18-785: DATA, INFERENCE & APPLIED MACHINE LEARNING

ASSIGNMENT 7

Yasser Corzo

Carnegie Mellon University

LIBRARIES USED

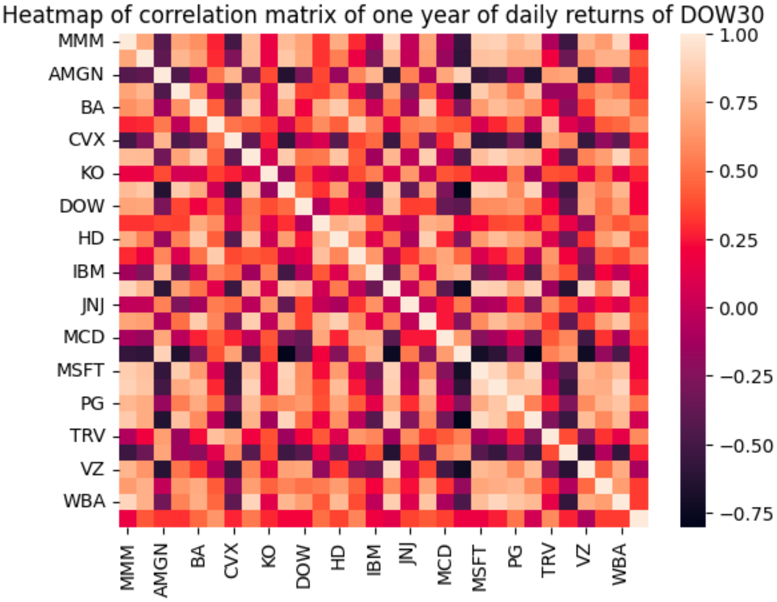
- Matplotlib
- Numpy
- Pandas
- Scipy
- Seaborn
- Sklearn
- yfinance

REPORT

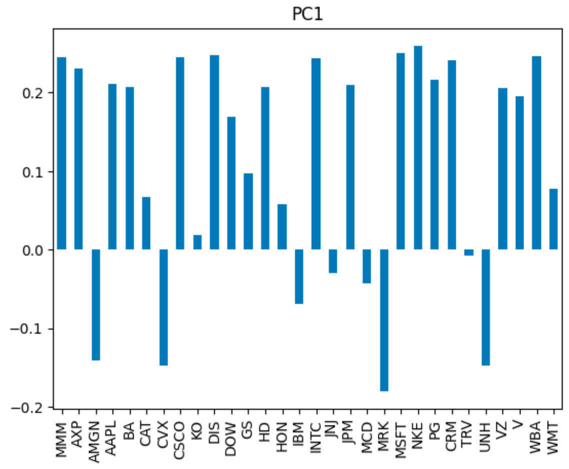
Q1

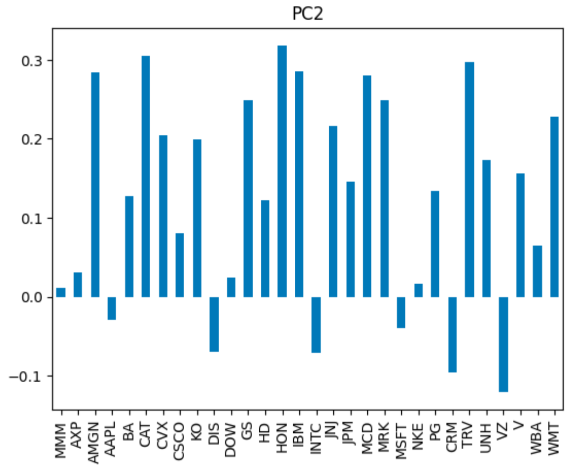
1. PCA finds direction of maximum variance. It does this by performing orthogonal transformations to create linearly uncorrelated variables called PCs. These PCs are ordered in terms of increasing to decreasing variance captured, with the first PC having the most captured variance. PCA also provides an eigenvalue spectrum where each value indicates the amount of variance represented by successive PCs. The applications of PCA in machine learning are in noise reduction, where the contribution of higher components is deleted. It might be useful to consider PCA to transform a set of explanatory variables because PCA helps reducing the dimensionality of datasets while preserving crucial information, especially with datasets that have many explanatory variables that are highly correlated [1]. By transforming the original variables into a set of new, uncorrelated variables (PCs), we can further discard less relevant features.
2. $\boldsymbol{Y} = \boldsymbol{X}\mathbf{V}$

   Each column in **V** represents a vector of weights ($\mathbf{v}_m$) that map each row vector ($\mathbf{x}_m$) in **X** to a new vector of PC scores ($\mathbf{y}_m$). Essentially, the new columns in Y ($\mathbf{y}_m$) capture the maximum possible variance from the data matrix **X**.
3.



