Project 3 Madelon

November 10, 2017

Mario Burstein

Project Overview

Background

The Madelon dataset represents synthetic data containing features that are used to predict a single class denoted as 1 or -1. There are 20 predictive features, broken down as follows:

- 5 informative features
- 15 linear combinations of of the informative features

We have been provided two different Madelon datasets that vary in size:

- Smaller set ("UCI_Data") – consists of 500 features (columns) and 4.4K observations (rows).
- Larger set ("Full_Data") – consists of 1K features and 200K observations.

The excess features, meaning non-predictive features, are present to create a noise in the dataset.

Problem Statement

Develop a scalable workflow to 1) identify the five informative features, 2) identify optimal models, and 3) generate predictions. The workflow must be able to scale on the Full_Data using an AWS T2.Micro with only 1GB of memory.

Approach

The workflow described in this report was developed using the UCI_Data and then tested against the Full_Data to determine scalability of the process. The high-level approach is described below:

1) **Exploratory Data Analysis ("EDA")** - Establish an understanding of the data and the interaction between features to identify the twenty predictive features.

2) **Benchmarking** – Build baseline models using the twenty predictive features to establish a benchmark for steps 4 and 5.

3) **Feature Selection** - Use feature selection techniques to identify the five informative features from the twenty predictive features.

4) **Model Pipelines** – Apply features to various models to identify optimal predictive model.

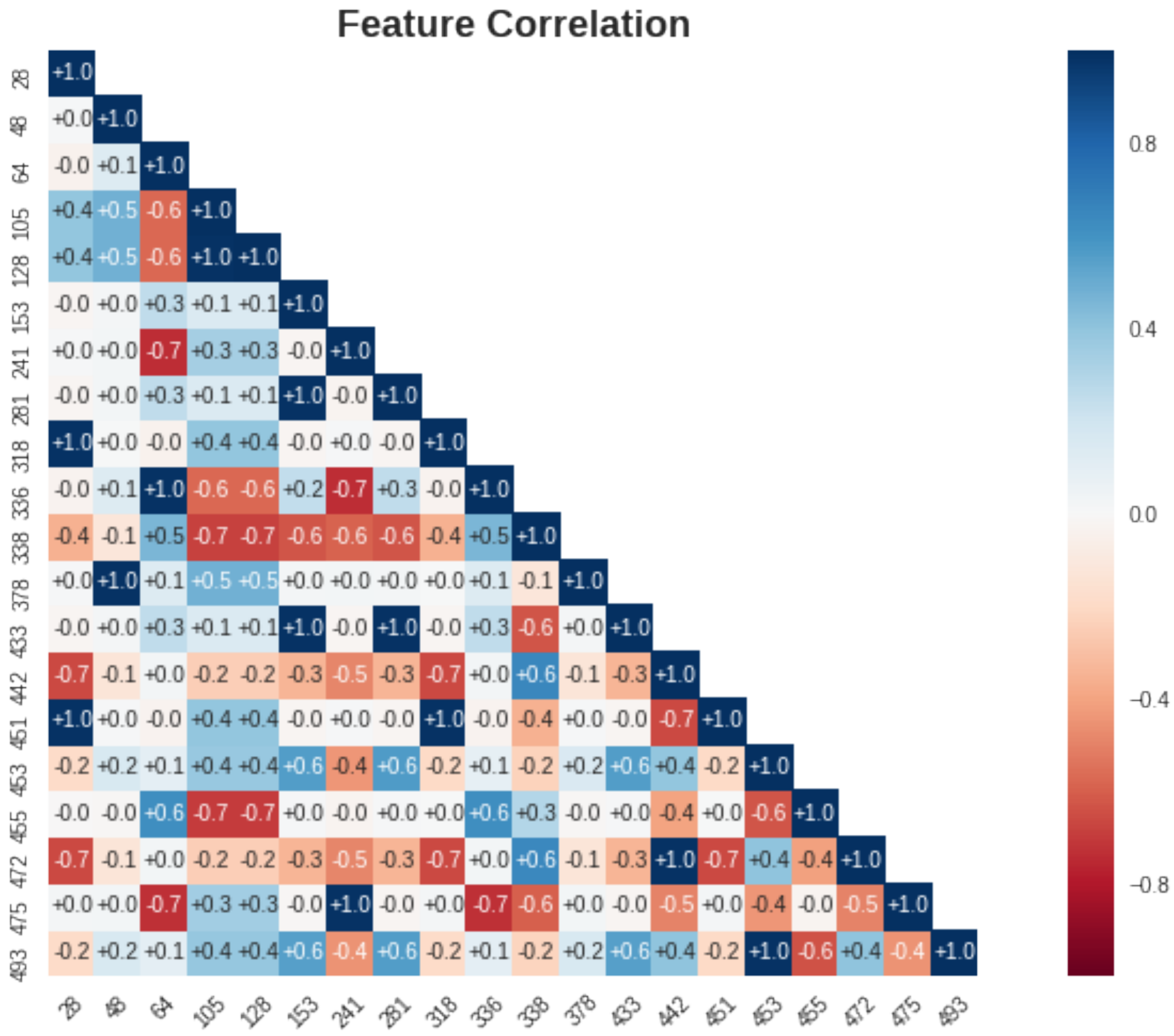
5) **Implement Model** – Tune the model selected from step three and implement the optimal model to generate a prediction score.

6) **Scale Workflow** – Repeat the above steps on the Full_Data.

Step 1: EDA

The adjacent graph presents the absolute value of the correlation coefficients between the twenty predictive features identified in the UCI_Data.

- The twenty features were identified through creating a correlation matrix with all 500 features. The feature was selected if it had an absolute value correlation greater than 0.3 with any other feature. The criteria resulted in exactly twenty features as presented on the X and Y axes of the adjacent graph.
  - The correlation between features was used to identify the twenty predictive features because correlation represents an interaction between two features. Interactions between features creates predictive power, which can increase accuracy of a model, therefore if no interaction with a single feature exists the feature is dropped.
- After narrowing down the features from 500 to 20 the distribution of each feature was plotted (*see Appendix for graphs*). Several features had bimodal distributions, which may be indicative of the two classes, however further feature reduction was required at this point.



