CS4641

CS-4641 Project Final Checkpoint: Human Age Prediction

Introduction/Background

Understanding age prediction based on health metrics is vital in fields like healthcare, medicine, and insurance. Recent advancements in machine learning have enabled more accurate analyses of complex datasets, revealing patterns related to aging and health.

Research indicates that various indicators can predict someone's biological age while providing insights into their developmental trends [2]. Furthermore, machine learning techniques have shown superior performance in estimating age from health data, enabling a deeper understanding of how these factors influence different age groups [1].

The data set for "Human Age Prediction" contains 3,000 rows and 25 features, including height, weight, and other lifestyle factors [3]. It is ideal for exploring relationships between health metrics and age, helping to identify trends for public health strategies.

Problem Definition

From 1900 to 2021, the average life expectancy of a newborn increased from 32 to 71 years [11]. This increment in human life expectancy causes the old age population to increase rapidly which then causes socio-economic burdens.

Mainly, using the aforementioned dataset, we pose a multi-class hard classification problem to categorize individuals into 10 age categories (Ages 18-25.2, ..., 81.8-89), enabling better assessment of trends allowing advances in healthcare and similar industries.

Methods

To pre-process the data, the following methods will be used:

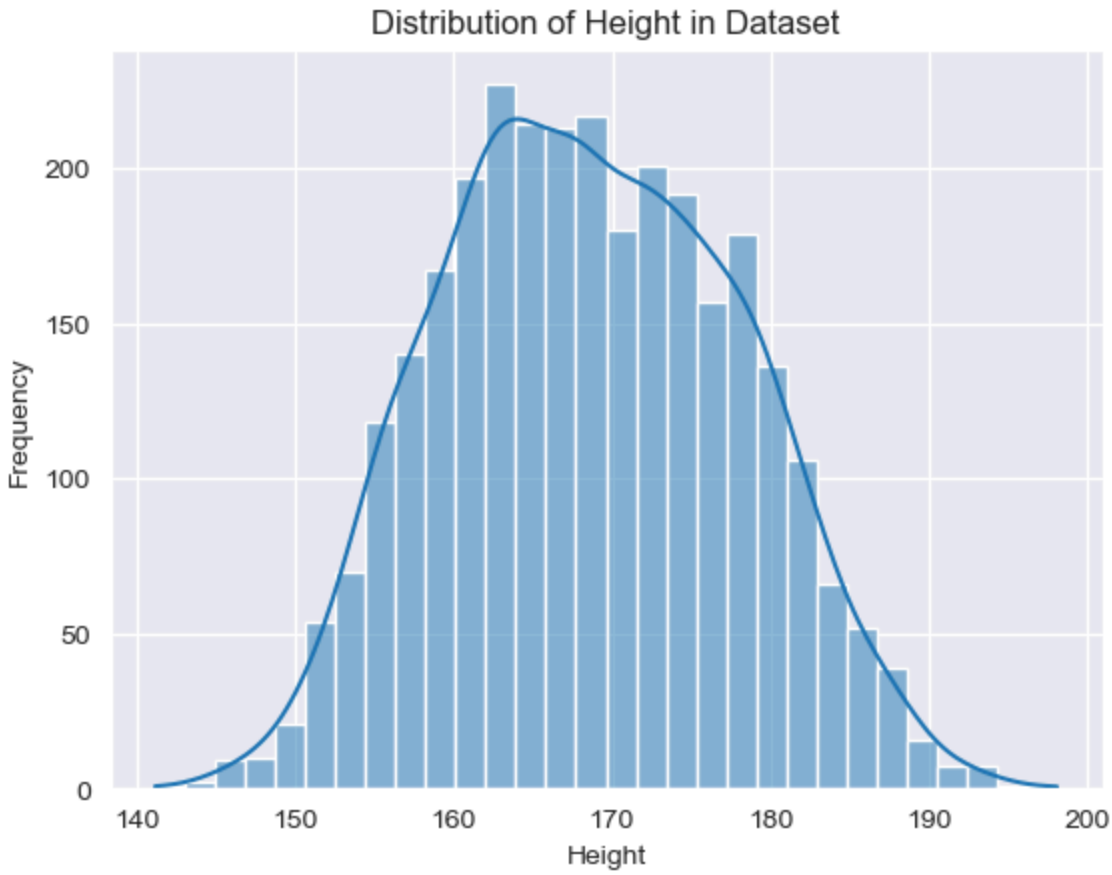
1. Of the 25 features in the dataset, 11 are in a categorical textual format, these will be numericized via **label encoding**. [8]

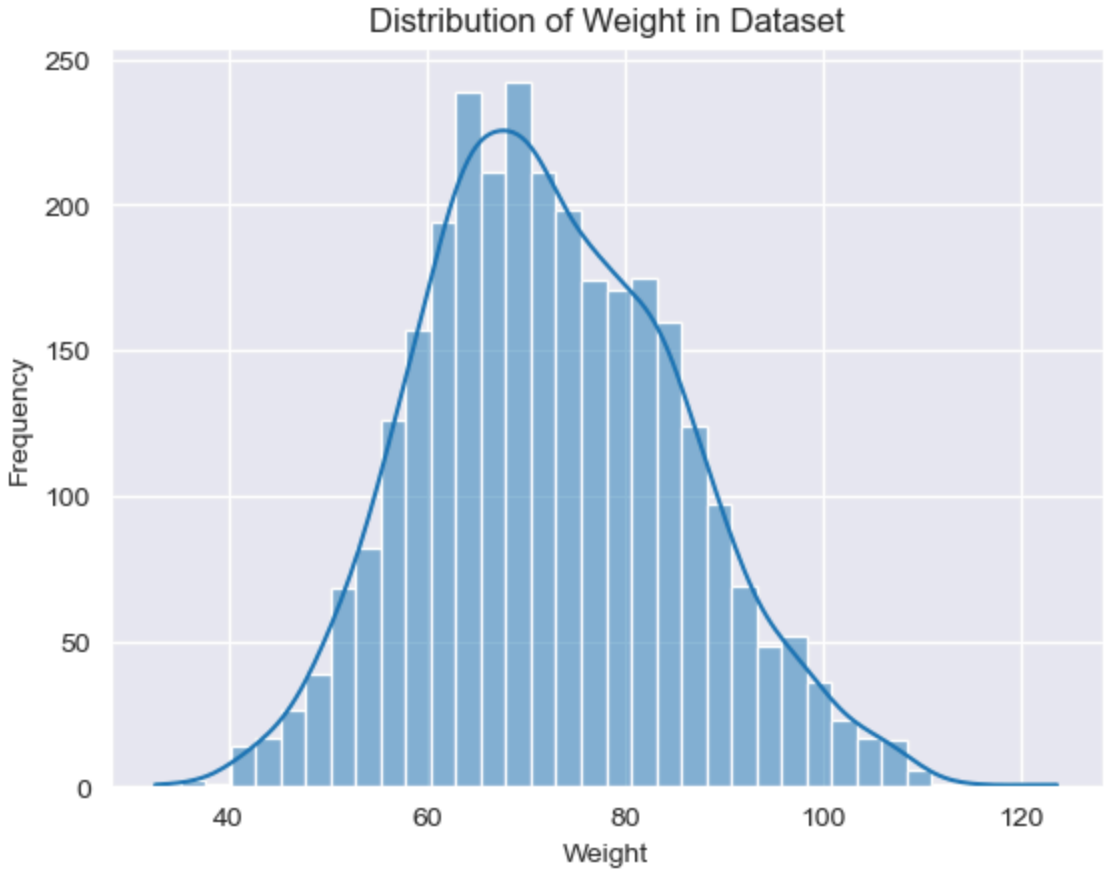
2. To identify any anomalies and explore the dataset, density based clustering algorithms such as **DB-Scan** will be used. This can help determine if any outliers exist in the dataset. Scikit Learn will be used for implementation. [5]
3. Furthermore, dimensionality reduction will likely be required as 25 features exist per data point, which can cause excessive computation during optimization as not all features might equally influence the label of the data point. **Principal Component Analysis (PCA)** via Scikit Learn will help achieve some dimensionality reduction [7].

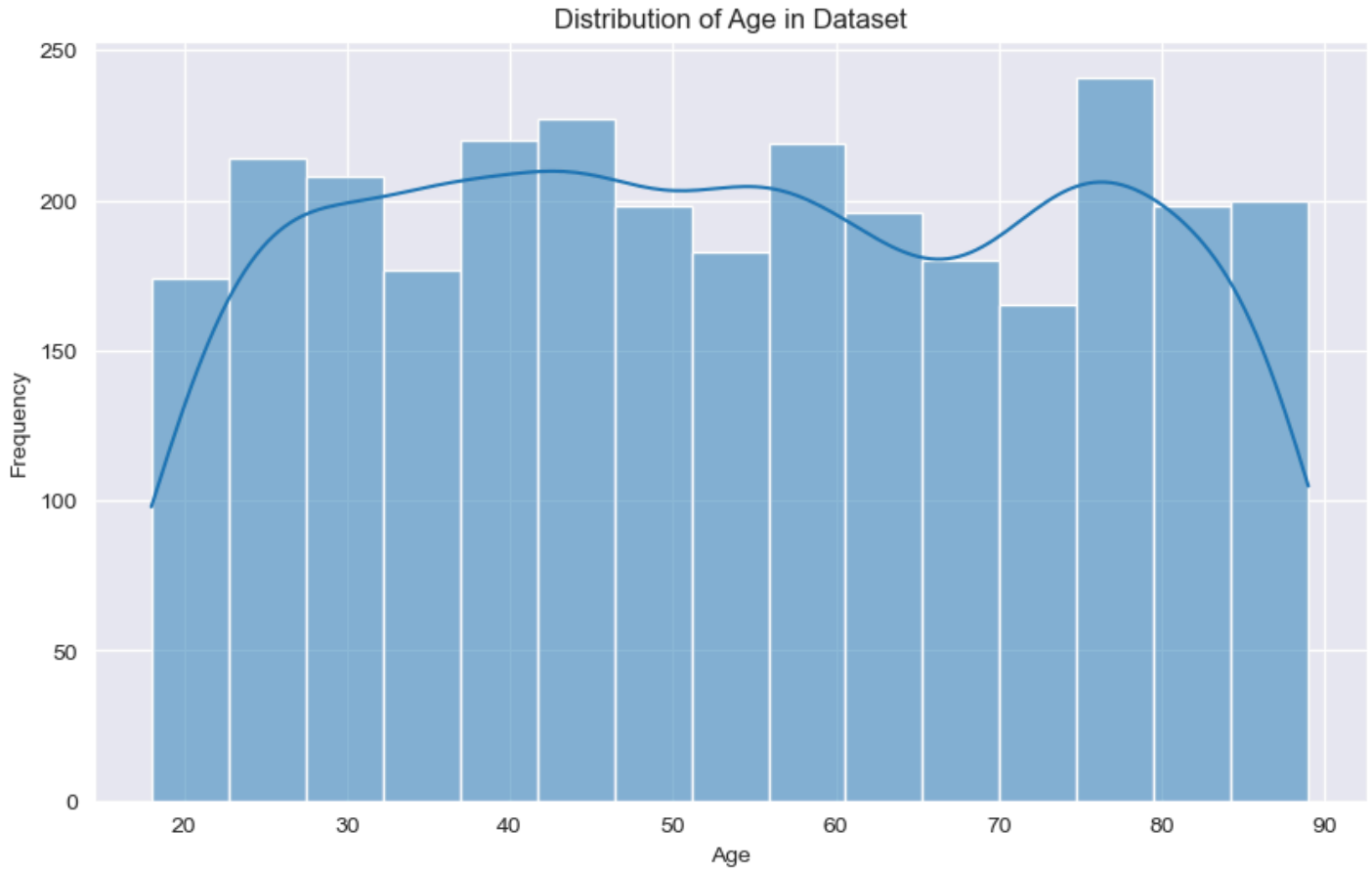
For solving the problem we propose the following methods:

1. **KNN**: KNN tends to be useful when data has some local structure pattern, which could be present in the age dataset, furthermore KNN could be less computationally complex during inference. This will be implemented using Scikit Learn [9].
2. **SVM**: SVM's tend to be useful with high dimensional data, which the age dataset has. Furthermore, it tends to be more robust to overfitting and supports versatile kernel functions. This will be implemented using Scikit Learn's SVC [10].
3. **Feed-Forward Neural Network**: A feed-forward neural network (two hidden layers, separated by ReLU activation, and a Softmax layer) can adapt to various nonlinearities and nuances in the data, potentially enabling more accurate analysis and inference on the data. This will be implemented manually through PyTorch.