This is the correlation matrix of daily returns of the DOW30 companies (for one year, from January 1st 2022 – January 1st 2023). This was done by first getting the daily adjusted close values (for one year) of the 30 DOW Jones companies via the yfinance library. After parsing the data into a pandas dataframe, I used pandas corr() method to calculate Pearson's correlation coefficients of these 30 companies.

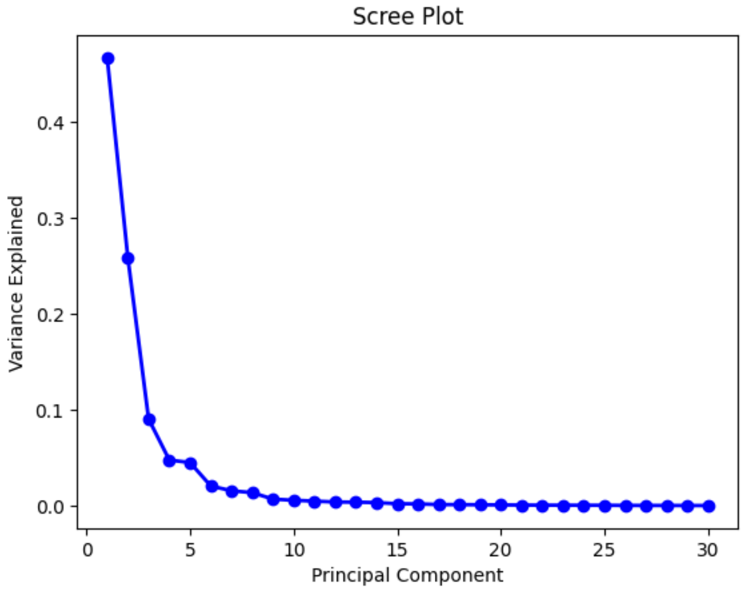




Above are the bar graphs for the first and second principal components (showing the weights of every stock). The second component is more similar to the markey than the first component because there's less variance in the weights, compared to the first components, evident in there being less negative weights in PC2 than in PC1. Most of the stock's adjusted close values in the DOW from this time period increased, thus PC2 is more similar to the market than PC1.

These graphs were calculated by using Sklearn's PCA() class where the first and second components were retrieved, then plotted via matplotlib's bar() function.

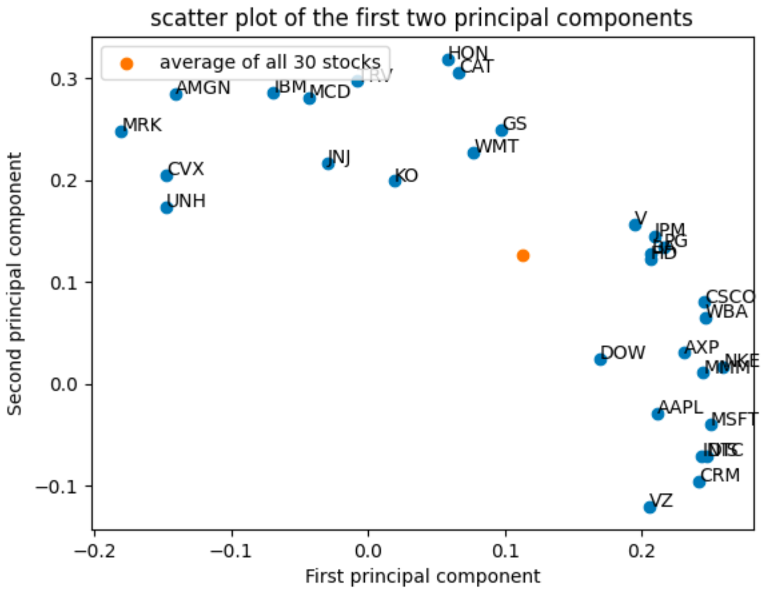
4.



Here is the scree plot where the x-axis represent the principal components and the y-axis represents the amount of variance explained by each principal component.
8 principal components are required to explain 95% of the variance.

The plot was created via sklearn's explained_variance_ratio_ method from the PCA() class for retrieving the amount of variance explained by each principal component, then plotted using matplotlib.

5.



The three most distant stocks are CVX, AMGN, and MRK. These stocks are unusual because their 1st and 2nd principal component datapoints are the furthest away from the average, hence one could infer that their principal components have greater variance than all the other companies' principal components.

The scatter plot was created by the retrieved first and second principal components from the previous question. To calculate the average point, I just calculated the averages of the first and second components from the dataset (using numpy's mean() method), yielding one value for the x-axis and one value for the y-axis.

To calculate the top 3 stocks that were furthest from the average, I used numpy's linalg.norm method which calculated the euclidean distance between two points.