Feature Correlation

	28	48	64	105	128	153	241	281	318	336	338	378	433	442	451	453	455	472	475	493
28	+1.0																			
48	+0.0	+1.0																		
64	-0.0	+0.1	+1.0																	
105	+0.4	+0.5	-0.6	+1.0																
128	+0.4	+0.5	-0.6	+1.0	+1.0															
153	-0.0	+0.0	+0.3	+0.1	+0.1	+1.0														
241	+0.0	+0.0	-0.7	+0.3	+0.3	-0.0	+1.0													
281	-0.0	+0.0	+0.3	+0.1	+0.1	+1.0	-0.0	+1.0												
318	+1.0	+0.0	-0.0	+0.4	+0.4	-0.0	+0.0	-0.0	+1.0											
336	-0.0	+0.1	+1.0	-0.6	-0.6	+0.2	-0.7	+0.3	-0.0	+1.0										
338	-0.4	-0.1	+0.5	-0.7	-0.7	-0.6	-0.6	-0.6	-0.4	+0.5	+1.0									
378	+0.0	+1.0	+0.1	+0.5	+0.5	+0.0	+0.0	+0.0	+0.0	+0.1	-0.1	+1.0								
433	-0.0	+0.0	+0.3	+0.1	+0.1	+1.0	-0.0	+1.0	-0.0	+0.3	-0.6	+0.0	+1.0							
442	-0.7	-0.1	+0.0	-0.2	-0.2	-0.3	-0.5	-0.3	-0.7	+0.0	+0.6	-0.1	-0.3	+1.0						
451	+1.0	+0.0	-0.0	+0.4	+0.4	-0.0	+0.0	-0.0	+1.0	-0.0	-0.4	+0.0	-0.0	-0.7	+1.0					
453	-0.2	+0.2	+0.1	+0.4	+0.4	+0.6	-0.4	+0.6	-0.2	+0.1	-0.2	+0.2	+0.6	+0.4	-0.2	+1.0				
455	-0.0	-0.0	+0.6	-0.7	-0.7	+0.0	-0.0	+0.0	+0.0	+0.6	+0.3	-0.0	+0.0	-0.4	+0.0	-0.6	+1.0			
472	-0.7	-0.1	+0.0	-0.2	-0.2	-0.3	-0.5	-0.3	-0.7	+0.0	+0.6	-0.1	-0.3	+1.0	-0.7	+0.4	-0.4	+1.0		
475	+0.0	+0.0	-0.7	+0.3	+0.3	-0.0	+1.0	-0.0	+0.0	-0.7	-0.6	+0.0	-0.0	-0.5	+0.0	-0.4	-0.0	-0.5	+1.0	
493	-0.2	+0.2	+0.1	+0.4	+0.4	+0.6	-0.4	+0.6	-0.2	+0.1	-0.2	+0.2	+0.6	+0.4	-0.2	+1.0	-0.6	+0.4	-0.4	+1.0

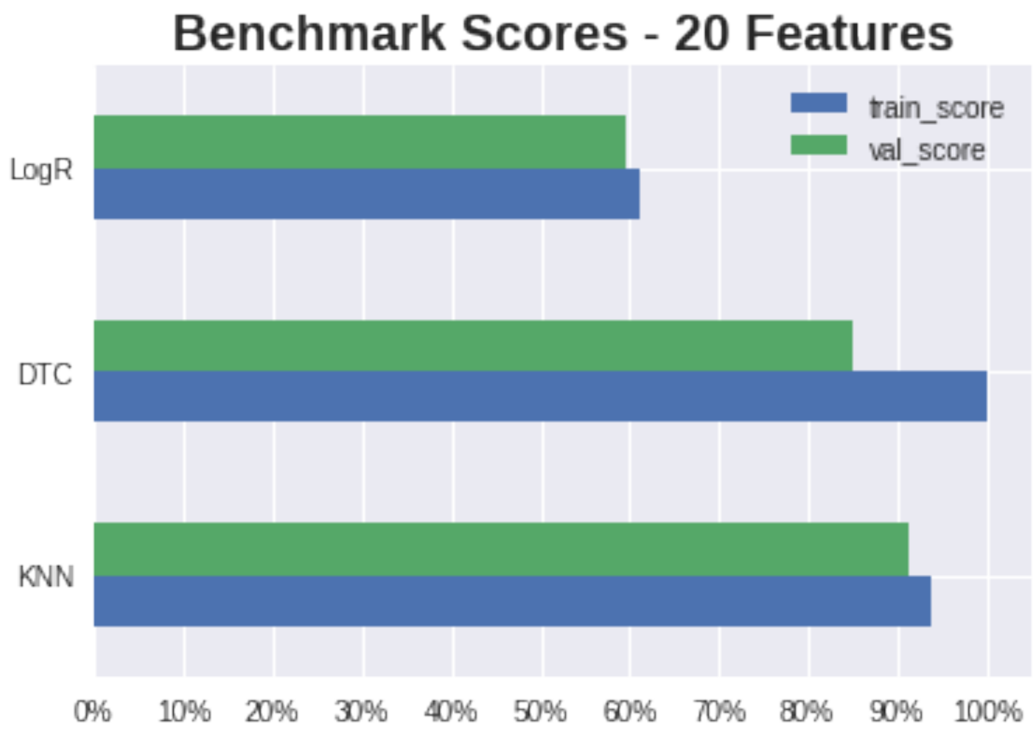
Step 2: Benchmarking

The adjacent graph presents the benchmark scores for the UCI_Data using the twenty predictive features using the following models:

  - DecisionTreeClassifier (“DTC”)
  - KNearestNeighbors (“KNN”)
  - Logistic Regression (“LogR”)

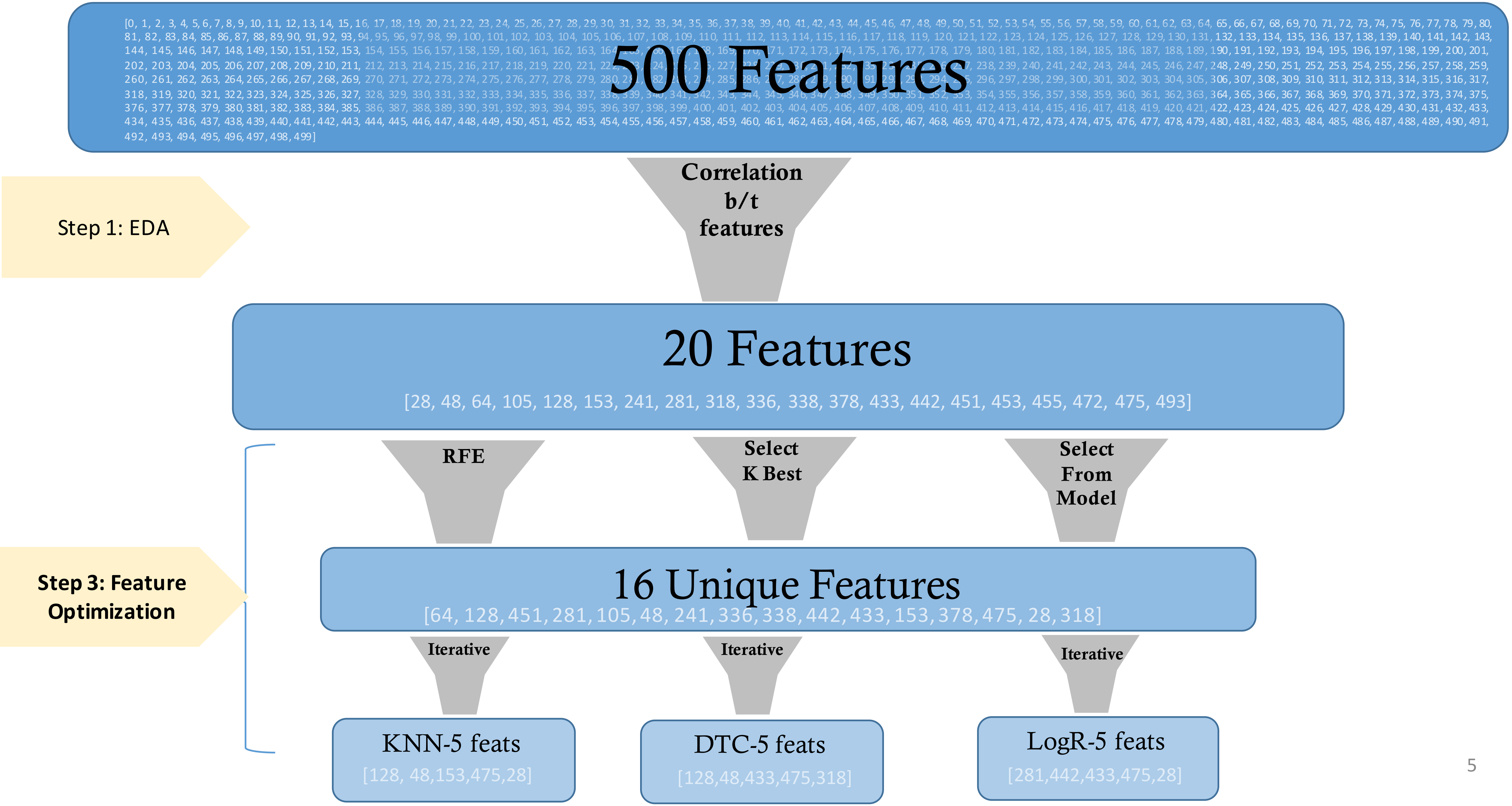
- The benchmark scores were derived using a scaled dataset of twenty features with no model tuning. The validation score benchmarks vary from 60% accuracy with LogR to 90% with KNN.

