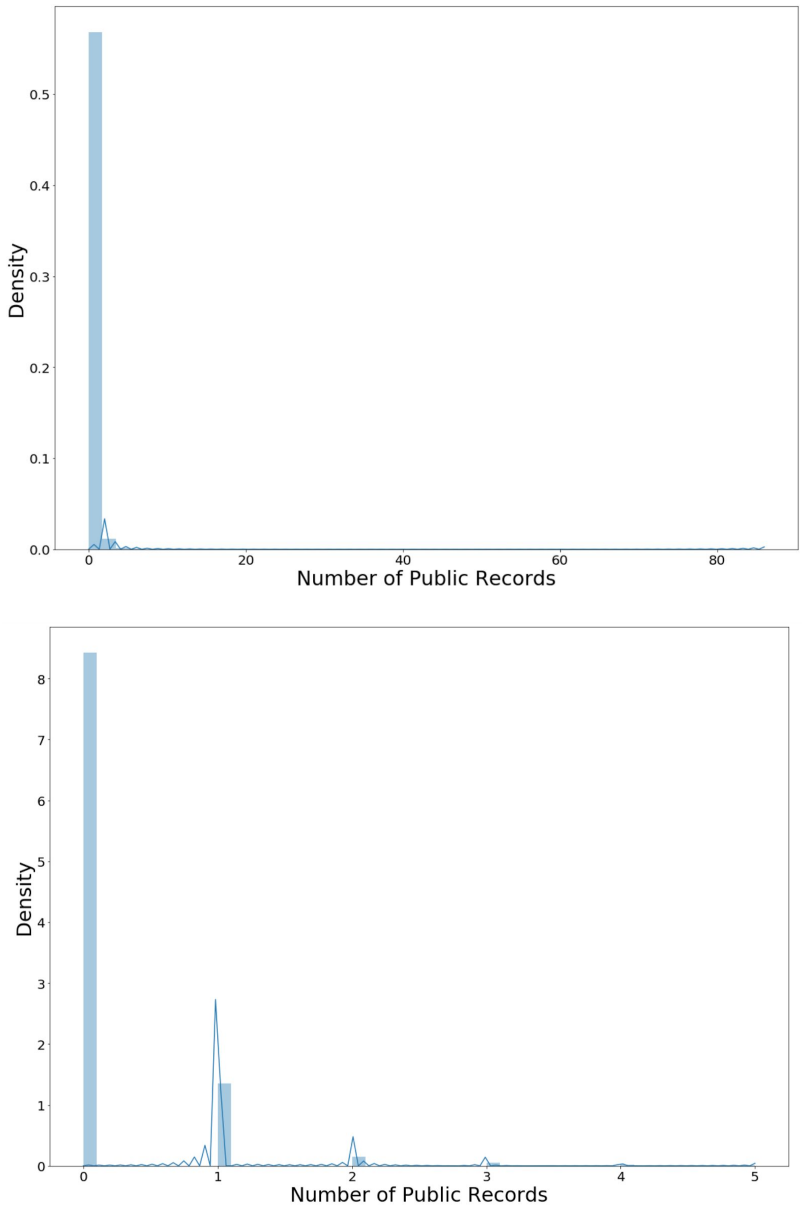

Figure 10: Total number of derogatory public records, before and after data cleaning