Q2

1. A dendrogram is a tree that shows the relationship between similar objects. The components of a dendrogram include clades (branches), nodes, and leaves. A clade connects leaves and nodes, indicating relationships. Nodes represent the merging of clades, indicating similarity [3].
   There are two methods to constructing a dendrogram: Bottom Up and Top down [3]. In the Bottom-Up algorithm:

1. Start with each data point as a separate cluster.
2. Merge two data points at a time into a new cluster. How the pairs merge involves calculating a similarity or dissimilarity between each merged pair and other samples. There are many ways to calculating the similarity:
   i. Complete linkage: calculates the similarity of the farthest pair.
   ii. Single linkage: calculates the similarity of the closest pair.
   iii. Group average: calculates the similarity between groups.

iv. Centroid similarity: each iteration merges the clusters with the most similar central point.

3. Iterate the pairing process until all data points are merged into a single cluster.

In the Top-Down method:

1. Data starts as one combined cluster.
2. Cluster splits into two parts, according to a degree of similarity.
3. Clusters split into two iteratively until all clusters contain one data point.

Dendrogram are interpreted in terms of how clades are arranged (according to how similar they are). Clades that are close to the same height are like each other; clades with different heights are dissimilar. Essentially, the greater the difference in height, the more dissimilarity.

2. Given a collection of pairwise dissimilarity values, the steps involved in constructing a dendrogram involve using the Bottom-Up approach:
   a. initialize all values in separate clusters such that each cluster has one data point. We'll use the collection of pairwise dissimilarity values as our distance matrix [2].
   b. Run the Hierarchical Clustering Algorithm where you find the closest pair of clusters by looking for the least distance between two clusters in the distance matrix (containing collection of dissimilarity values) and merging those two clusters together. Update the distance matrix after forming a cluster by recalculating the dissimilarities between the new cluster and all other clusters and updating the matrix with the new dissimilarity values [2].
   c. Repeat previous step until all clusters/points are grouped into one cluster.
3. Distance matrix:

```
          MMM       AXP      AMGN      AAPL        BA       CAT       CVX  \
MMM   0.000000  0.770925  1.692126  0.789961  0.886265  1.204774  1.733314
AXP   0.770925  0.000000  1.665663  0.684975  0.811588  1.187361  1.583920
AMGN  1.692126  1.665663  0.000000  1.705716  1.510213  0.972853  0.693379
AAPL  0.789961  0.684975  1.705716  0.000000  0.874454  1.431547  1.710858
BA    0.886265  0.811588  1.510213  0.874454  0.000000  1.058993  1.648048
CAT   1.204774  1.187361  0.972853  1.431547  1.058993  0.000000  0.989679
CVX   1.733314  1.583920  0.693379  1.710858  1.648048  0.989679  0.000000
CSCO  0.644043  0.650838  1.626328  0.788415  0.530198  1.054010  1.662028
KO    1.289123  1.299856  1.115167  1.376962  1.362147  1.138227  1.211957
DIS   0.657839  0.571743  1.810096  0.548148  0.762465  1.382514  1.779598
DOW   0.793034  0.770245  1.596440  1.126116  1.268387  1.116463  1.426973
GS    1.171911  1.173273  1.126533  1.139182  0.766930  0.902651  1.357330
HD    0.743434  0.952566  1.524219  0.942903  0.552323  1.060700  1.683202
HON   1.188937  1.292715  0.924622  1.352630  0.996644  0.534993  1.123789
IBM   1.501845  1.591992  0.682985  1.656597  1.413228  0.928634  1.023362
INTC  0.456710  0.679752  1.788090  0.820499  0.994748  1.326175  1.754962
JNJ   1.426034  1.422675  0.944117  1.595377  1.499545  0.966220  1.021802
JPM   0.794376  0.815735  1.482007  1.010187  0.540807  0.925963  1.591541
MCD   1.486607  1.530315  0.777127  1.444112  1.204904  1.061173  1.197312
MRK   1.773163  1.786214  0.469708  1.819622  1.591521  1.095267  0.804001
MSFT  0.517578  0.611832  1.773311  0.532591  0.831394  1.361776  1.773267
NKE   0.488041  0.604434  1.733931  0.754841  0.648654  1.193524  1.760311
PG    0.678962  0.742423  1.532962  0.950782  0.756645  1.031877  1.616800
CRM   0.548902  0.740950  1.812881  0.614113  0.901481  1.429246  1.812087
TRV   1.471453  1.266614  0.813469  1.522977  1.240989  0.624053  0.776990
UNH   1.753331  1.636297  0.784380  1.519067  1.559009  1.333697  0.915631