- Based on benchmark scoring it is evident the LogR model will not perform well with this dataset, which may indicate data complexity and non-linearity.


Benchmark Scores - 20 Features

Model	Train Accuracy	Val Accuracy
LogR	61.05%	59.67%
KNN	93.65%	91.33%%
DTC	100.00%	85.00%

Step 3: Feature Selection Overview



[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143,
144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201,
202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259,
260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317,
318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375,
376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433,
434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491,
492, 493, 494, 495, 496, 497, 498, 499]

500 Features

Step 1: EDA

Correlation
b/t
features

20 Features

[28, 48, 64, 105, 128, 153, 241, 281, 318, 336, 338, 378, 433, 442, 451, 453, 455, 472, 475, 493]

RFE

Select
K Best

Select
From
Model

Step 3: Feature
Optimization

16 Unique Features

[64, 128, 451, 281, 105, 48, 241, 336, 338, 442, 433, 153, 378, 475, 28, 318]

Iterative

Iterative

Iterative

KNN-5 feats

[128, 48,153,475,28]

DTC-5 feats

[128,48,433,475,318]

LogR-5 feats

[281,442,433,475,28]

Step 3: Feature Selection (continued)

Feature Optimization

The following feature selections techniques were implemented to further narrow down the predictive features to the informative features:

1) **Select K-Best ("SKB")** identifies the *K* highest scoring features based on the features F-value. The F-value measures the impact of individual features based on the mean variance between groups when the feature is included. SKB was used to determine the ten (*K)* most significant features.

2) **Recursive Feature Elimination ("RFE")** iteratively removes features based on model accuracy. Using RFE with an LogR and DTC resulted in ten features, of which three were also in the features return from SKB.

3) **Select From Model ("SFM")** returns all features with a coefficient above a determined threshold. Using the median as the threshold ten features were returned, of which one were also in the SKB and RFE feature sets.

The three feature optimization techniques produced conflicting feature sets with minimal overlap. As a result, there were sixteen unique features, which equates to 4.4K combinations of five features.

Iterative Approach

The 4.4K combinations were used in an iterative approach to produce a set of five unique features. Each combination was tested with an un-tuned LogR, DTC, and KNN model to determine the highest accuracy scoring feature sets. Note the entire Train and Validation sets were used.

- The validation accuracy score of each of the three models produced different highest scoring sets of five features as displayed below:

Model	Features	Accuracy
LogR	281, 442, **433**, **475**, **28**	64.33%
KNN	**128** , **48**, 153, *475* , *28*	92.10%
DTC	*128, 48, 433, 475,* 318	85.60%

- The three models did have five common features across the highest scoring combinations, but **generally speaking the un-tuned models with five features did not significantly outperform the benchmark models with twenty features**.

Steps 4 & 5: Model Pipelines & Implementation

Parameter Tuning

Each of the three models was refined through grid searching with five folds and tested against the three feature sets previously discussed, as well as a fourth feature set that included the common features identified during feature selection . Additionally a Random Forest ("RanF") was used on the DTC model.

1) **LogR** performed best with a *C: 10* and *model penalty : l2*

2) **DTC** performed best with a *max_depth: 8* and *min_sample_split: 7*

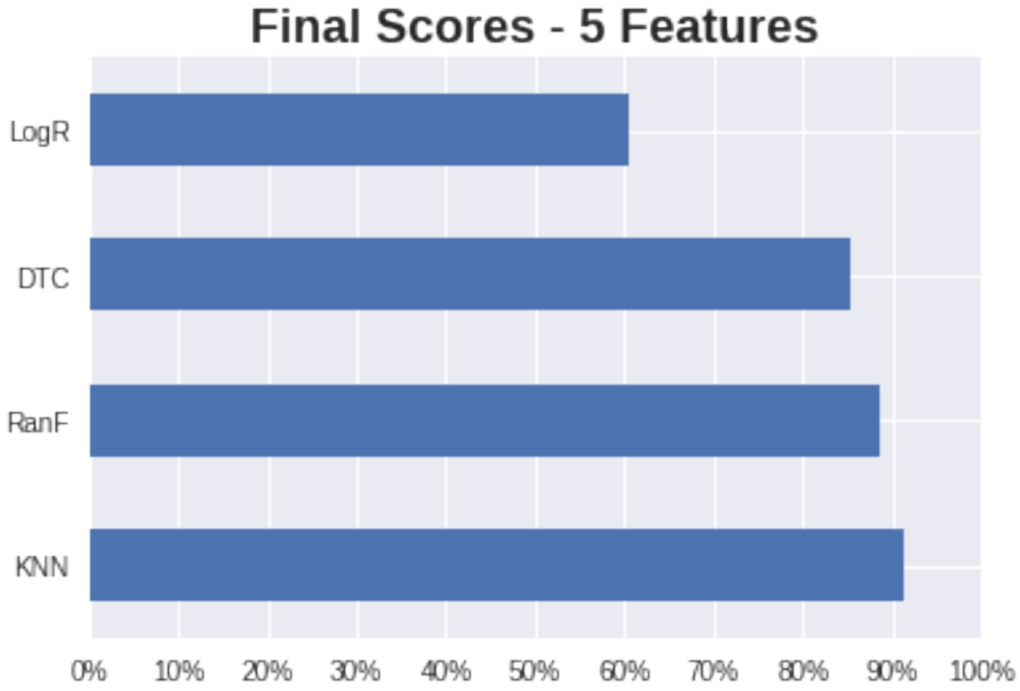
3) **RanF** performed best with *max_features: auto* and *n_estimators: 200*

4) **KNN** performed best with *n_neighbors: 5* ,

Final Scoring

Each of the models performed best with the final feature set comprised of the common features identified during feature selection, however the top scores were derived using nine features.