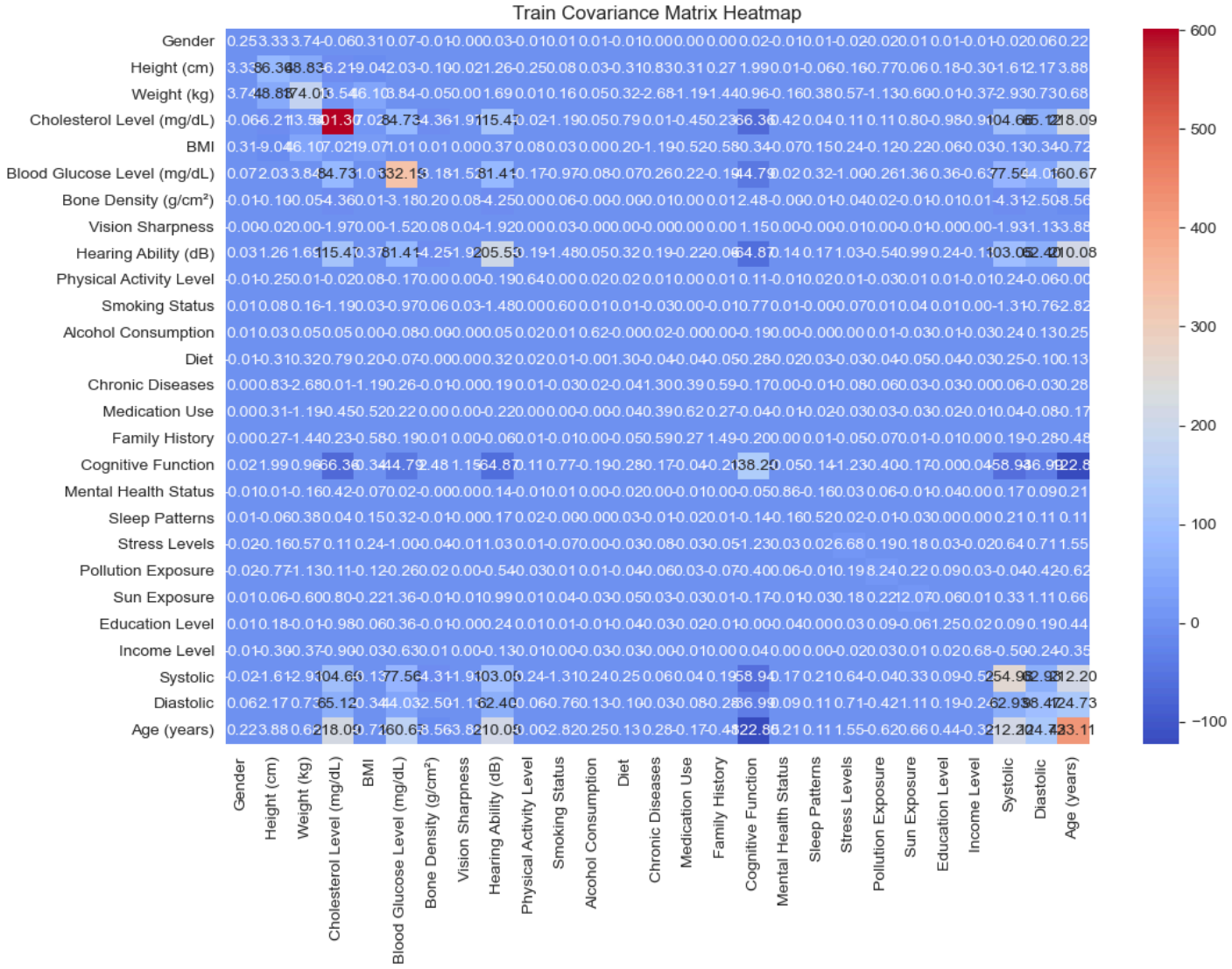
Data Preprocessing

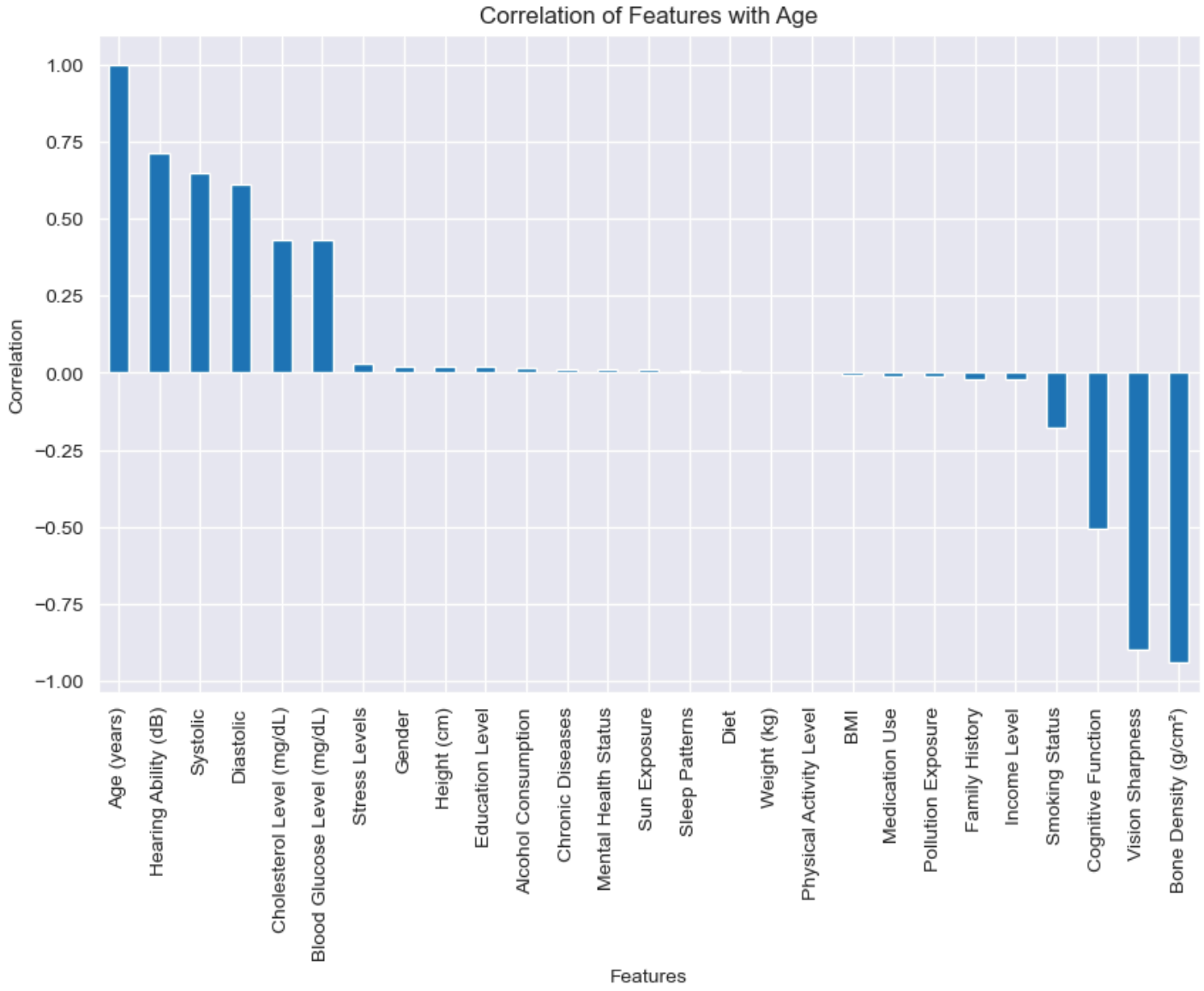

Distribution of Height in Dataset
Frequency
0
50
100
150
200
140
150
160
170
180
190
200
Height

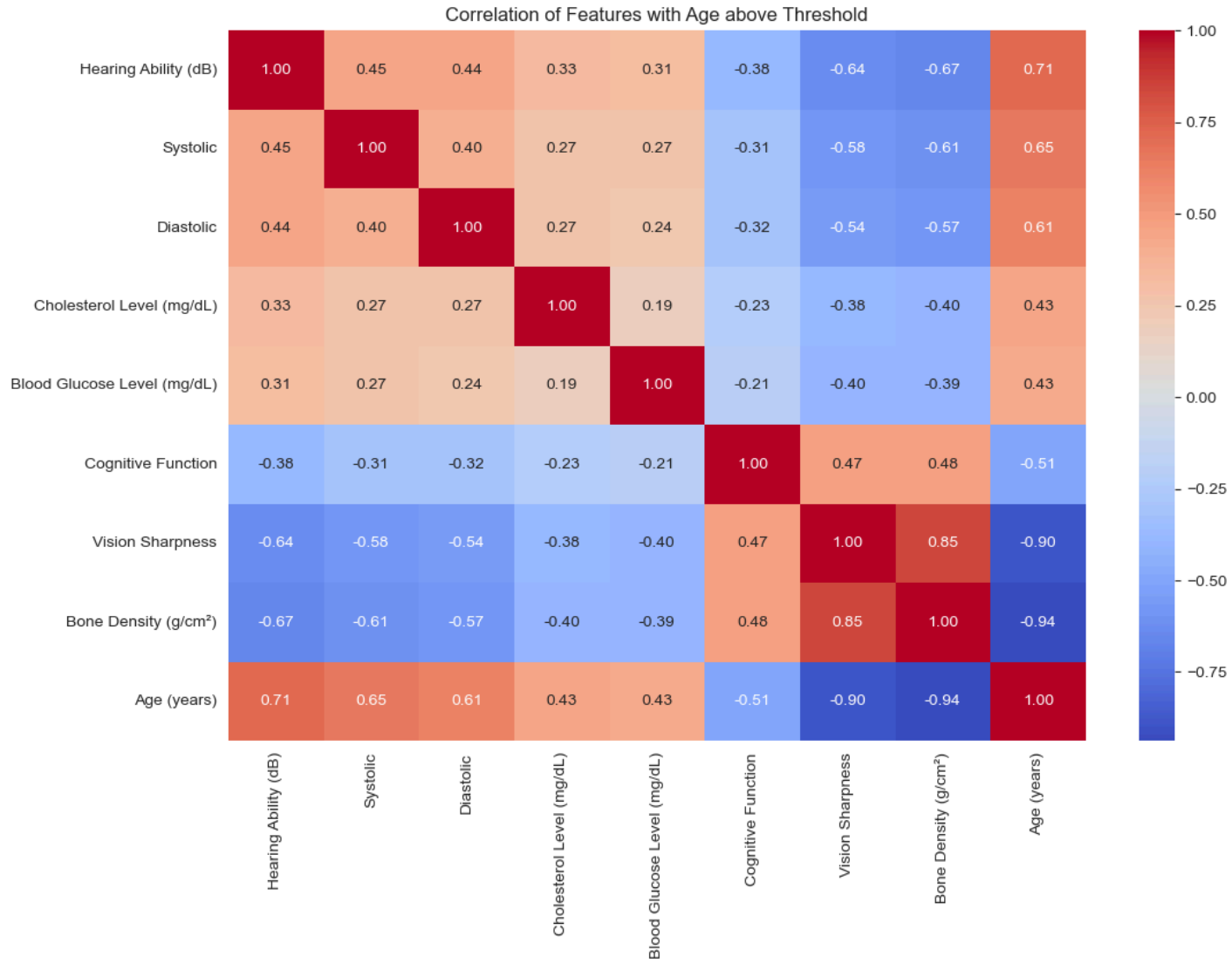

Distribution of Weight in Dataset
Frequency
0
50
100
150
200
250
40
60
80
100
120
Weight



The data is clean and more or less evenly balanced across age classes so we don't necessarily need to perform any synthetic data balancing. While there are some missing values, these are only in categorical features where no value is indicative of a field, and can be taken care of when numericizing the features. There is one feature however, 'Blood Pressure (s/d)' which is provided as a string, to encode this, we will split up the systolic and diastolic blood pressures into separate features. For categorical encoding, we will use Scikit Learn's LabelEncoder module.