VZ    0.696189  0.856285  1.803259  0.991354  1.155159  1.457435  1.763449
V     0.824326  0.717441  1.422677  0.854675  0.757559  1.068917  1.542440
```

```
WBA   0.443513  0.719690  1.625091  0.936015  0.735459  1.035779  1.669219
WMT   1.291164  1.087269  1.172709  1.174804  1.027807  0.857226  1.201661

      CSCO        KO       DIS  ...      MSFT       NKE        PG  \
   MMM   0.644043  1.289123  0.657839  ...  0.517578  0.488041  0.678962
   AXP   0.650838  1.299856  0.571743  ...  0.611832  0.604434  0.742423
   AMGN  1.626328  1.115167  1.810096  ...  1.773311  1.733931  1.532962
   AAPL  0.788415  1.376962  0.548148  ...  0.532591  0.754841  0.950782
   BA    0.530198  1.362147  0.762465  ...  0.831394  0.648654  0.756645
   CAT   1.054010  1.138227  1.382514  ...  1.361776  1.193524  1.031877
   CVX   1.662028  1.211957  1.779598  ...  1.773267  1.760311  1.616800
   CSCO  0.000000  1.369553  0.546493  ...  0.625244  0.464615  0.628985
   KO    1.369553  0.000000  1.523020  ...  1.322560  1.324437  0.970591
   DIS   0.546493  1.523020  0.000000  ...  0.487474  0.522710  0.896626
   DOW   0.998440  1.110875  1.025897  ...  0.896917  0.890104  0.840441
   GS    0.972627  1.328669  1.176184  ...  1.240235  1.120960  1.103938
   HD    0.593573  1.396029  0.823400  ...  0.834156  0.648403  0.786134
   HON   1.079027  1.111959  1.386319  ...  1.355371  1.223372  1.061907
   IBM   1.508210  0.949698  1.731559  ...  1.620084  1.537417  1.318568
   INTC  0.689636  1.360770  0.595886  ...  0.488376  0.516790  0.766541
   JNJ   1.441788  0.854829  1.653566  ...  1.490223  1.462154  1.186060
   JPM   0.529684  1.427176  0.776612  ...  0.901759  0.651903  0.797934
   MCD   1.424218  1.090795  1.589835  ...  1.565693  1.487103  1.316751
   MRK   1.712616  1.133830  1.896389  ...  1.846907  1.791235  1.577703
   MSFT  0.625244  1.322560  0.487474  ...  0.000000  0.456455  0.744395
   NKE   0.464615  1.324437  0.522710  ...  0.456455  0.000000  0.573896
   PG    0.628985  0.970591  0.896626  ...  0.744395  0.573896  0.000000
   CRM   0.703945  1.503996  0.445326  ...  0.393844  0.559702  0.942542
   TRV   1.268114  1.102093  1.515295  ...  1.506927  1.427355  1.209707
   UNH   1.685244  1.125085  1.745937  ...  1.693131  1.762103  1.591566
   VZ    0.937082  1.318101  0.797196  ...  0.670079  0.713539  0.898830
   V     0.820816  1.050049  0.928347  ...  0.775141  0.736301  0.693307
   WBA   0.450379  1.322003  0.682176  ...  0.637102  0.435653  0.604641
   WMT   0.971768  1.195333  1.246809  ...  1.281229  1.210776  0.990831

     CRM       TRV       UNH        VZ         V       WBA       WMT
MMM   0.548902  1.471453  1.753331  0.696189  0.824326  0.443513  1.291164
AXP   0.740950  1.266614  1.636297  0.856285  0.717441  0.719690  1.087269
AMGN  1.812881  0.813469  0.784380  1.803259  1.422677  1.625091  1.172709
AAPL  0.614113  1.522977  1.519067  0.991354  0.854675  0.936015  1.174804
BA    0.901481  1.240989  1.559009  1.155159  0.757559  0.735459  1.027807
CAT   1.429246  0.624053  1.333697  1.457435  1.068917  1.035779  0.857226
CVX   1.812087  0.776990  0.915631  1.763449  1.542440  1.669219  1.201661
CSCO  0.703945  1.268114  1.685244  0.937082  0.820816  0.450379  0.971768
KO    1.503996  1.102093  1.125085  1.318101  1.050049  1.322003  1.195333
DIS   0.445326  1.515295  1.745937  0.797196  0.928347  0.682176  1.246809
DOW   1.011505  1.258606  1.686019  0.769020  1.004839  0.818855  1.246160
GS    1.284439  1.085470  1.300634  1.540791  0.958099  1.085135  1.008774
HD    0.882152  1.340502  1.624828  1.163449  0.822929  0.663645  1.132736
HON   1.438139  0.848503  1.259670  1.574232  1.048310  1.115779  0.928096
IBM   1.688179  0.922020  1.104437  1.633088  1.252180  1.431301  1.273888
INTC  0.479535  1.511374  1.808991  0.409379  0.922209  0.512246  1.324729
JNJ   1.580162  0.898358  1.126056  1.390151  1.238114  1.343177  1.136097
JPM   0.905707  1.161823  1.663287  1.127877  0.799136  0.591713  1.078737
MCD   1.617567  1.077307  0.922025  1.735227  1.170617  1.482917  1.141411
MRK   1.898302  0.943757  0.831398  1.859926  1.534716  1.717977  1.284852
MSFT  0.393844  1.506927  1.693131  0.670079  0.775141  0.637102  1.281229
NKE   0.559702  1.427355  1.762103  0.713539  0.736301  0.435653  1.210776
PG    0.942542  1.209707  1.591566  0.898830  0.693307  0.604641  0.990831
CRM   0.000000  1.586454  1.755908  0.645829  0.906103  0.669335  1.380709
TRV   1.586454  0.000000  1.118087  1.566338  1.173078  1.311061  0.909043
```