The upper right and graph displays the final scores for the final five features.



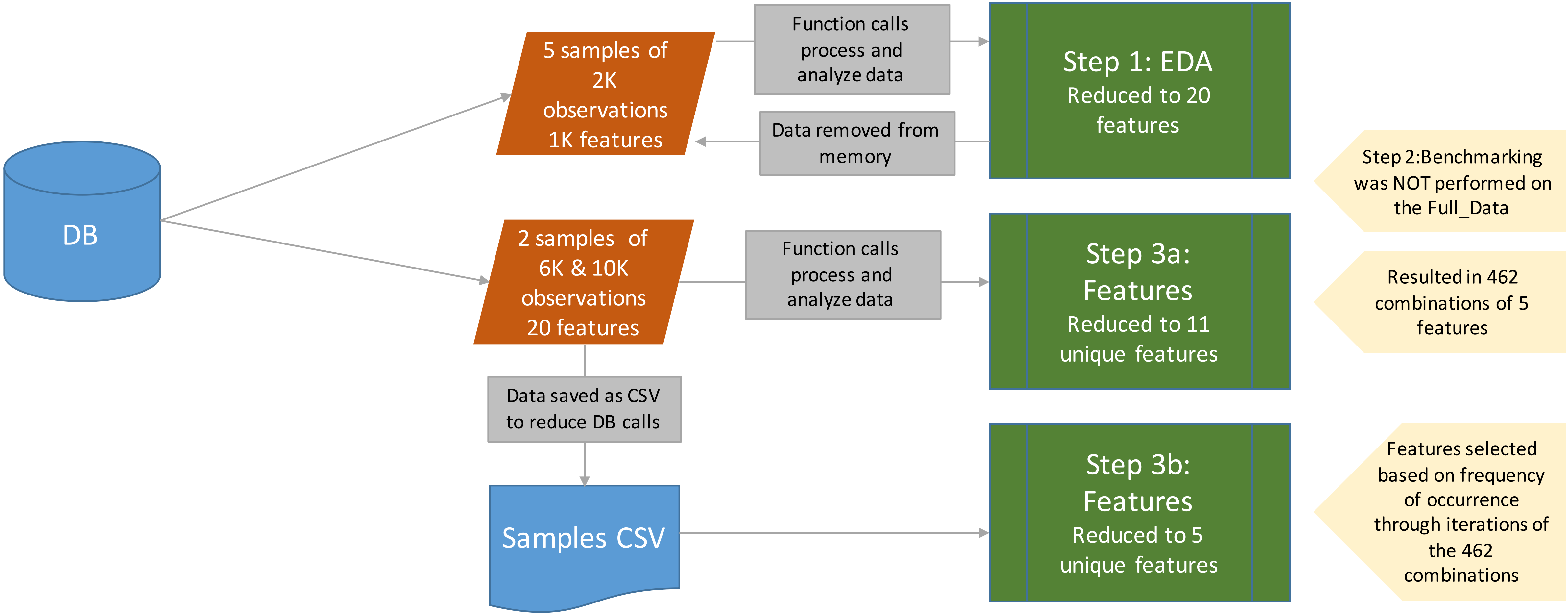
UCI_Data Conclusion

The KNN model performed the best with 5 neighbors and the following features: **128 , 48, 433, 475, 28**. Although the KNN model performed the best it also displayed ZERO improvement, from the benchmark after feature selection and tuning. This indicates that the above features may not be the five informative features. The DTC model with a Random Forest improved 5% from it benchmarks, which indicates the removal of redundant features had a positive impact. *Refer to the Appendix for a visual comparison of the Benchmark scores v. Final scores.*

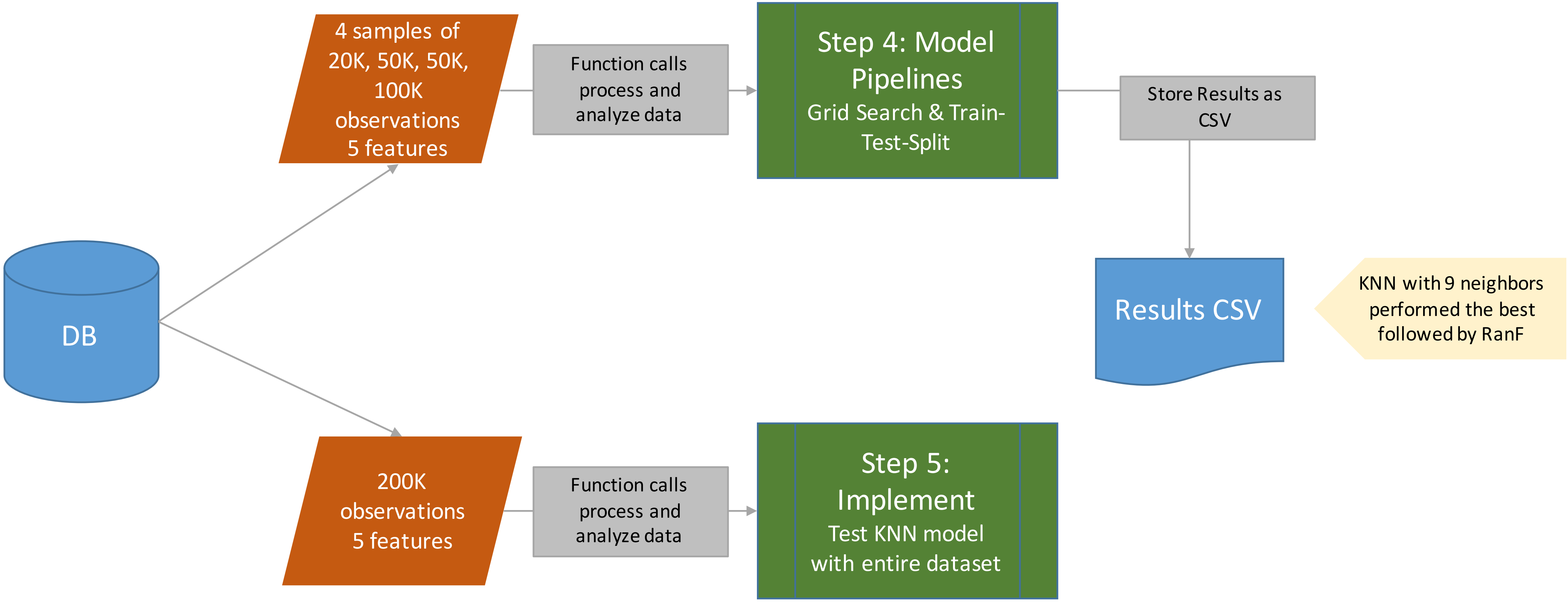
Step 6: Scale Workflow

Scaling Impediments

The Full_Data was comprised of 1K columns and 200K rows, which totaled to ~1.5GB of data. The T2.Micro contained 12 GB of storage and 1GB of memory, however with the Docker Image and library imports only 850GB of memory were available for querying the data. Given the memory constraints the dataset had to be sampled until the features (columns) were reduced. The below diagram outlines the high-level process. Note the methodologies applied in steps 1-5 were the same as previously discuss, unless otherwise mentioned.