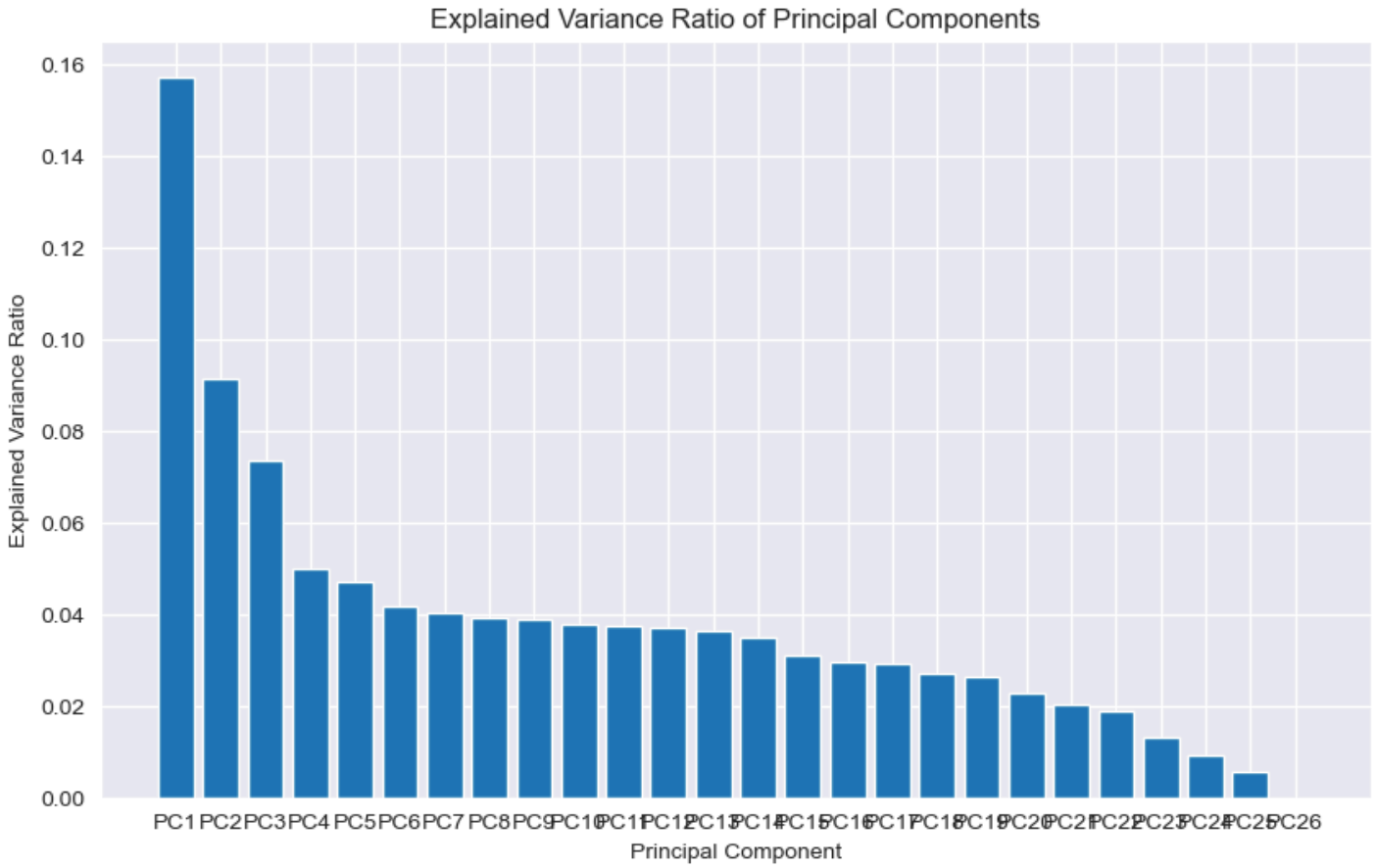

Train Covariance Matrix Heatmap
Gender
Height (cm)
Weight (kg)
Cholesterol Level (mg/dL)
BMI
Blood Glucose Level (mg/dL)
Bone Density (g/cm²)
Vision Sharpness
Hearing Ability (dB)
Physical Activity Level
Smoking Status
Alcohol Consumption
Diet
Chronic Diseases
Medication Use
Family History
Cognitive Function
Mental Health Status
Sleep Patterns
Stress Levels
Pollution Exposure
Sun Exposure
Education Level
Income Level
Systolic
Diastolic
Age (years)
600
500
400
300
200
100
0
−100

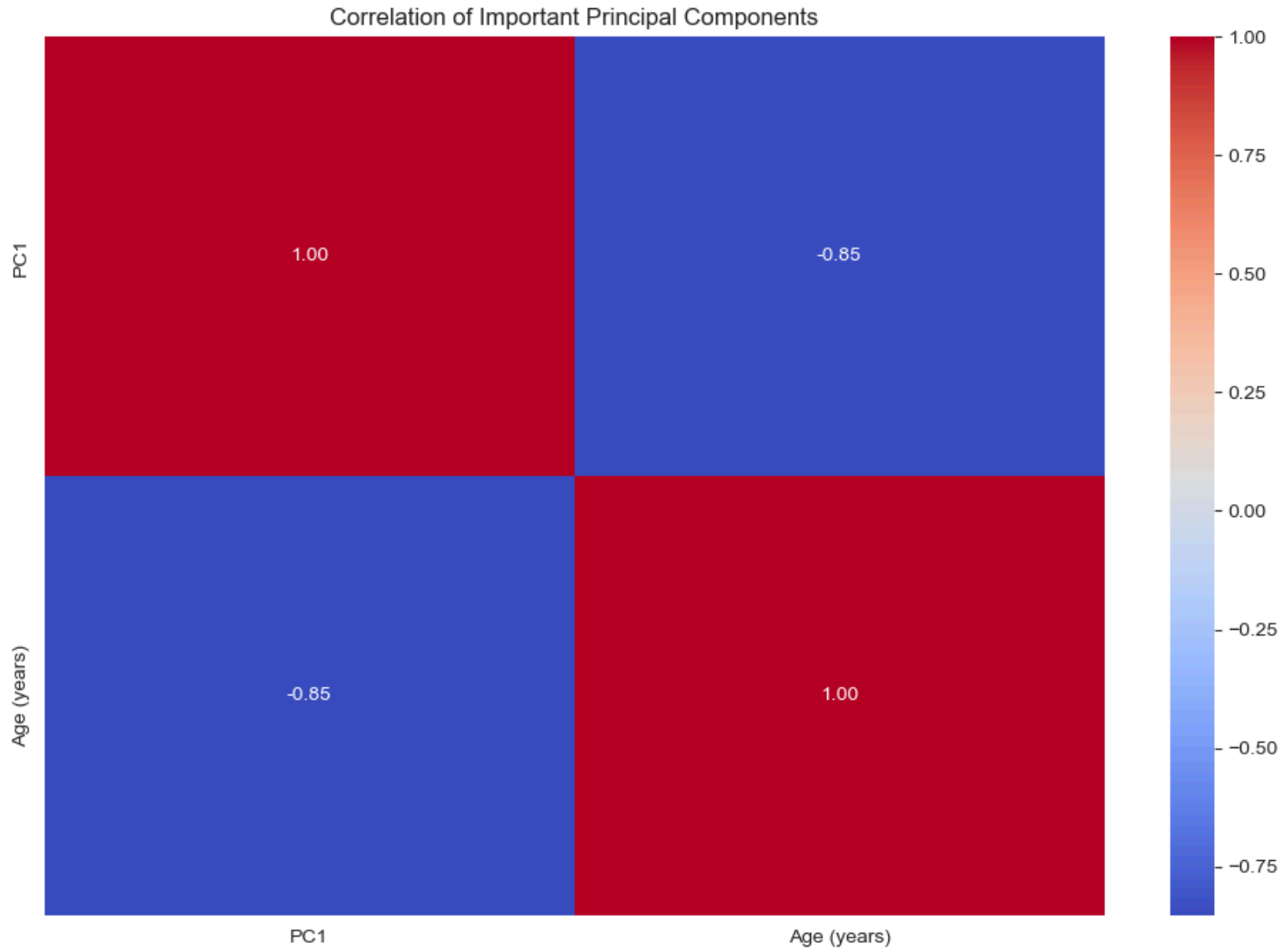

Correlation of Features with Age
Correlation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Age (years)
Hearing Ability (dB)
Systolic
Diastolic
Cholesterol Level (mg/dL)
Blood Glucose Level (mg/dL)
Stress Levels
Gender
Height (cm)
Education Level
Alcohol Consumption
Chronic Diseases
Mental Health Status
Sun Exposure
Sleep Patterns
Diet
Weight (kg)
Physical Activity Level
BMI
Medication Use
Pollution Exposure
Family History
Income Level
Smoking Status
Cognitive Function
Vision Sharpness
Bone Density (g/cm²)
Features



While the first covariance heatmap might be confusing, when plotting feature's correlation with the target variable (Age), we can see that some features are highly correlated with the target variable which leads to a smaller more useful heatmap. On the negative end, features like 'Bone Density', 'Vision Sharpness', and 'Cognitive Function' are highly correlated with the target variable, which more or less makes sense, as people age, they lose bone density, their vision gets blurrier, and their cognitive function reduces. On the positive end, features like 'Hearing Ability', 'Systolic', 'Diastolic', 'Cholesterol Level', and 'Blood Glucose' are highly correlated with the target variable, which also makes sense, as people age, they tend to have higher blood pressure, cholesterol, and blood glucose levels, and their hearing ability might also reduce.

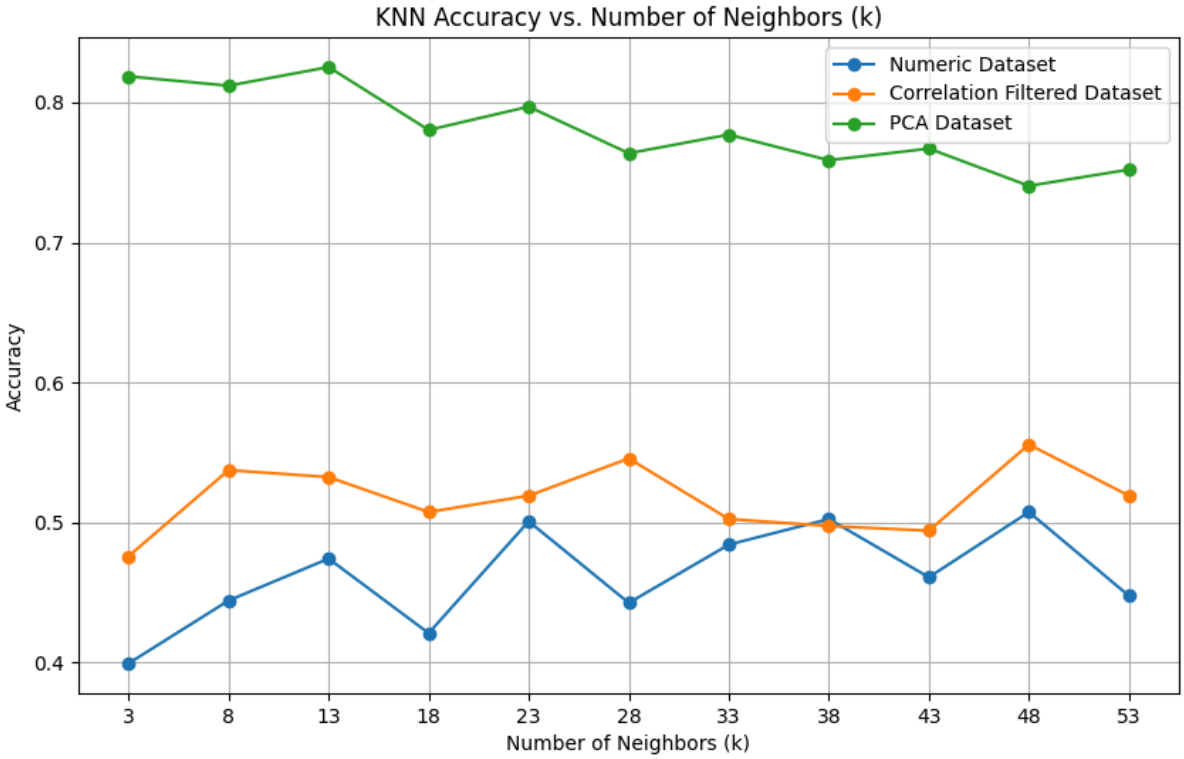
From the covariance heatmap, we can construct a dataset with features that have a correlation above a certain threshold, in this case, 0.3, to determine its impacts. Moreover, one of the good things we can determine from this, is that not all the features are incredibly influential or store the same amount of unique information about the target variable, which means that more features can be dropped using Principal Component Analysis (PCA). As seen, features from the original covariance heatmap have been dropped leaving a simpler and easier to read heatmap.

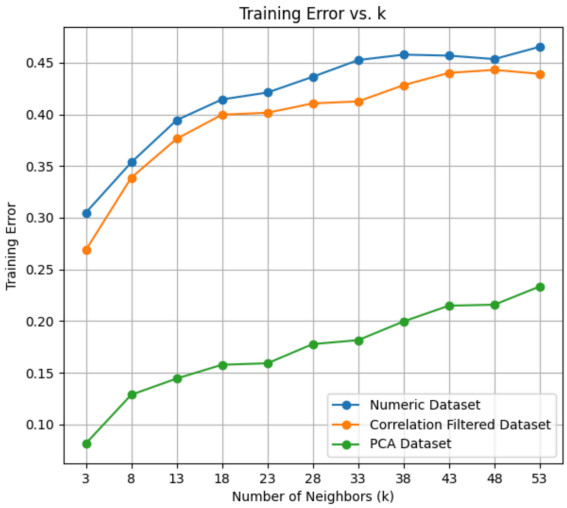

Explained Variance Ratio of Principal Components
Explained Variance Ratio
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9
Principal Component

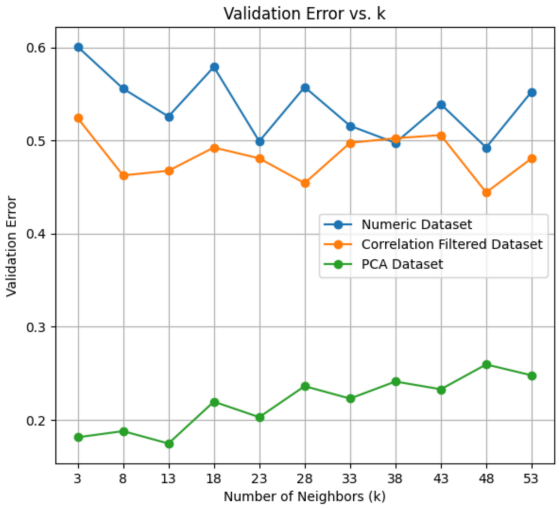


Applying PCA to the dataset, we are able to drastically reduce the size of the dataset while still retaining the most important features. In this case, only four principal components had an explained variance ratio of over 5%, which is much smaller than the original dataset which contained 25 features, and the correlation filtered dataset which contained 8 features. Furthermore, we create a second dataset with only the two most principal components for data visualization purposes.