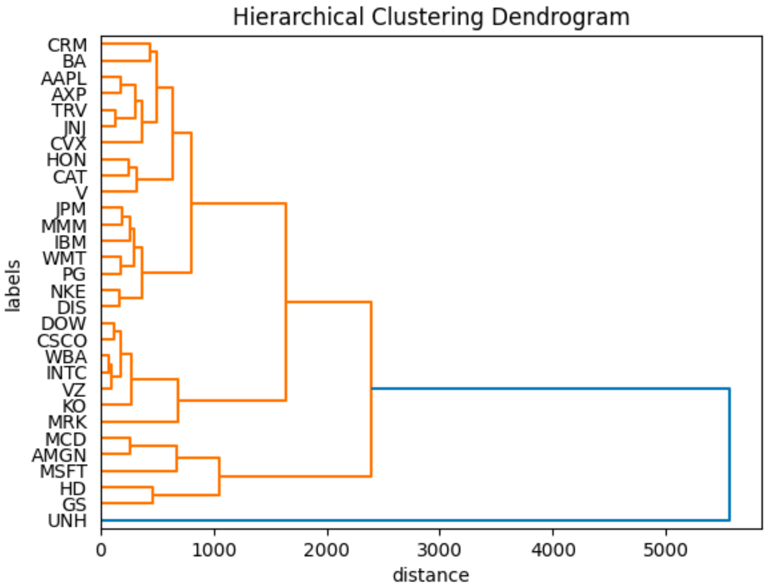
```
UNH   1.755908  1.118087  0.000000  1.799841  1.461445  1.773704  1.207225
VZ    0.645829  1.566338  1.799841  0.000000  1.025126  0.743987  1.486048
V     0.906103  1.173078  1.461445  1.025126  0.000000  0.813617  1.152568
WBA   0.669335  1.311061  1.773704  0.743987  0.813617  0.000000  1.156038

WMT   1.380709  0.909043  1.207225  1.486048  1.152568  1.156038  0.000000
```

Above is the pairwise distances derived from the correlation matrix from Q1.3. The formula for this rescaled distance is: $d_{ij} = (2(1 – p_{ij}))^{-1/2}$. This was the formula used to convert the correlation matrix from Q1.3 into a distance matrix.

Small distances are interpreted by the Hierarchical Clustering Algorithm as a two datapoints being very "similar" thus are grouped into one cluster. The larger the distances, the less similar two datapoints/clusters are, hence the less likely these two points/clusters are to be clustered into the same group.

4.

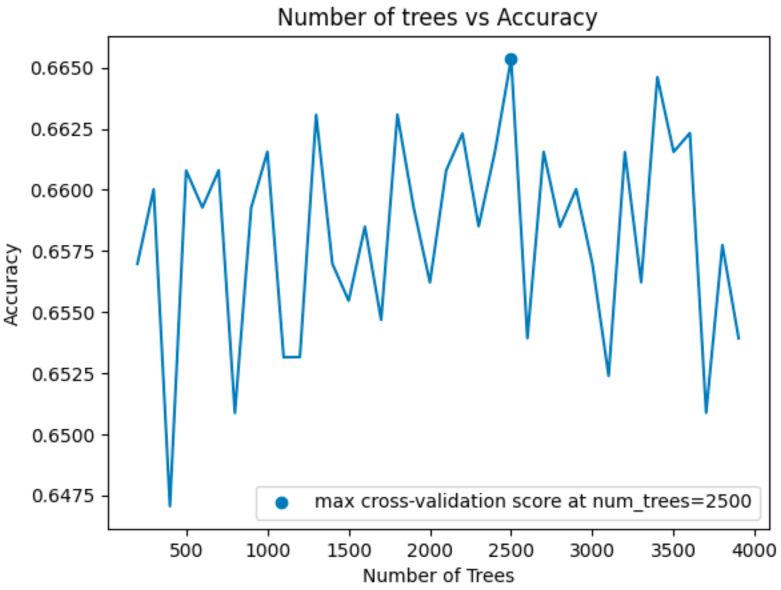


This dendrogram was created using scipy's linkage function to create the linkage matrix (with average linkage). Scipy's dendrogram function was used, along with the linkage matrix, to construct the dendrogram above.