DB
5 samples of 2K observations 1K features
Function calls process and analyze data
Step 1: EDA
Reduced to 20 features
Data removed from memory
Step 2:Benchmarking was NOT performed on the Full_Data
2 samples of 6K & 10K observations 20 features
Function calls process and analyze data
Step 3a: Features
Reduced to 11 unique features
Resulted in 462 combinations of 5 features
Data saved as CSV to reduce DB calls
Samples CSV
Step 3b: Features
Reduced to 5 unique features
Features selected based on frequency of occurrence through iterations of the 462 combinations

Step 6: Scale Workflow (continued)



DB

4 samples of 20K, 50K, 50K, 100K observations 5 features

Function calls process and analyze data

Step 4: Model Pipelines

Grid Search & Train-Test-Split

Store Results as CSV

Results CSV

KNN with 9 neighbors performed the best followed by RanF

200K observations 5 features

Function calls process and analyze data

Step 5: Implement

Test KNN model with entire dataset

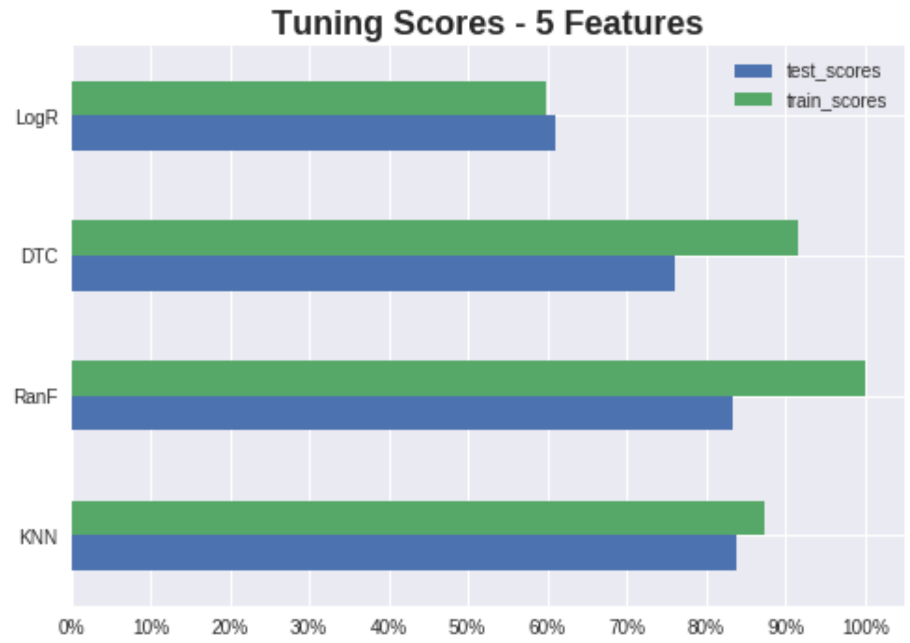
KNN model scoring decreased on larger dataset

Informative Features

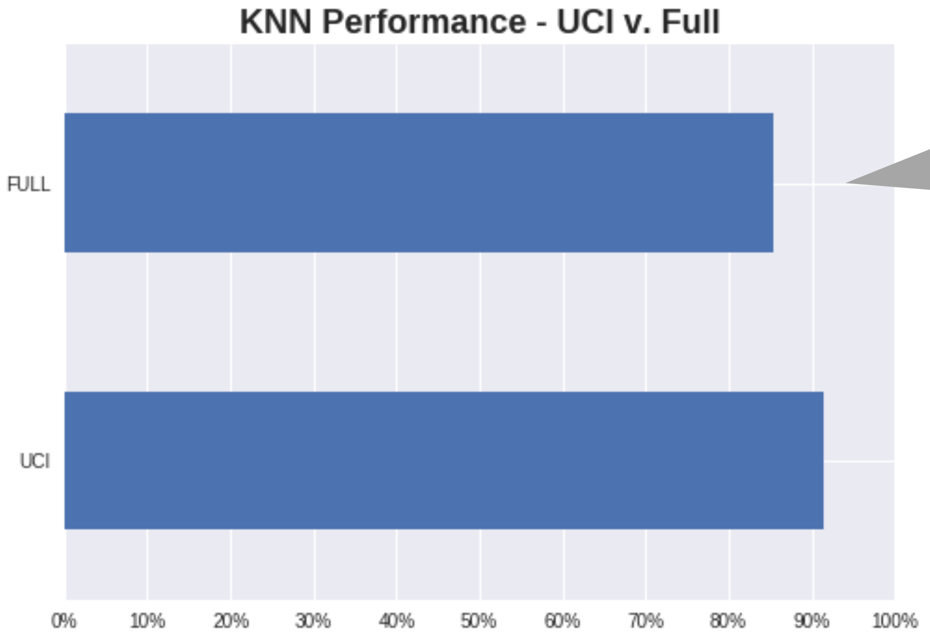
Through EDA and feature selections the following five features were identified as informative: **feat_257 , feat_ 341, feat_681, feat_701, feat_808**

Tuning Results

The below graph represents the top accuracy scores for each model during tuning (Step 4).



KNN Model Scoring Comparison



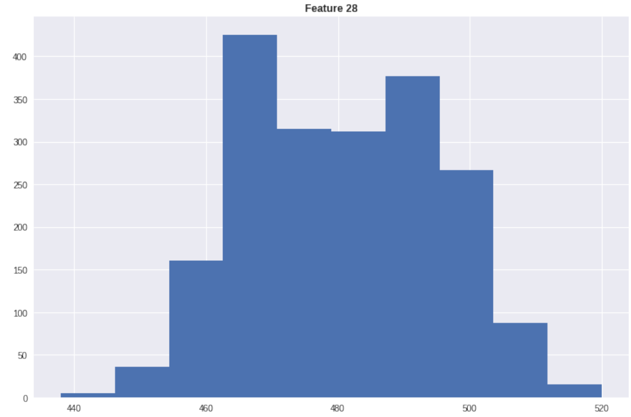
Full_data accuracy was 5% less.