Results and Discussion - KNN Visualizations and Metrics







Numeric and Filtered Accuracies

- Both datasets exhibit a lack of optimism in their predictive capabilities, suggesting a need for further feature engineering or alternative modeling approaches.

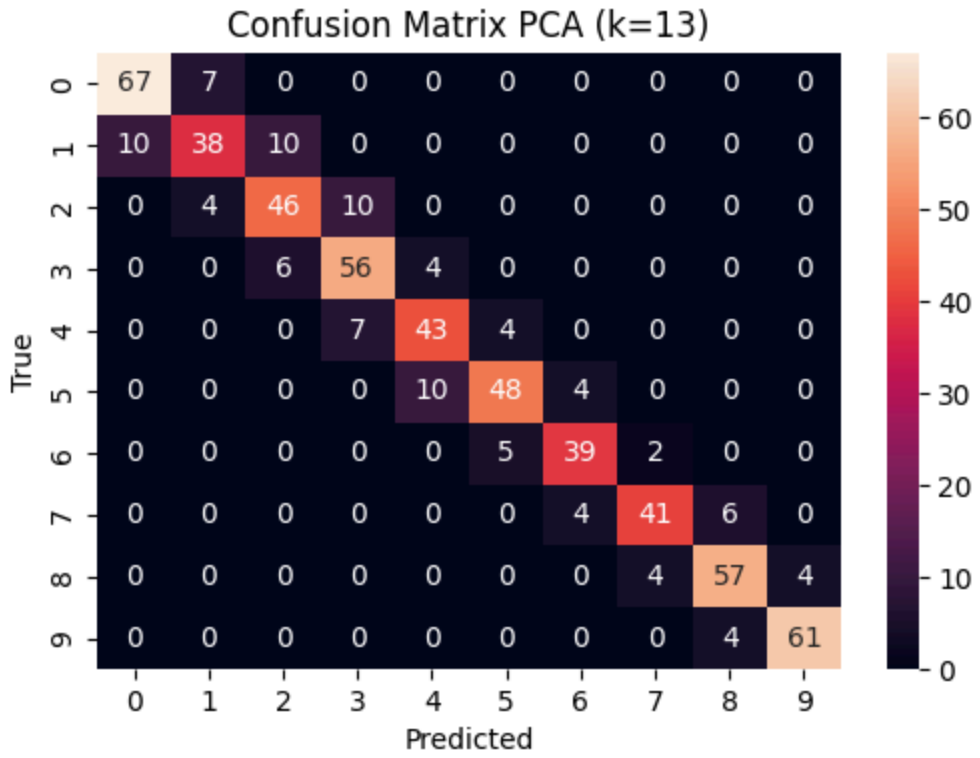
PCA Dataset Performance

- The PCA dataset demonstrates a large performance boost, with accuracies ranging from much higher across all k-values. This significant improvement highlights the effectiveness of dimensionality reduction in enhancing model accuracy and capturing essential data features.

Performance Stability

- When iterating through the k-values, the numeric and filtered datasets show minimal variation in performance. This stability at lower accuracy levels suggests that neither dataset is particularly sensitive to changes in k.
- While the PCA dataset maintains high accuracy overall, there is a gradual decline in performance as k increases, particularly at higher k-values. This decline might indicate that as k increases, the model may start to generalize too much, potentially leading to loss of specificity in predictions.

PCA Confusion Matrix and Metrics



- Accuracy: 0.8252911813643927
- Validation Error: 0.17470881863560728
- Cross Validation Mean: 0.7970587566768951
- Mean Absolute Error: 0.17470881863560733
- R2: 0.9801487061819769
- F1: 0.8245191619880474

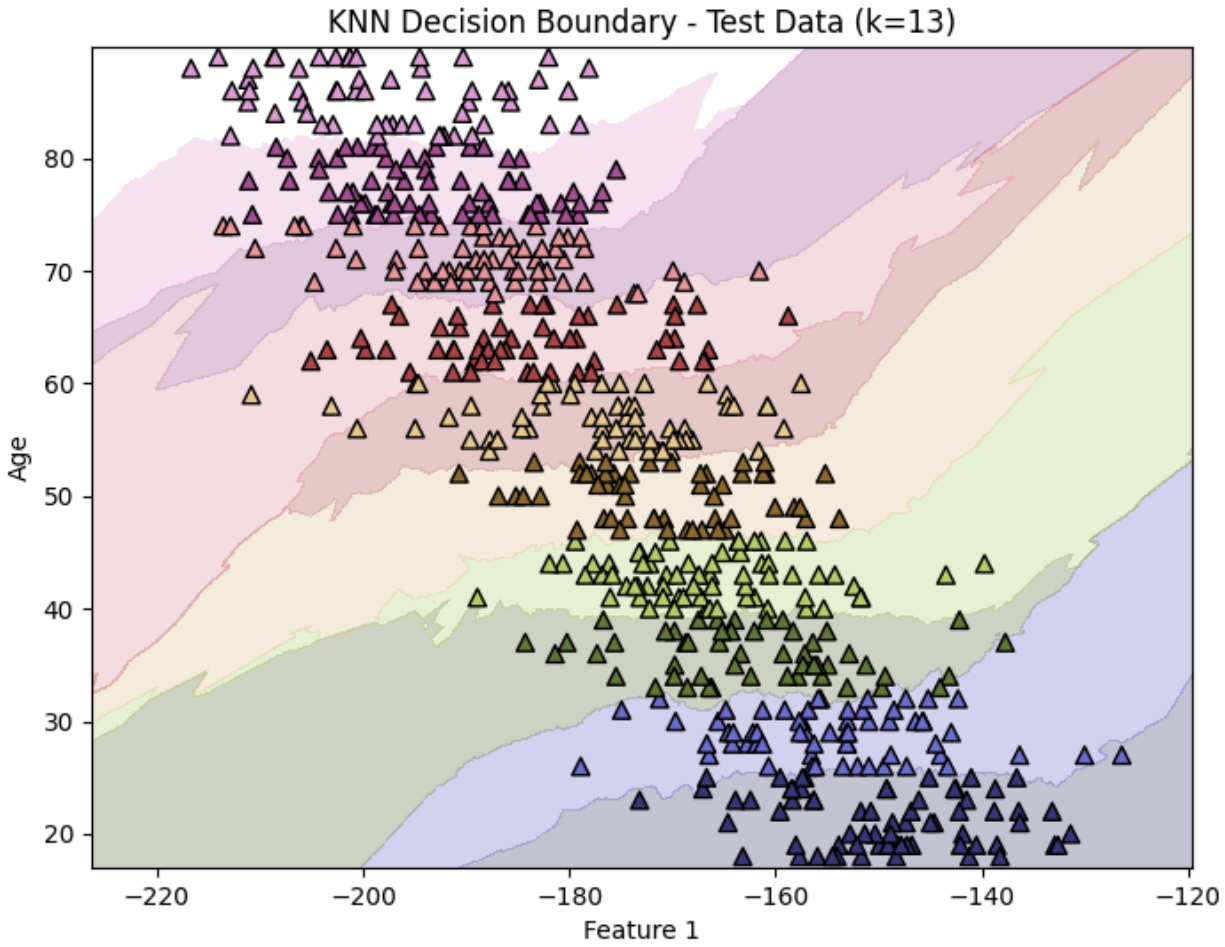
PCA

- As expected the PCA dataset demonstrates much better classification capabilities, confirming that dimensionality reduction and noise filtering has led to a more enhanced KNN

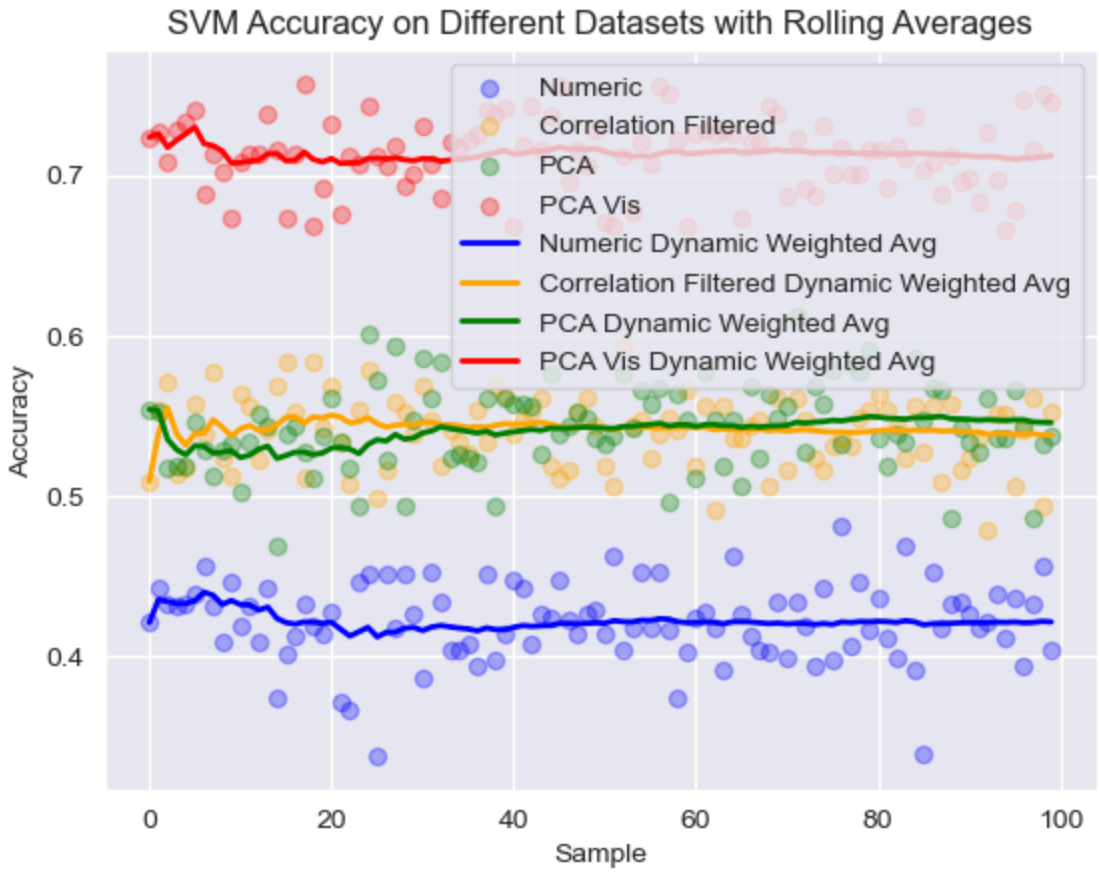
model.

- Minimal misclassifications are shown in the Confusion Matrix.
- Numeric and filtered datasets' poor performance might be explained by the curse of dimensionality, where the distance between points will become less meaningful as there are more features. This makes it harder for KNN to find relevant neighbors ultimately leading to a lower accuracy.

Decision boundaries of KNN model using 1 principal component to predict age



Results and Discussion - SVM Visualizations and Metrics (RBF Kernel)



Numeric Metrics

- Accuracy Mean: 0.421514143094842 STD: 0.024932937646870136
- Precision Mean: 0.3712622598667418 STD: 0.06156750821590125
- Recall Mean: 0.421514143094842 STD: 0.024932937646870136
- F1 Score Mean: 0.34344042399099045 STD: 0.03466794555986116
- Mean Absolute Error Mean: 0.6856572379367722 STD: 0.0357877772908454
- R2 Score Mean: 0.8923948713637677 STD: 0.009141230322467267

Correlation Filtered Metrics

- Accuracy Mean: 0.5401497504159734 STD: 0.02235279149900984
- Precision Mean: 0.548707792642991 STD: 0.03166563958770604
- Recall Mean: 0.5401497504159734 STD: 0.02235279149900984
- F1 Score Mean: 0.520188720537152 STD: 0.03064315938965716
- Mean Absolute Error Mean: 0.48905158069883525 STD: 0.025553287591348624
- R2 Score Mean: 0.9349458719836702 STD: 0.005235336338637109