5. The one cluster that stands out is UNH. This cluster has one sole leaf which could be explained by UNH being the only company in the DOW30 in the managed health care industry. One other cluster that stands out is HON (Honeywell) and CAT (Caterpillar). Honeywell is a conglomerate that operates in aerospace, building technologies, performance materials and technologies, and safety and productivity solutions [8]. Caterpillar specializes in building equipment for construction and mining [9]. Since both companies operate in the construction space, one could infer that would be the reason they are grouped in one cluster. One other cluster that stands out is WMT (Walmart) and PG (Proctor & Gamble). Walmart is heavily involved in the retail business [11] and Proctor & Gamble is a multinational consumer goods corporation [10], hence both work in consumer sales which would explain why both companies are clustered together.

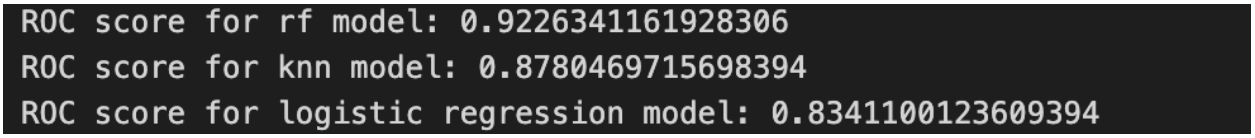
Q3

1. Three sources of uncertainty are observational, parametrical, and structural uncertainty. In the case of observational uncertainty, the impact this has on the modeling process when using ML approaches is on the outputs. Variability in observations not only affects the inputs but also the outputs [4]. For instance, an observation could be incorrectly labeled, producing a false sense that our model could be incorrect when it is in fact the data that is incorrect. Parametrical uncertainty can affect the ability of the model to generalize over unseen data. If the scope of the data isn't enough for the model to capture patterns, this will result in underfitting, affecting the ability of the model to make predictions over unseen data. However, in all cases, we will never have all the observations. This means that there will always be unobserved cases. No matter how hard we make our models generalize, models will only be limited by the cases presented in training datasets. Because of this uncertainty we split a dataset into train and test sets or use resampling methods to handle the uncertainty in the representation of the dataset and estimate the accuracy of a model on unseen data [4]. Structural uncertainty refers to uncertainty on not just in the model, but also in the whole procedure used to prepare the model, including the choice and preparation of data, choice of hyperparameters, etc. This uncertainty is stemmed from uncertainty in the observations (because of noise) and in parameters (stemmed from always having incomplete coverage in the domain). The impact structural uncertainty has on the modeling process is that we seek a model that is just good enough, interpreted as choosing a model that is skillful compared to a naïve method or other established models [4].
2. The concept behind model averaging is that it helps account for uncertainty inherent in the model selection process. To be more specific, such as in the case of Bayesian Model Averaging, parameter uncertainty is modeled through the prior distribution and model uncertainty by obtaining posterior parameter and model posteriors using Bayes' theorem. This technique can be implemented when generating predictions by averaging over many different competing models.
   Two examples where model averaging is implemented in practice to generate predictions are in Finance and in Natural Language Processing. In finance, various models can be used for predicting the stock market [5]. Model averaging can be applied by combining predictions from different models to get a more reliable prediction. In NLP, specifically sentiment analysis in social media, different models can be trained using different natural language processing techniques [6]. Combining their predictions can lead to more accurate predictions.
3. Kinds of ensemble methods that can be used to reduce the effects of uncertainty and improve on individual models are bagging, boosting, and stacking. Bagging achieves this goal by generates M new training sets (given a training set X of size N), each of size N', by sampling from X uniformly and with replacement. The M models are then fitted using these M bootstrap samples and combined by averaging the output. This method reduces variance which in turn increases accuracy, eliminating overfitting. Boosting involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models mis-classified. Essentially, boosting works by arranging weak learners in a sequence, such that weak learners learn from the next learner in the sequence to create better predictive models [7]. Thus, with better predictive models, accuracy increases. Stacking allows a training algorithm to ensemble several other similar learning algorithm predictions [7]. It is often used to measure the error rate during bagging, hence giving the statistician a clearer picture how they're generating ensembles.
4.



To build a random forest model, we need to know which parameter value to choose from for the number of trees. The optimal number of trees I found was 2500 trees. This value was calculated via 5-fold cross validation on every random forest model with trees from 200-4000. After each cross validation, the score was calculated. The model with n trees that yielded the highest cross validation score was considered the best model. The graph above backs my results.

5.