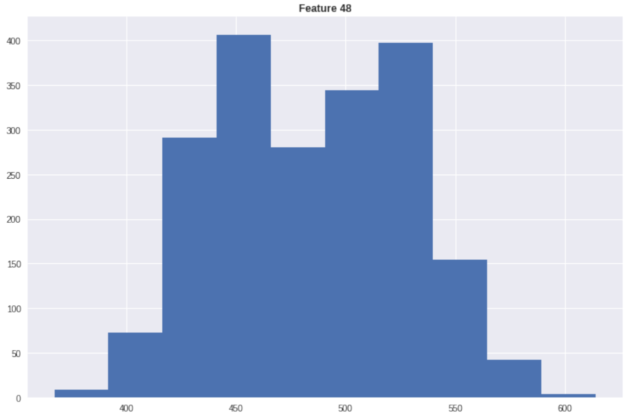
Full_Data Conclusion

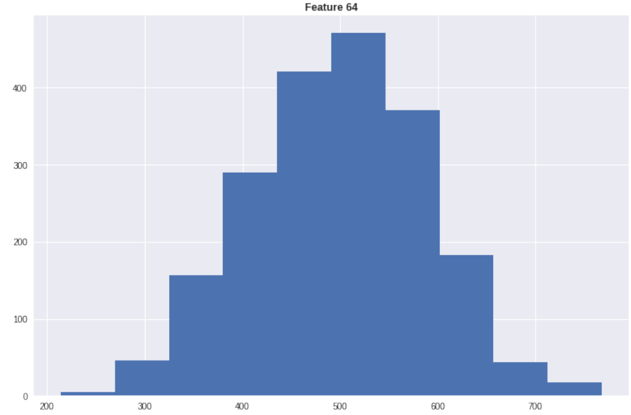
Each of the models were less accurate on the large dataset than the small one, however within the large dataset the models improved as the sample sizes increased and the training improved. For example the KNN model with 9 neighbors scored 83% with a sample size of 100K and 85% with the entire 200K dataset. The large dataset also performed well with a Random Forest and could improve with additional tuning and Boosting.