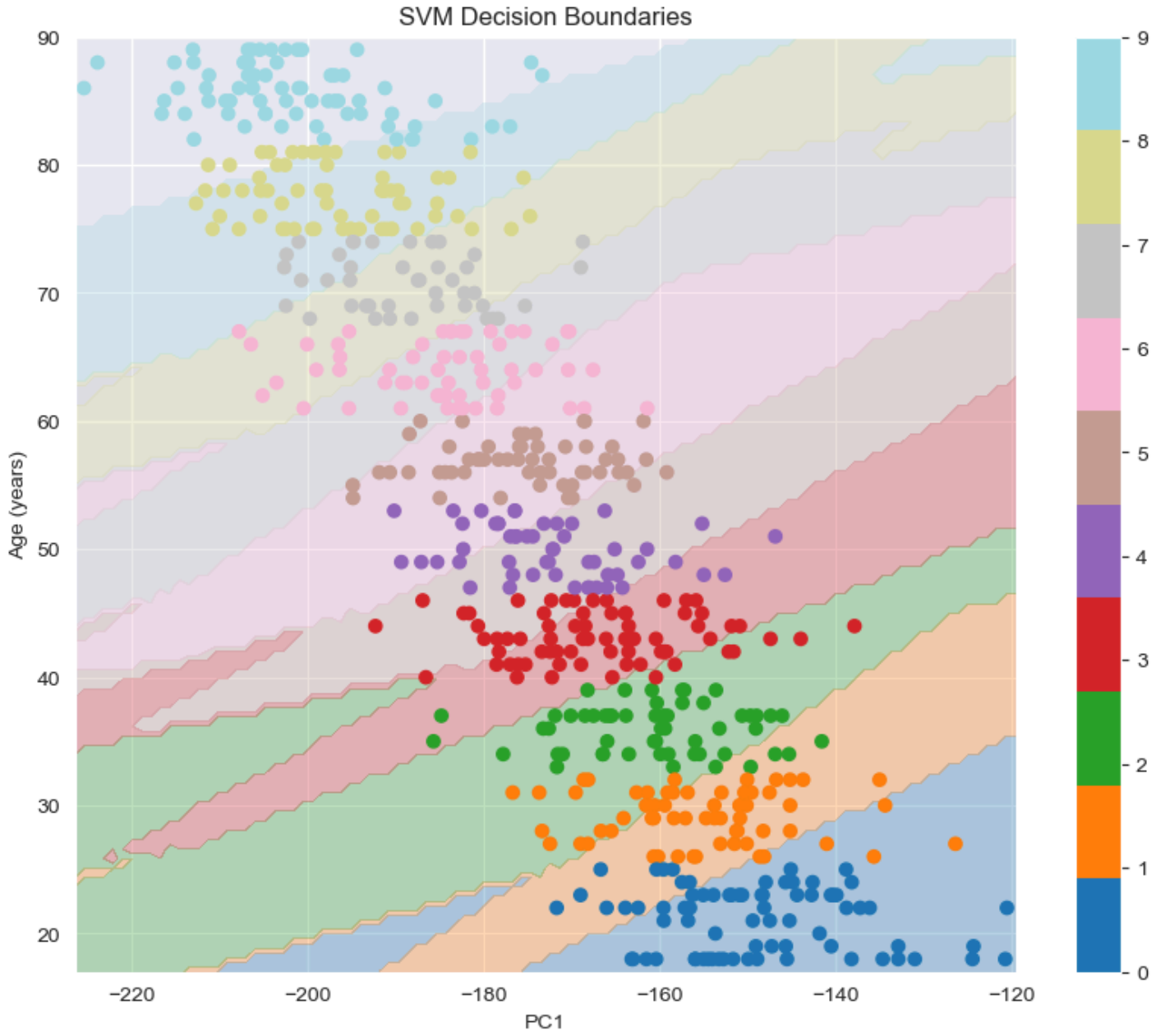
PCA Metrics

- Accuracy Mean: 0.5430782029950083 STD: 0.026561802109686464
- Precision Mean: 0.5143382336263218 STD: 0.06123127347228393
- Recall Mean: 0.5430782029950083 STD: 0.026561802109686464
- F1 Score Mean: 0.48781074627330673 STD: 0.04321788451012288
- Mean Absolute Error Mean: 0.477287853577371 STD: 0.028819841232423576
- R2 Score Mean: 0.9385490892683864 STD: 0.004610513622573076

PCA Vis Metrics

- Accuracy Mean: 0.7128618968386022 STD: 0.02265616329048303
- Precision Mean: 0.7256919792583584 STD: 0.021584669098429588
- Recall Mean: 0.7128618968386022 STD: 0.02265616329048303
- F1 Score Mean: 0.7095098893179292 STD: 0.024303990796568163
- Mean Absolute Error Mean: 0.2891680532445924 STD: 0.022888420191199212
- R2 Score Mean: 0.9650483110437378 STD: 0.003269671239720684

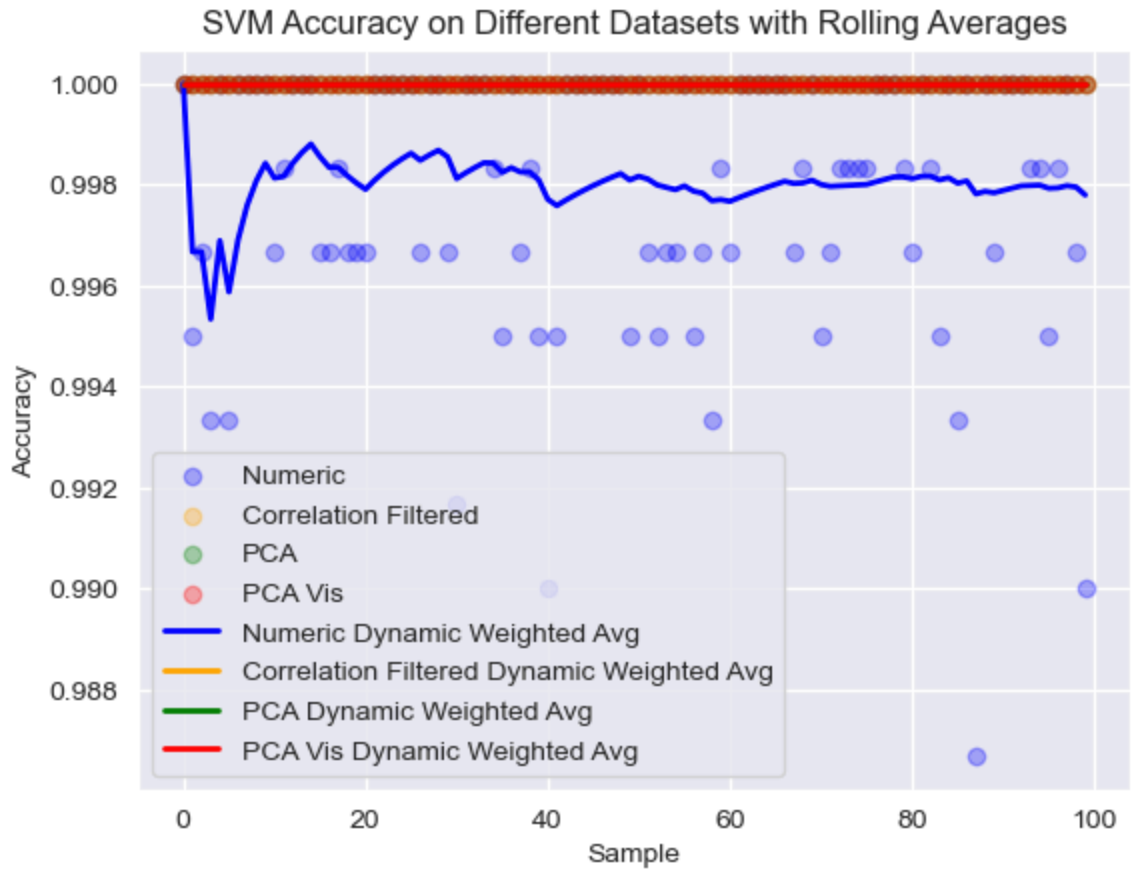
Decision Boundaries



Analysis

- The numeric dataset performed poorly through the SVM evaluation where accuracy and precision were low and MAE was high
- This is likely due to the fact that the numeric dataset has more features that can confuse or are not as influential to the target variable, and the SVM model is not able to effectively classify the data.

Results and Discussion - SVM Visualizations and Metrics (Linear Kernel)



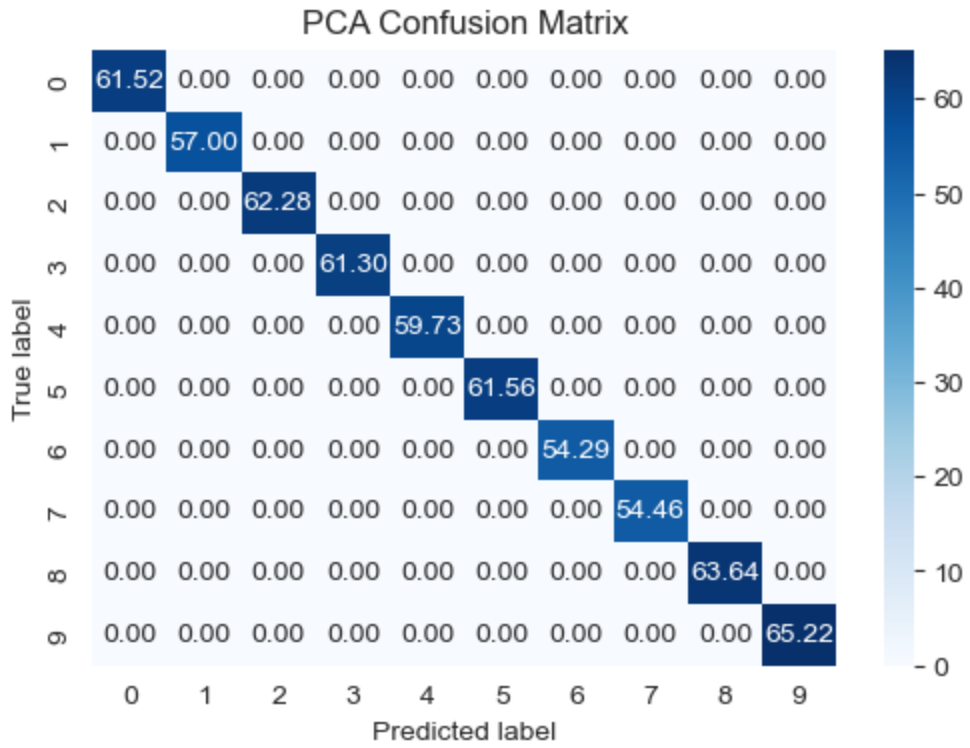
Numeric Metrics

- Accuracy Mean: 0.5747420965058238 STD: 0.018833892813391425
- Precision Mean: 0.5826977917204026 STD: 0.02157054373282149
- Recall Mean: 0.5747420965058238 STD: 0.018833892813391425
- F1 Score Mean: 0.5688542807298379 STD: 0.020948159667904594
- Mean Absolute Error Mean: 0.4485856905158069 STD: 0.02117136952401662
- R2 Score Mean: 0.941277167762021 STD: 0.004078180236735414

Correlation Filtered Metrics

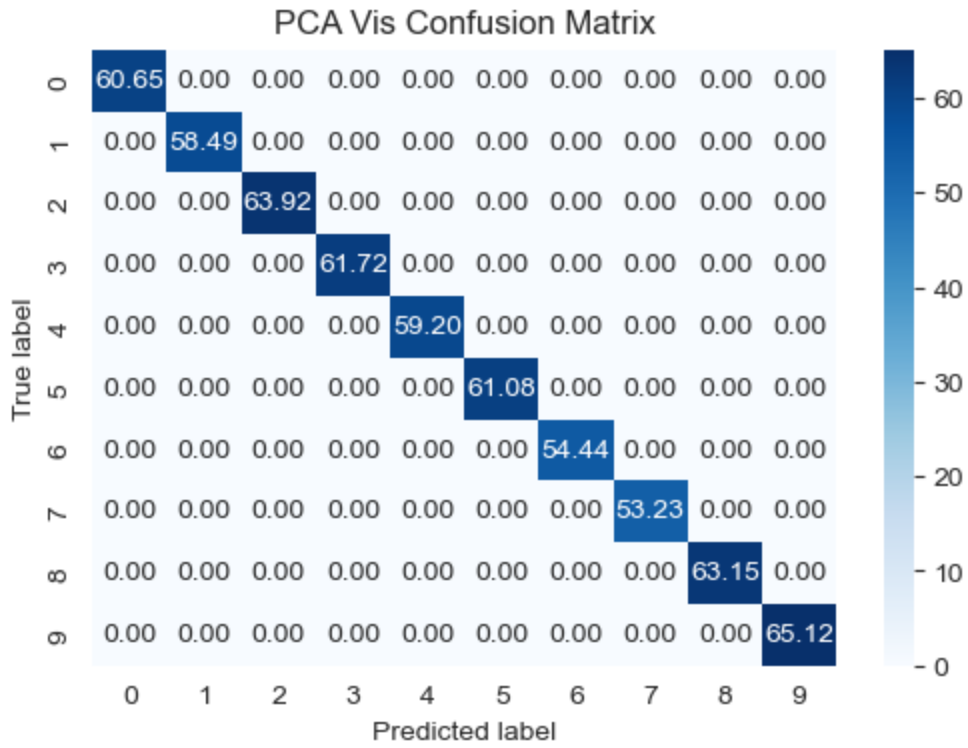
- Accuracy Mean: 0.9006821963394342 STD: 0.013157800139806433
- Precision Mean: 0.9035277521003486 STD: 0.012656360491929936
- Recall Mean: 0.9006821963394342 STD: 0.013157800139806433
- F1 Score Mean: 0.9008337053465385 STD: 0.013146264639472627
- Mean Absolute Error Mean: 0.09931780366056571 STD: 0.013157800139806443
- R2 Score Mean: 0.9881240699262489 STD: 0.001688985631218321