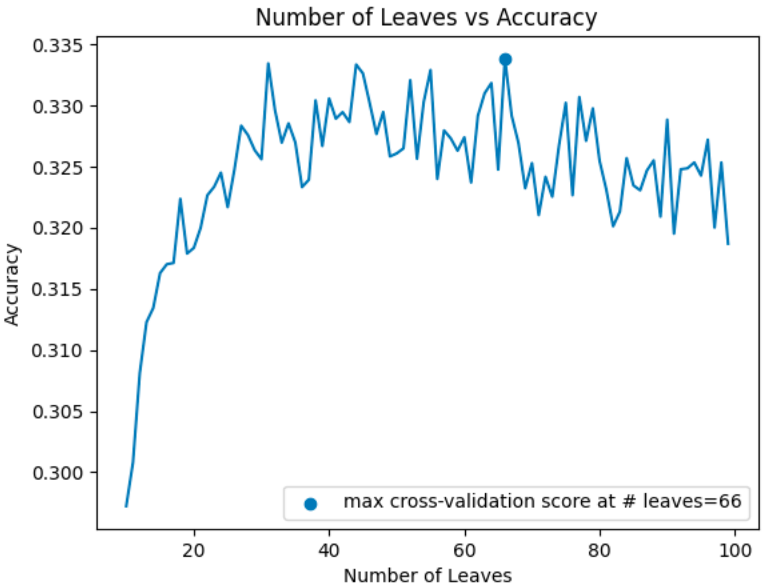
```
ROC score for rf model: 0.9226341161928306
ROC score for knn model: 0.8780469715698394
ROC score for logistic regression model: 0.8341100123609394
```

From the results above, Random Forests is the best possible model (compared to KNN and Logistic Regression) for classifying survival on the Titanic as the ROC score for Random Forests is the highest.

These results were calculated via sklearn's RandomForestClassifier(), KNeighborsClassifier(), and LogiticRegression() class for building the model. For Random Forest, I used the best number of trees parameter obtained from the previous question. For KNeighborsClassifier(), I calculated the optimal k value via cross-validation, using a very similar approach similar to the previous problem (when calculating the optimal number of trees value). By using sklearn's roc_auc_score() function for ROC analysis, I was able to calculated the scores above.

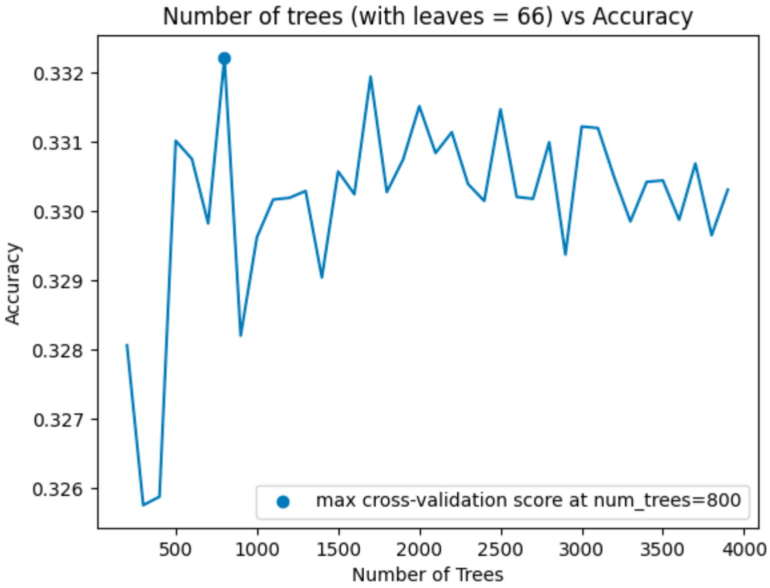
Q4

1. A random forest regression model is an approach which combines bagging of trees with the random selection of features. Random Forests averages across multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance.
2.



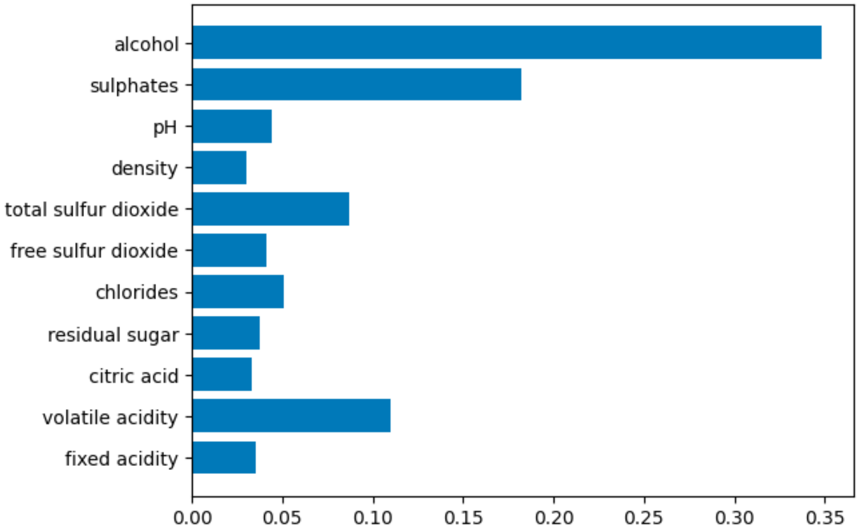
The optimal number of leaves were estimated via cross validation. I fed the model a wide range of number of leaves values ranging from 10-100, fitting the wine data on the Random Forest Regression model with the given number of leaves parameter, performing cross validation and calculating the cross-validation score. The model with the highest score is considered the best model and thus contains the optimal value for the number of leaves, which in this case turned out to be around 66.