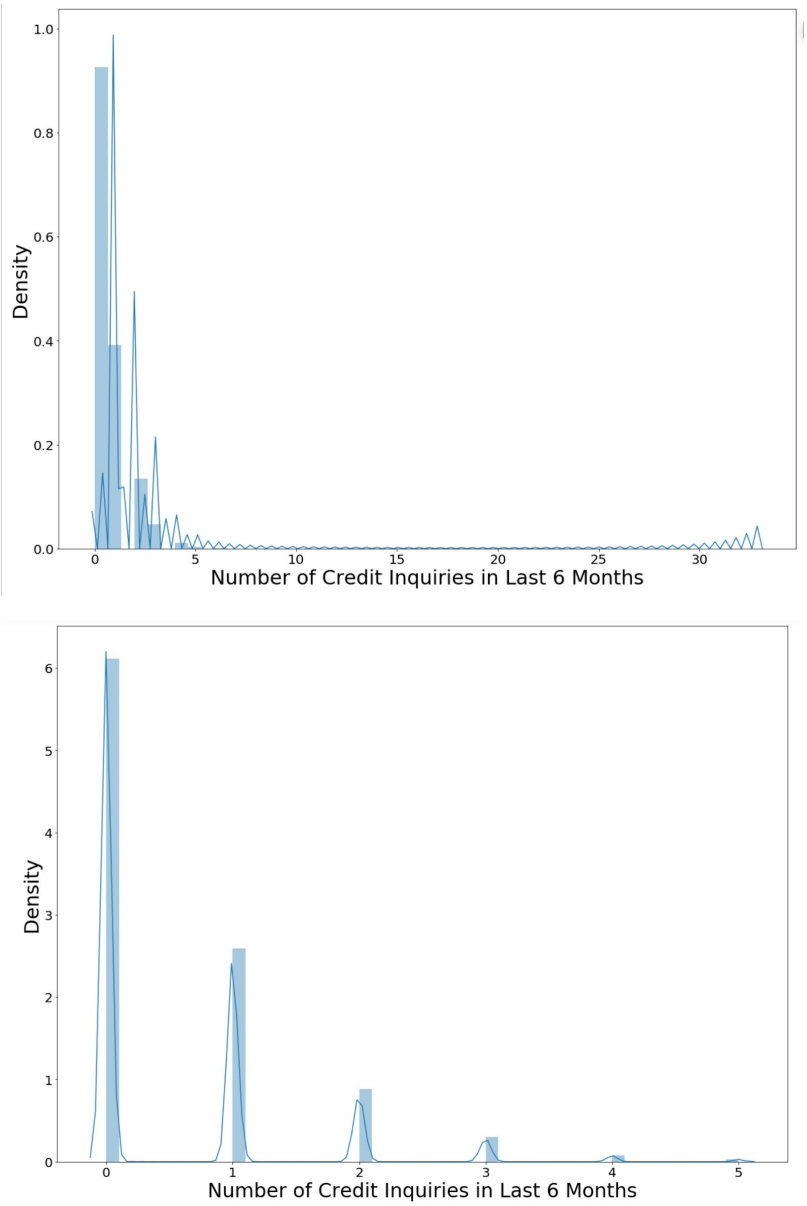

Figure 11: Total number of credit inquiries in the last 6 months, before and after data cleaning

These actions resulted in an accuracy improvement of only another ~1% over our second run over a variety of k-values (Figure 12).

```
Accuracy_3: 0.2837238971328812
Accuracy_4: 0.2856711206607131
Accuracy_5: 0.29346001477204053
Accuracy_6: 0.2962129859665615
Accuracy_7: 0.2961458403276707
Accuracy_8: 0.29920096689720005
Accuracy_9: 0.3050090646612503
Accuracy_10: 0.3064191230779561
Accuracy_15: 0.3123279393003424
Accuracy_20: 0.3166252601893507
```

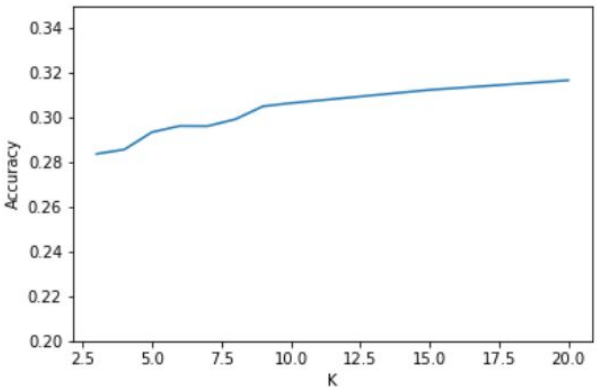

Figure 12: K Accuracy, Third Run

Determining the K value was an important step in our decision making process. The K value is important because too small of a value and not enough data will be taken into account resulting in a model that is partially fit to the present noise, and too large a K will lead to overfitting the data. Doing research online lead to a variety of different answers, with the only really solid method being using the square root of the number of observations. With the subset that we used for testing and the square root method (100,000 rows), the K would have been 315. In order to cast a wider net, we also ran K for values between 2 and 10, 15, and 20 (Figure 13). The accuracy that resulted from each of these K values was then plotted against the K value in a line graph. Despite being vastly more computationally expensive, using a K of 315 resulted in an accuracy score that was very similar to the values between 2 and 10, 15, and 20. As a result, there were diminishing returns after K = 11, and due to the computationally intensive nature of running a large K, we decided 11 to be our K.

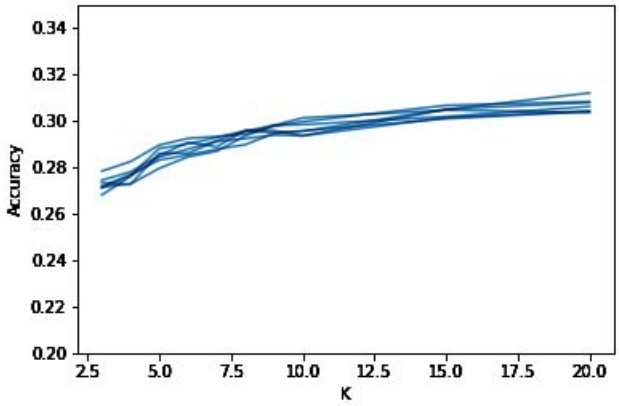

Figure 13: Multiple Run K Accuracy

After running the system as a whole, we decided to see what would happen if we ran specific grades against each other as opposed to all of the grades together. We therefore decided to run grade A vs B, A vs C, A vs D, A vs E, and A vs F & G. What we found was a significant increase in predictive power immediately due to only having to pick from two categories as opposed to seven. In addition, as the grades got further from each other, the accuracy score increased significantly, with the furthest distance predicting correctly 90% of the time.

Grade Comparison	Accuracy
A vs B	59.86%
A vs C	66.15%
A vs D	71.87%
A vs E	81.69%
A vs F&G	90.83%

Figure 14: Grade Comparison Accuracy Scores

The overall results of our model show that from an investor perspective, there is little difference between loan grades that are close to each other (ie. A and B). Despite this, there is a significant difference in the rate of return that an investor can expect to see on the back end due to the interest rate. Therefore, as an investor I would be very likely to choose a loan grade in the C range as they have a lot of similar characteristics to the A grade, but pay out at a higher rate.

In conclusion, due to the size of the data set as well as the potential number of features that could be taken into account, KNN may not be the best algorithm choice for this particular data set. As mentioned earlier, linear regression is the next obvious algorithm to evaluate the data set and it will be able to take additional features into account at a significantly cheaper computation rate.