PCA Metrics



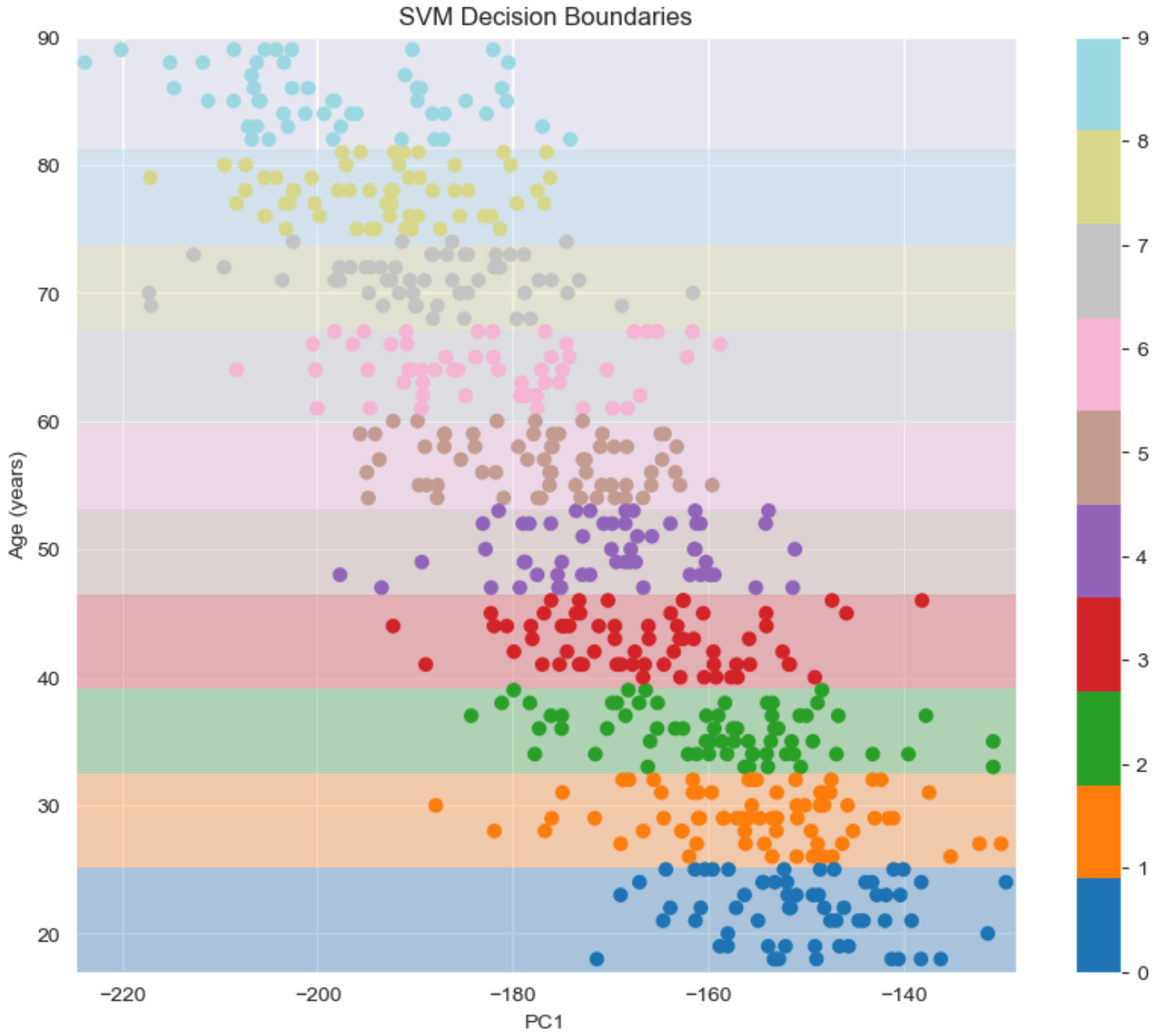
- Accuracy Mean: 0.6070382695507489 STD: 0.019368542388972825
- Precision Mean: 0.6151490171602801 STD: 0.021599230216888116
- Recall Mean: 0.6070382695507489 STD: 0.019368542388972825
- F1 Score Mean: 0.5991263383617028 STD: 0.022554808802885476
- Mean Absolute Error Mean: 0.41412645590682184 STD: 0.021635600347186132
- R2 Score Mean: 0.9456091640806926 STD: 0.004186925373694726

PCA Vis Metrics



- Accuracy Mean: 0.8263560732113145 STD: 0.01594652484979123
- Precision Mean: 0.831355887696339 STD: 0.015667055361420604
- Recall Mean: 0.8263560732113145 STD: 0.01594652484979123
- F1 Score Mean: 0.826205020494002 STD: 0.01601169505716374
- Mean Absolute Error Mean: 0.17439267886855245 STD: 0.01615039017772788
- R2 Score Mean: 0.9789961012434055 STD: 0.0020714950806615486

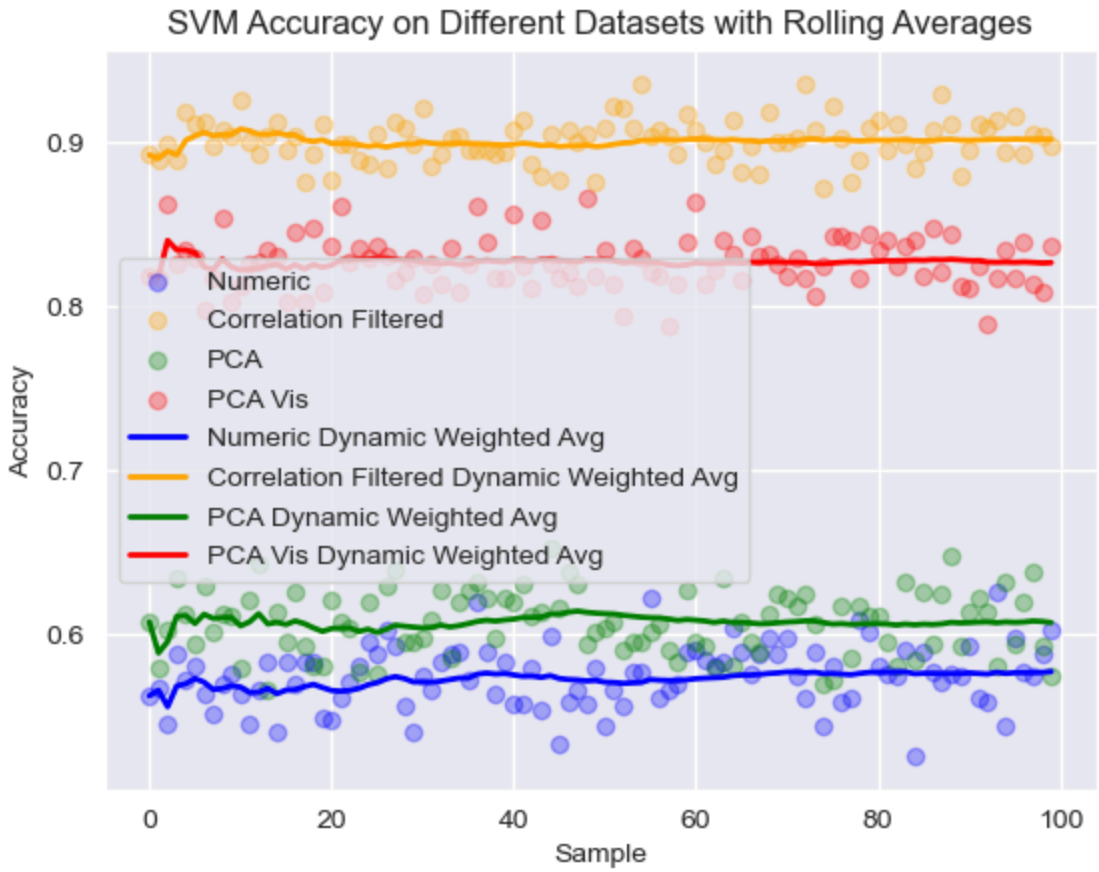
Decision Boundaries



Analysis

- Surprisingly the SVM model with the linear kernel performed the best across all datasets. The accuracy of the SVM model with the linear kernel was consistently higher than the other kernels, and had 100% accuracy with the correlation filtered and pca datasets over 100 samples, which is surprisingly good and achieves all target goals.
- There is no real explanation for why the linear kernel performed the best aside, from the fact that some features of the data may be linearly separable, and the linear kernel is able to effectively classify the data, more so than a polynomial, rbf, or sigmoid kernel.

Results and Discussion - SVM Visualizations and Metrics (Polynomial Kernel)



Numeric Metrics

- Accuracy Mean: 0.5747420965058238 STD: 0.018833892813391425
- Precision Mean: 0.5826977917204026 STD: 0.02157054373282149
- Recall Mean: 0.5747420965058238 STD: 0.018833892813391425
- F1 Score Mean: 0.5688542807298379 STD: 0.020948159667904594
- Mean Absolute Error Mean: 0.4485856905158069 STD: 0.02117136952401662
- R2 Score Mean: 0.941277167762021 STD: 0.004078180236735414

Correlation Filtered Metrics

- Accuracy Mean: 0.9006821963394342 STD: 0.013157800139806433
- Precision Mean: 0.9035277521003486 STD: 0.012656360491929936
- Recall Mean: 0.9006821963394342 STD: 0.013157800139806433
- F1 Score Mean: 0.9008337053465385 STD: 0.013146264639472627
- Mean Absolute Error Mean: 0.09931780366056571 STD: 0.013157800139806443
- R2 Score Mean: 0.9881240699262489 STD: 0.001688985631218321

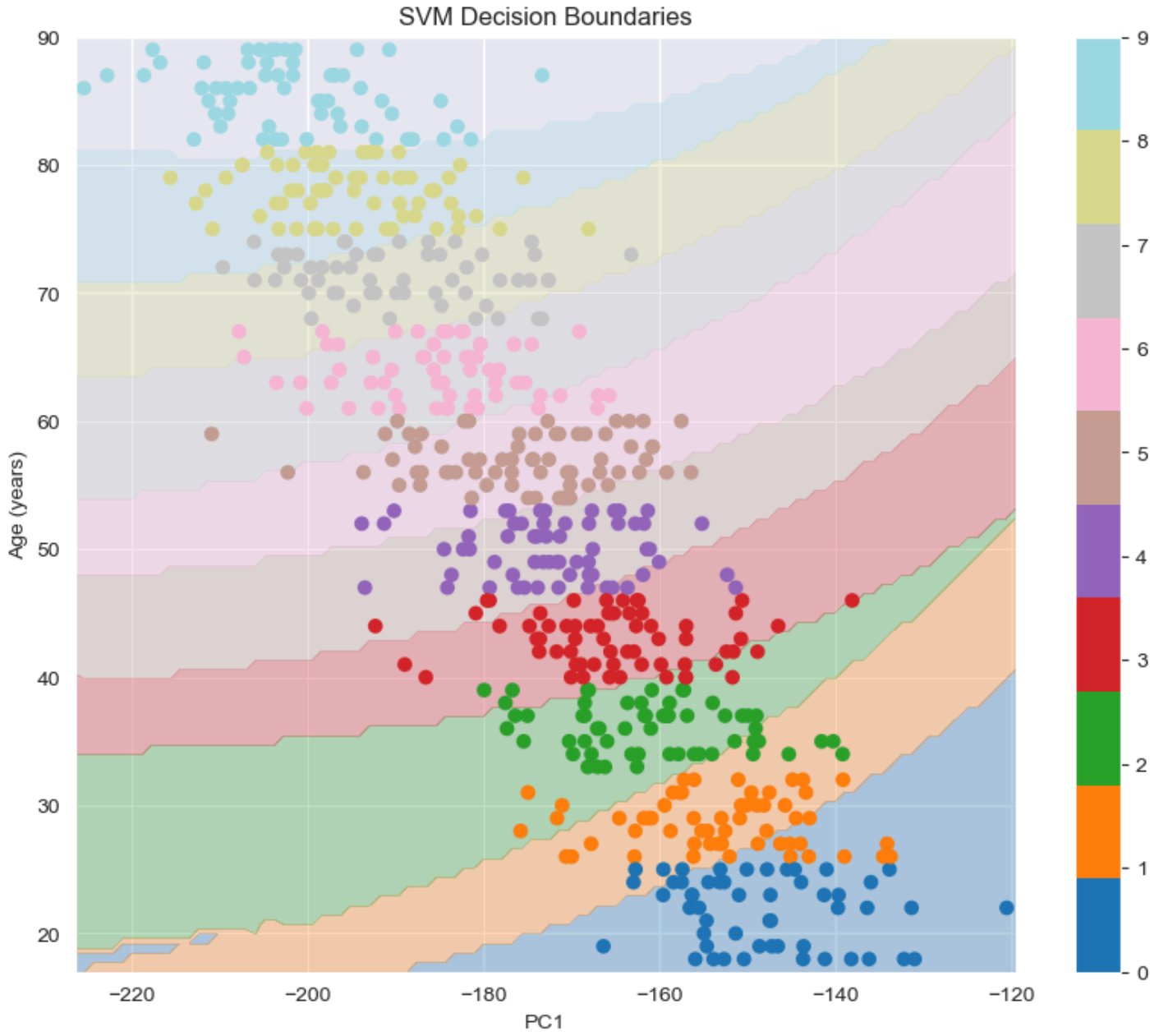
PCA Metrics

- Accuracy Mean: 0.6070382695507489 STD: 0.019368542388972825
- Precision Mean: 0.6151490171602801 STD: 0.021599230216888116
- Recall Mean: 0.6070382695507489 STD: 0.019368542388972825
- F1 Score Mean: 0.5991263383617028 STD: 0.022554808802885476
- Mean Absolute Error Mean: 0.41412645590682184 STD: 0.021635600347186132
- R2 Score Mean: 0.9456091640806926 STD: 0.004186925373694726