3.



The optimal value for number of trees was calculated via cross-validation. I fed the RandomForestRegressor model (from sklearn) a wide range of number of leaves values ranging from 200-4000 (with a step size of 100), fitting the wine data on the Random Forest Regression model with the given number of leaves parameter, performing cross validation and calculating the cross-validation score. It is vital to note that I used used the optimal number of leaves value from the previous question. The model with the highest score is considered the best model and thus contains the optimal value for the number of trees, which in this case turned out to be around 800.

4.



The important features from correlation and Lasso from Assignment 6 were fixed acidity, volatile acidity, free sulfur dioxide, total sulfur dioxide, sulphates, and alcohol. This is like the bar graph above (calculated by Random Forest Regression) where the features mentioned are the ones with the most importance. However, free sulfur dioxide does seem to be less important than chlorides, one feature chosen by Random Forest Regression that is different from Lasso. Otherwise, both sets of features are similar.

The importance values of the features were calculated via creating a Random Forest Regression model via sklearn's RandomForestRegressor() class with the optimal trees and leaves values calculated in previous questions. I also split the wine data into train and test sets (75% and 25% respectively). I retrieved the importance of features via the feature_importances_ method and plotted the values via matplotlib.

5.

```
R-squared value for Random Forest Regressor Model: 0.3652494212653904.
MSE for Random Forest Regressor model: 0.4148690110698237

R-squared value for Linear Regression model: 0.3231496544789565
MSE value for Linear Regression model: 0.44228362796606624

R-squared value for KNN Regression model: 0.24897810554024846
MSE value for KNN Regression model: 0.49075056304788417
```

Compared to the Linear and KNN Regression model from Assignment 6, Random Forest Regression has the highest R-squared value and the lowest MSE value. Let us remember that R-squared shows how well the data fit the regression model. Hence a higher R-squared value indicates the model fits the data very well. MSE calculates how off the predicted values are from the true values. Hence a lower MSE value indicates the model predicts well. This comparison in performance highlights my decision of using Random Forest Regression on the wine dataset over Linear and KNN regression.

REFERENCE

[1] Avcontentteam, “PCA: What is Principal Component Analysis &amp; How It Works? (updated 2023),” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/ (accessed Dec. 1, 2023).

[2] H. Bonthu, “Single-link hierarchical clustering clearly explained!,” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/06/single-link-hierarchical-clustering-clearly-explained/ (accessed Dec. 2, 2023).

[3] Stephanie, “Hierarchical clustering / dendrogram: Simple definition, examples,” Statistics How To, https://www.statisticshowto.com/hierarchical-clustering/ (accessed Dec. 2, 2023).

[4] J. Brownlee, “A gentle introduction to uncertainty in machine learning,” MachineLearningMastery.com, https://machinelearningmastery.com/uncertainty-in-machine-learning/ (accessed Dec. 2, 2023).

[5] M. F. J. Steel, “Model averaging and its use in economics,” Journal of Economic Literature, https://www.aeaweb.org/articles?id=10.1257%2Fjel.20191385 (accessed Dec. 4, 2023).

[6] M. M. Baldi, E. Fersini, and E. Messina, “Relational bayesian model averaging for sentiment analysis in social networks,” SpringerLink, https://link.springer.com/chapter/10.1007/978-3-030-64583-0_27 (accessed Dec. 4, 2023).

[7] “Ensemble methods,” Corporate Finance Institute, https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/ (accessed Dec. 4, 2023).

[8] “Honeywell,” Wikipedia, https://en.wikipedia.org/wiki/Honeywell (accessed Dec. 4, 2023).

[9] “Caterpillar Inc..,” Wikipedia, https://en.wikipedia.org/wiki/Caterpillar_Inc. (accessed Dec. 4, 2023).

[10] “Procter &amp; Gamble,” Wikipedia, https://en.wikipedia.org/wiki/Procter_%26_Gamble (accessed Dec. 4, 2023).

[11] “Walmart,” Wikipedia, https://en.wikipedia.org/wiki/Walmart (accessed Dec. 4, 2023).