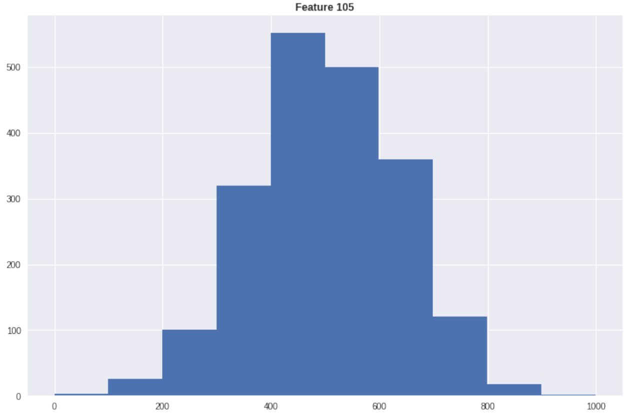
Appendix

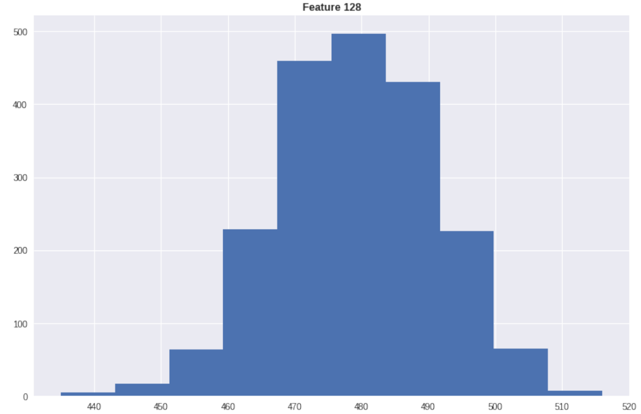
Appendix: UCI_Data EDA

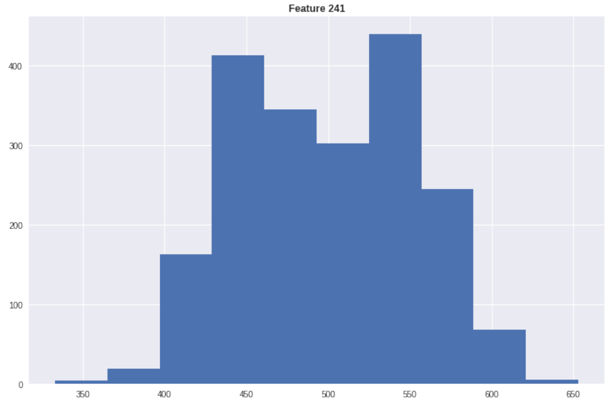

Feature 28

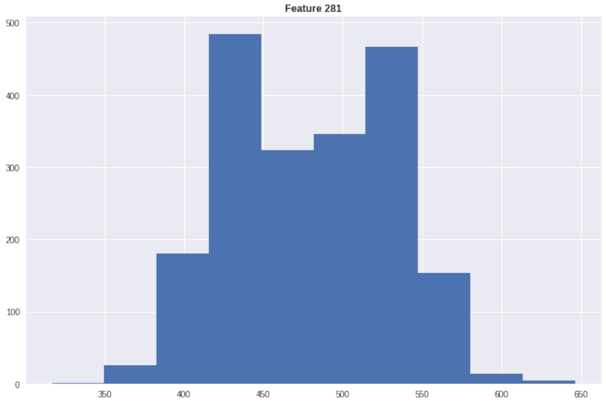

Feature 48

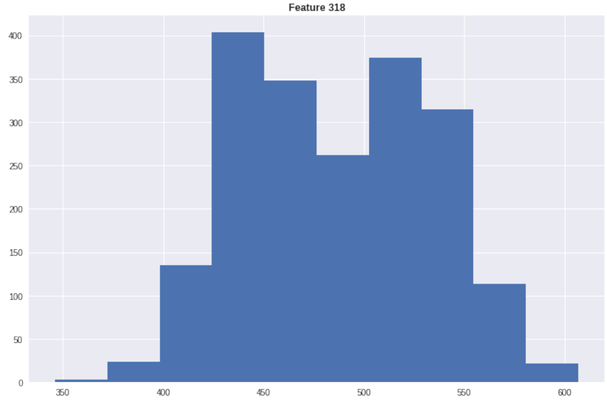

Feature 64

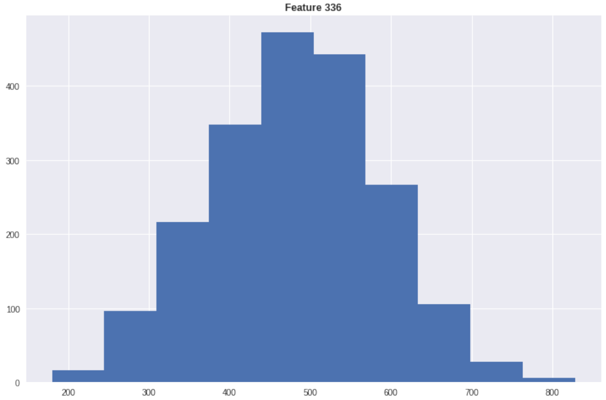

Feature 105

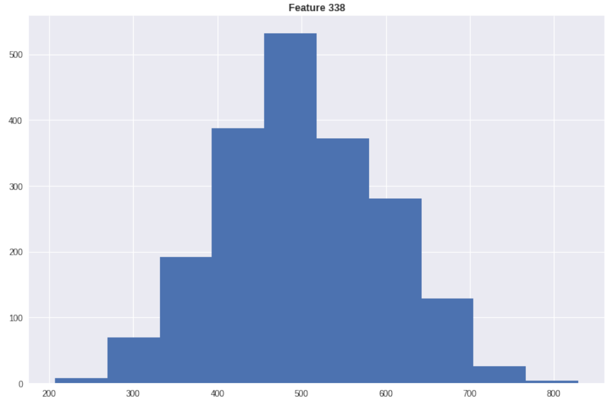