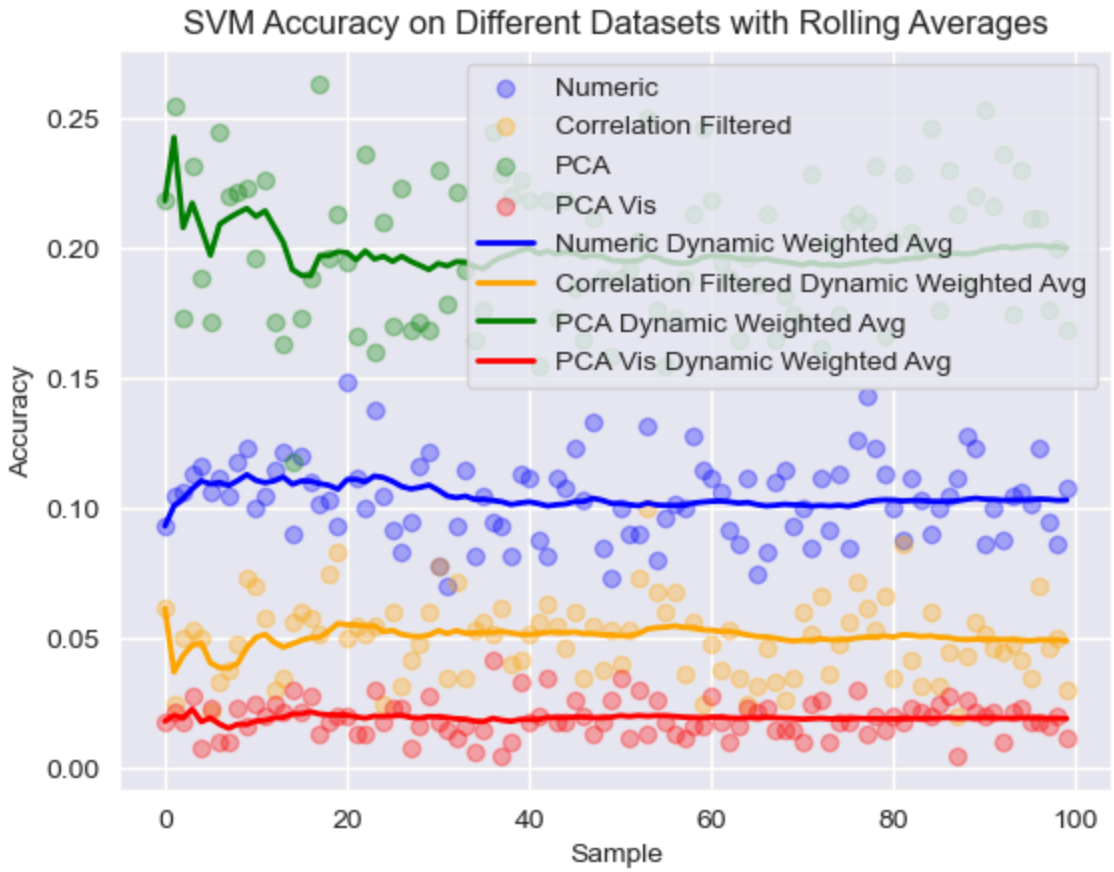
PCA Vis Metrics

- Accuracy Mean: 0.8263560732113145 STD: 0.01594652484979123
- Precision Mean: 0.831355887696339 STD: 0.015667055361420604
- Recall Mean: 0.8263560732113145 STD: 0.01594652484979123
- F1 Score Mean: 0.826205020494002 STD: 0.01601169505716374
- Mean Absolute Error Mean: 0.17439267886855245 STD: 0.01615039017772788
- R2 Score Mean: 0.9789961012434055 STD: 0.0020714950806615486

Decision Boundaries



Results and Discussion - SVM Visualizations and Metrics (Sigmoid Kernel)



Numeric Metrics

- Accuracy Mean: 0.10376039933444259 STD: 0.015559886621615929
- Precision Mean: 0.03628702663036114 STD: 0.030776847094765084
- Recall Mean: 0.10376039933444259 STD: 0.015559886621615929
- F1 Score Mean: 0.038795621434817795 STD: 0.011498950840730266
- Mean Absolute Error Mean: 3.4621131447587357 STD: 0.46144971183908773
- R2 Score Mean: -1.182155315014982 STD: 0.577052429616782

Correlation Filtered Metrics

- Accuracy Mean: 0.04988352745424292 STD: 0.015125844494589287
- Precision Mean: 0.009108907239750122 STD: 0.004464819675586144
- Recall Mean: 0.04988352745424292 STD: 0.015125844494589287
- F1 Score Mean: 0.014069885837160512 STD: 0.004863581451480146
- Mean Absolute Error Mean: 5.087903494176373 STD: 0.6036460755048091
- R2 Score Mean: -2.962409213719724 STD: 0.7459943432708384

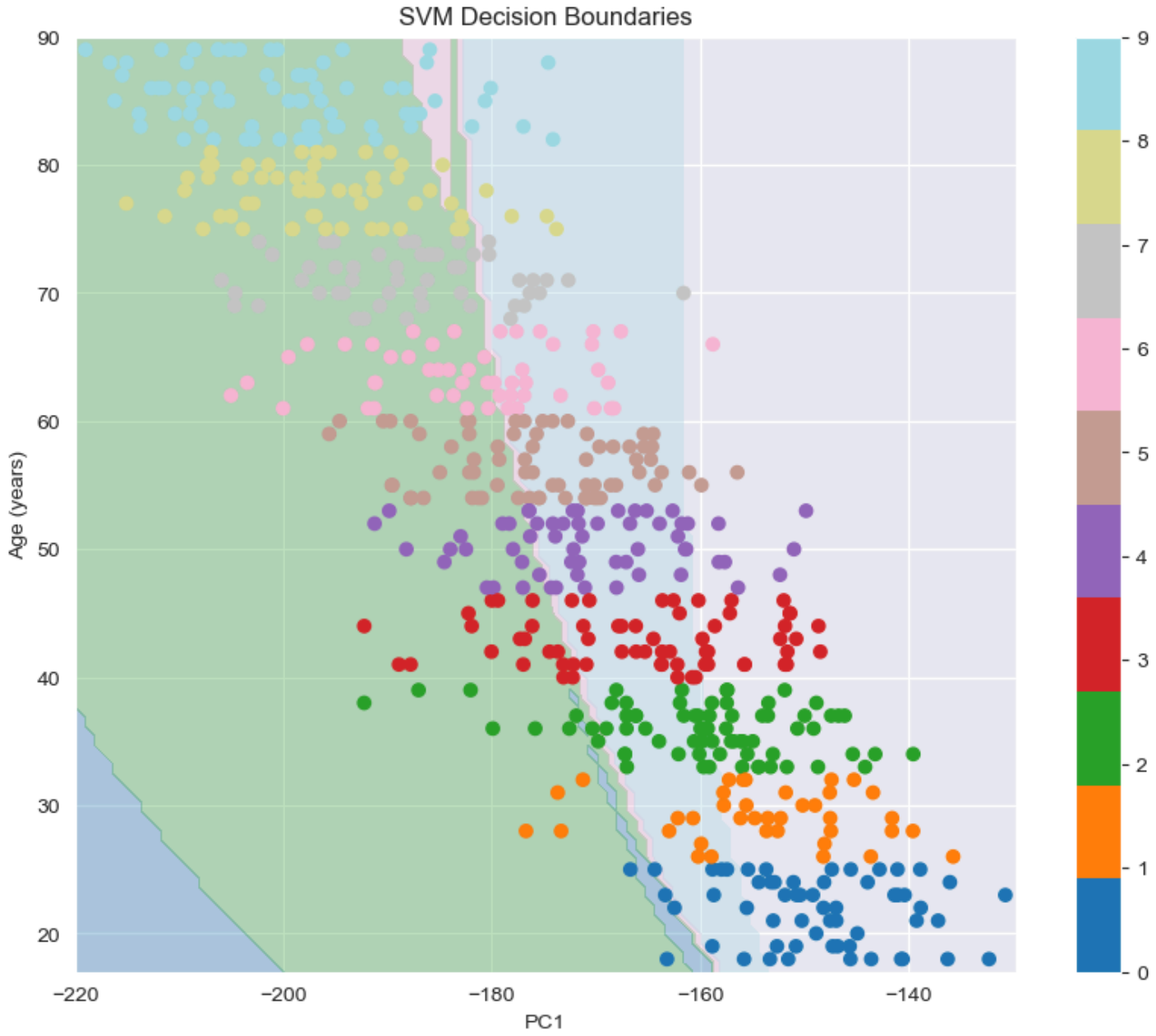
PCA Metrics

- Accuracy Mean: 0.19905158069883527 STD: 0.02810616589971061
- Precision Mean: 0.11562925120334688 STD: 0.043996164968679655
- Recall Mean: 0.19905158069883527 STD: 0.02810616589971061
- F1 Score Mean: 0.10279303373407755 STD: 0.025515358165831132
- Mean Absolute Error Mean: 2.07738768718802 STD: 0.2224358034624756
- R2 Score Mean: 0.09911786635322738 STD: 0.1825011089038412

PCA Vis Metrics

- Accuracy Mean: 0.01940099833610649 STD: 0.006849180720849276
- Precision Mean: 0.006468709108867939 STD: 0.011689634156619229
- Recall Mean: 0.01940099833610649 STD: 0.006849180720849276
- F1 Score Mean: 0.007544790113203495 STD: 0.004181693191204887
- Mean Absolute Error Mean: 5.888868552412646 STD: 0.23442391244021965
- R2 Score Mean: -3.7995563359579245 STD: 0.3194514577739988