Feature 128


Feature 241

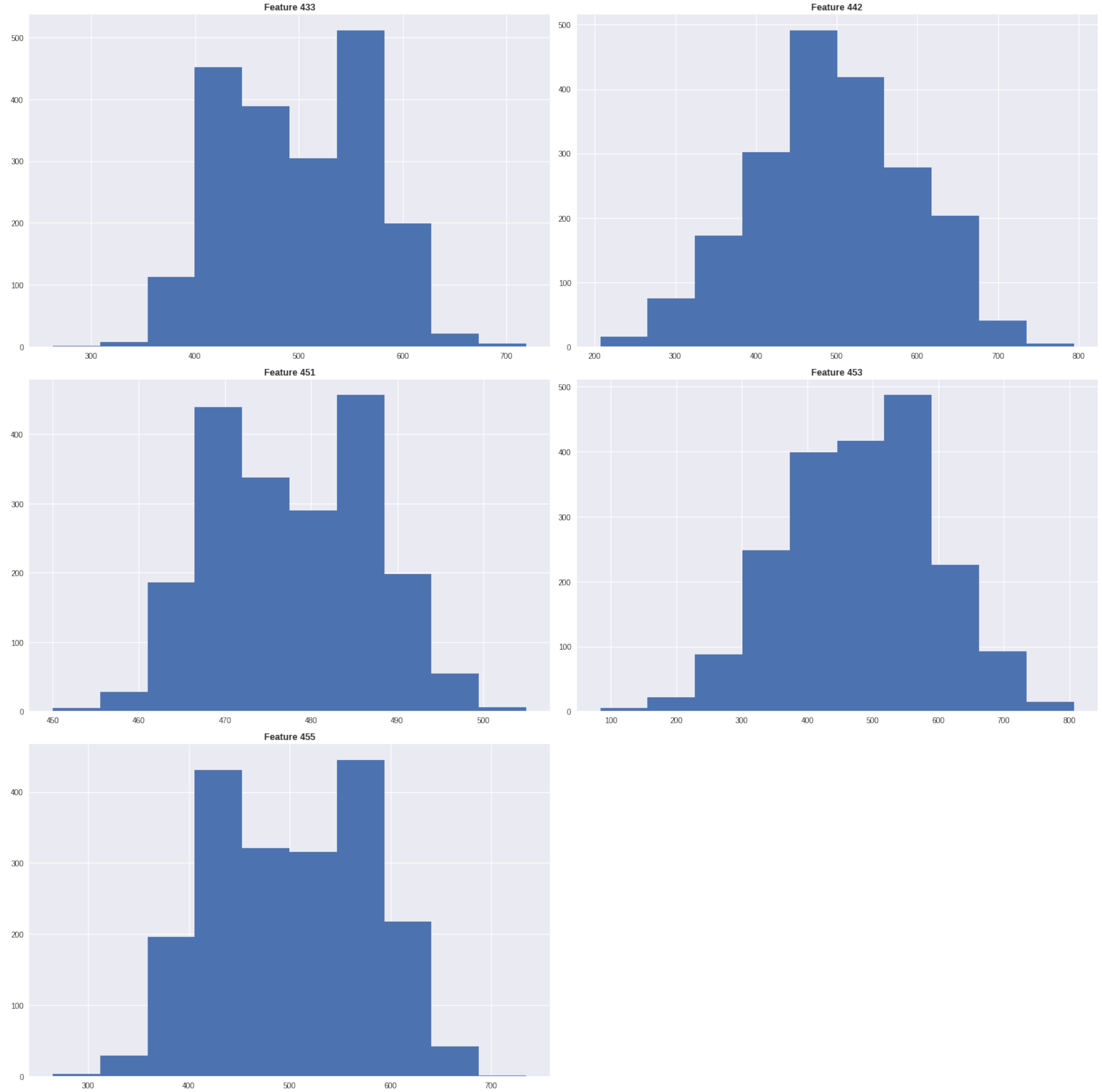

Feature 281

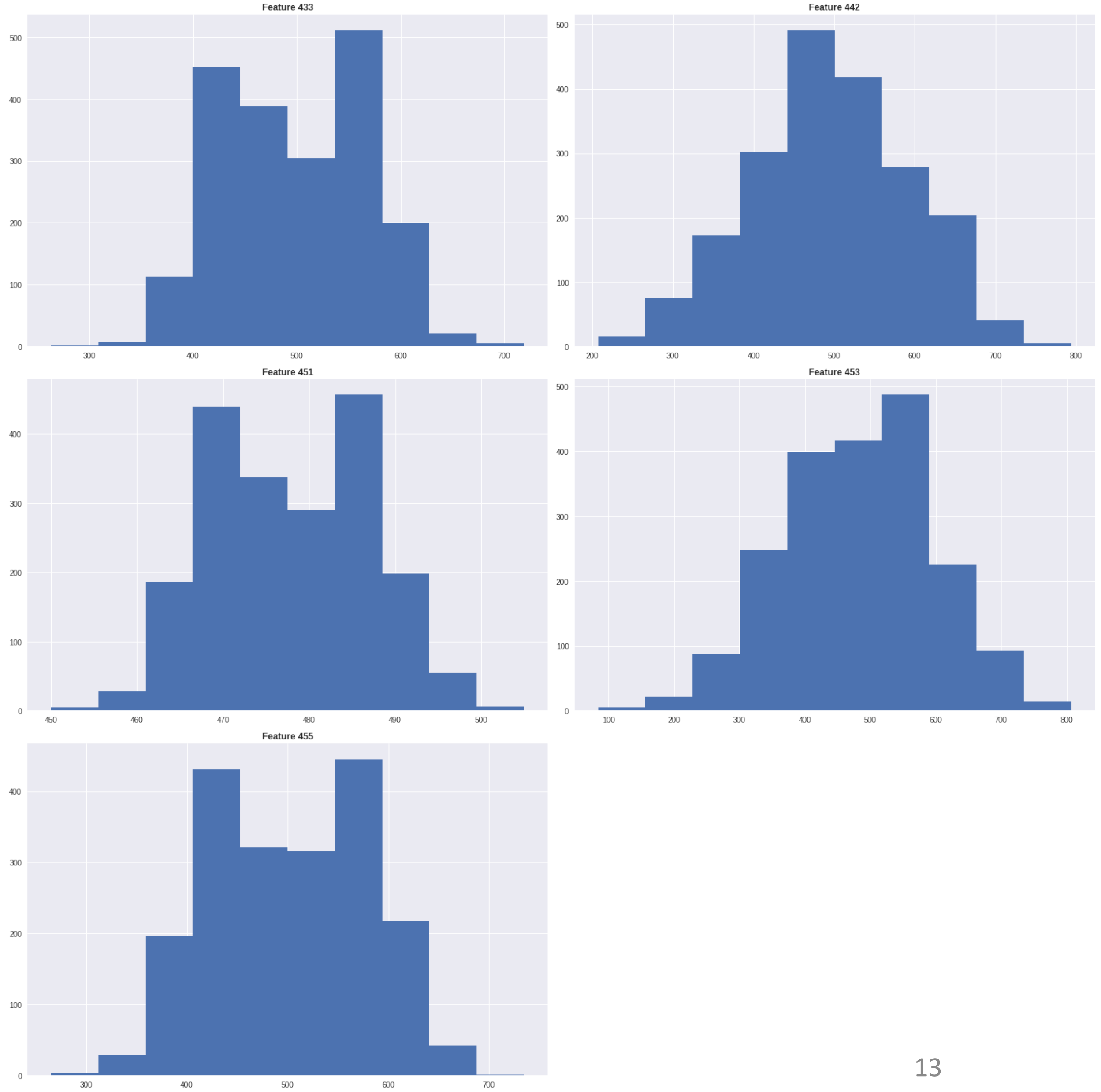

Feature 318


Feature 336

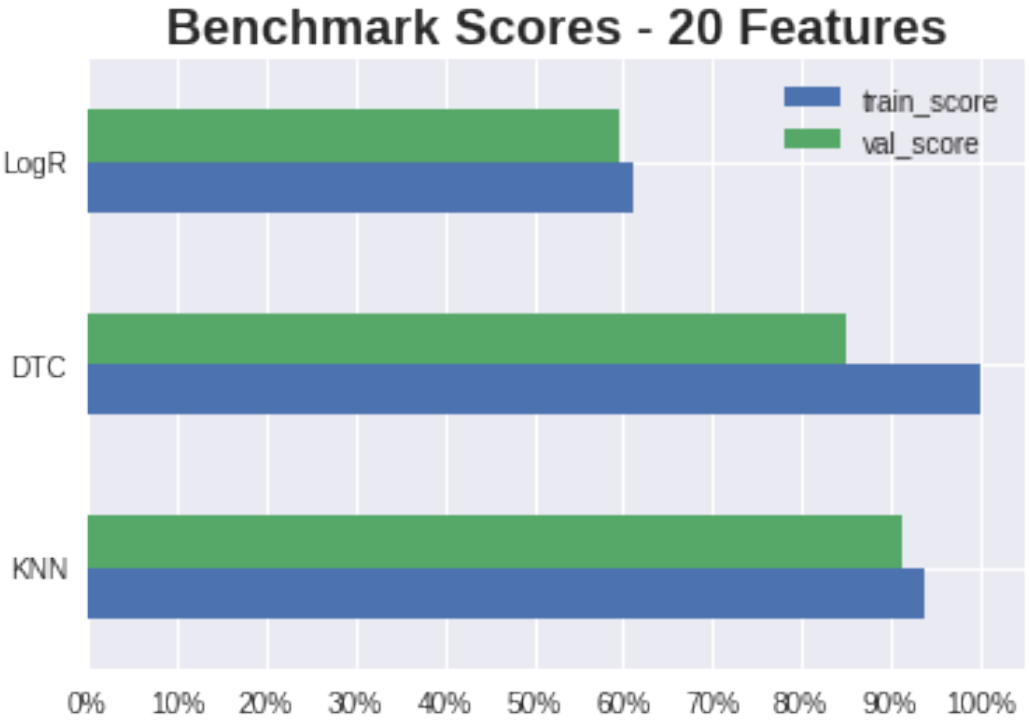

Feature 338

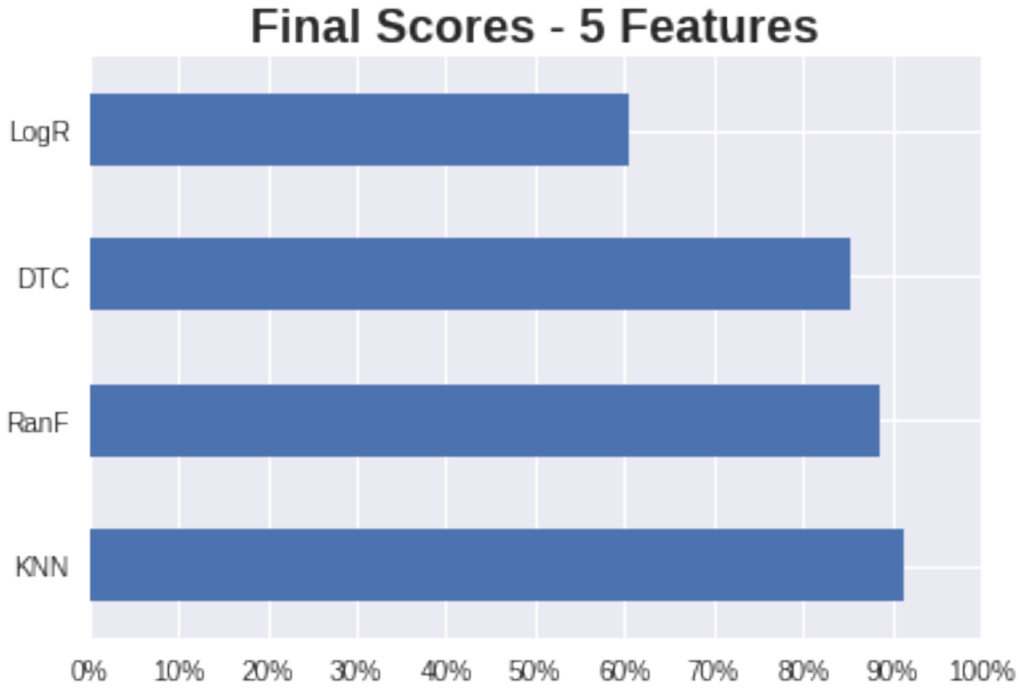
Appendix: UCI_Data EDA





Appendix: UCI_Data Benchmark v. Final


Benchmark Scores - 20 Features


Final Scores - 5 Features