Decision Boundaries

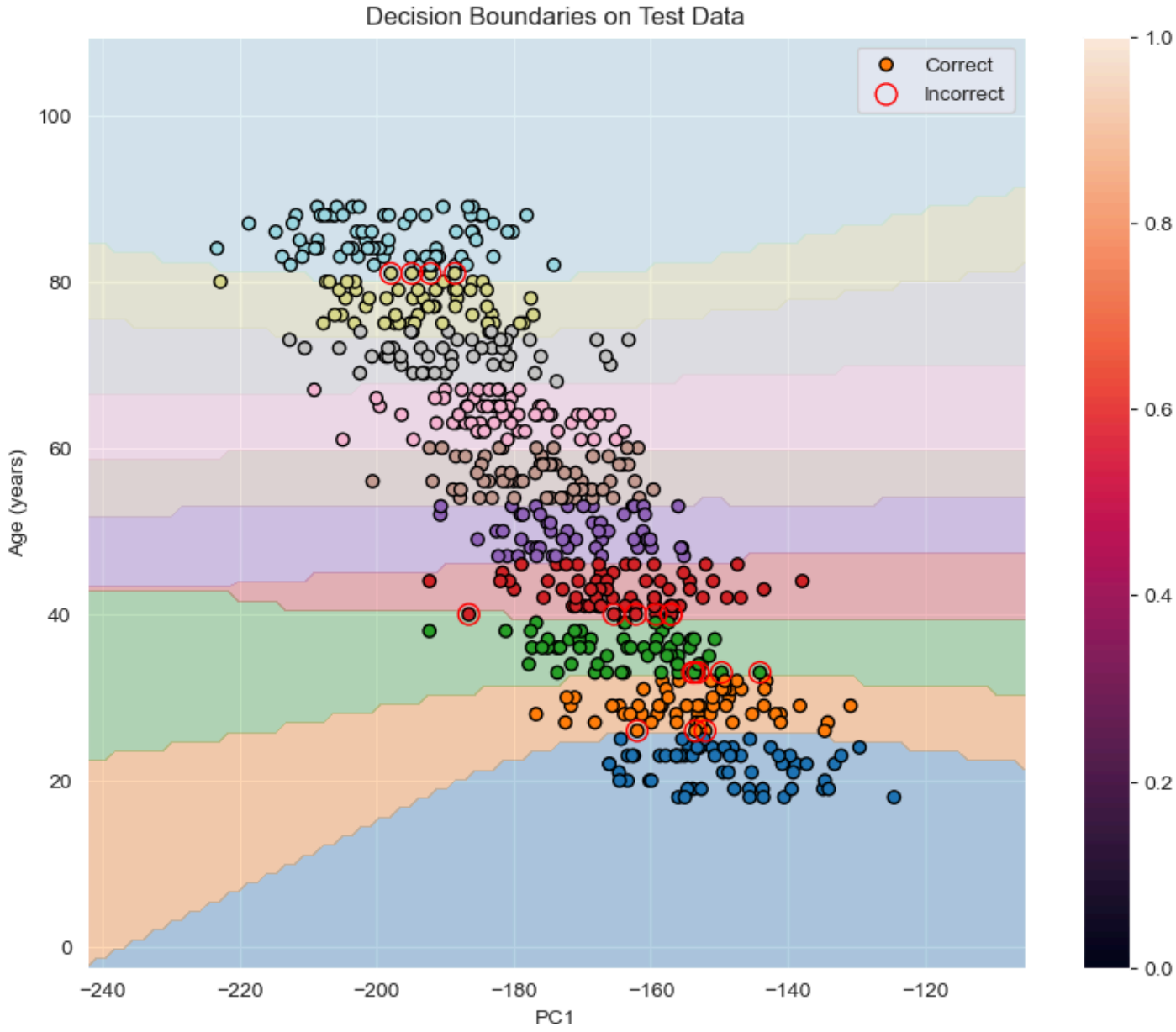


Analysis

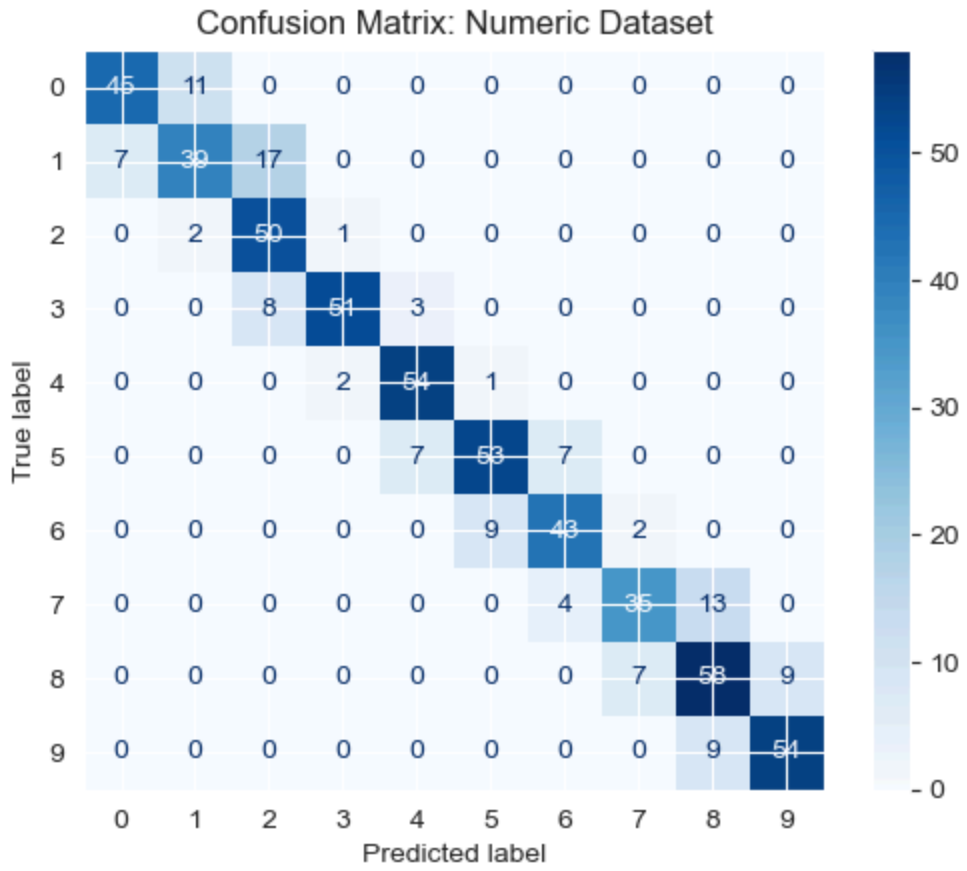
- Lastly, the sigmoid kernel performed the worst across all datasets, with the lowest accuracy. This is expected and is likely due to the fact that the sigmoid kernel is not as effective at separating the data as the other kernels

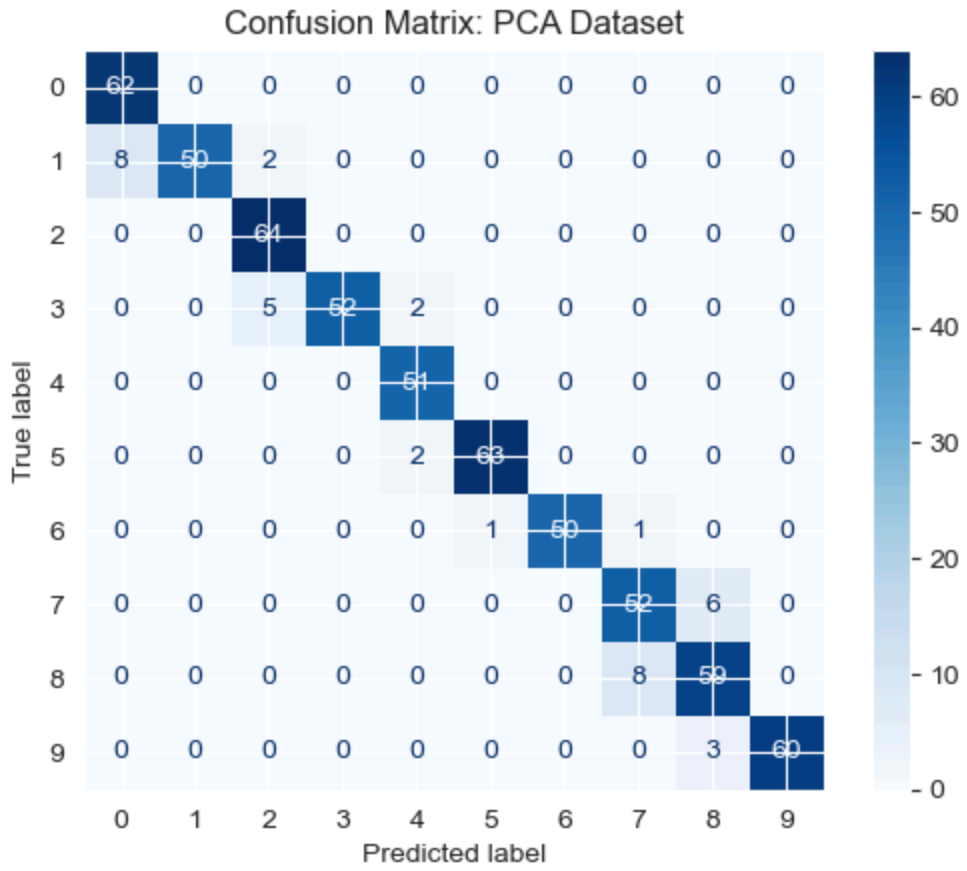
Results and Discussion - FNN Visualizations

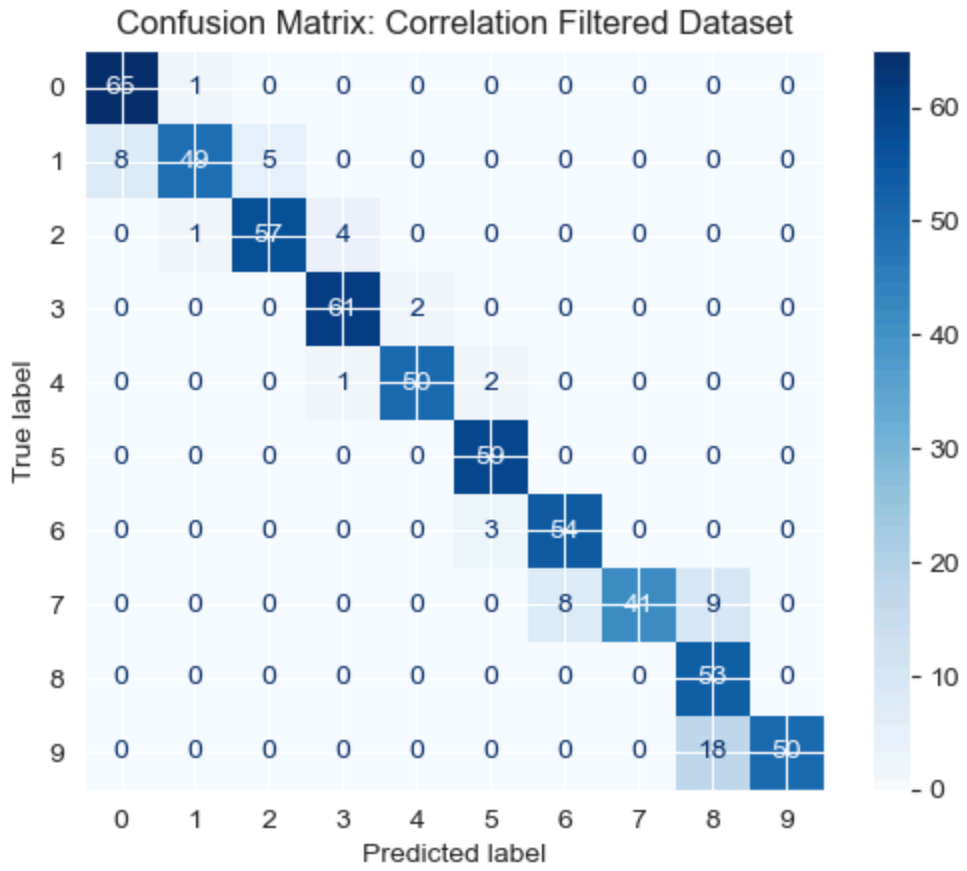
Decision Boundaries



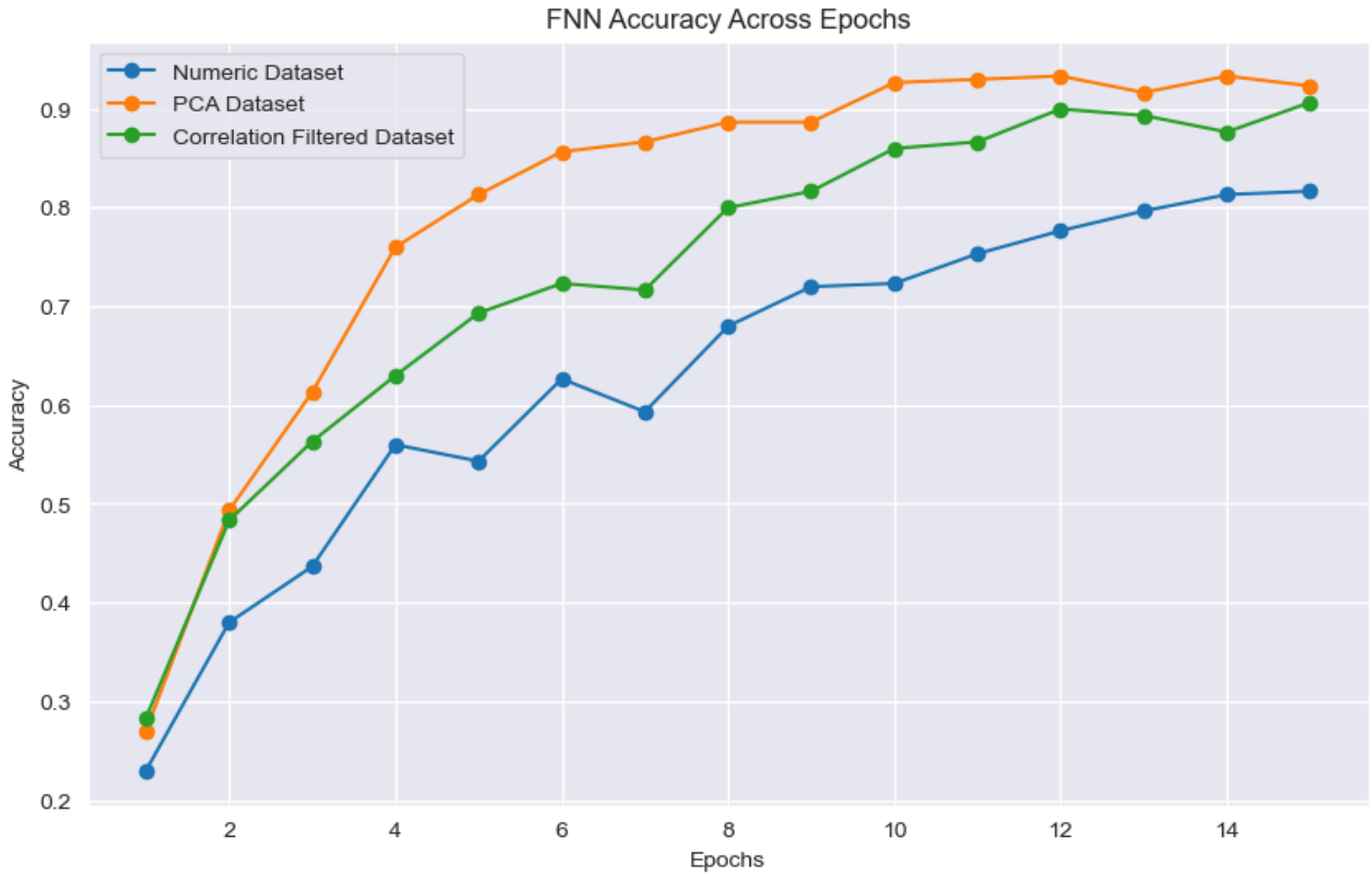
Confussion Matrix


Confusion Matrix: Numeric Dataset
True label
Predicted label
0 1 2 3 4 5 6 7 8 9
45 11 0 0 0 0 0 0 0 0
7 39 17 0 0 0 0 0 0 0
0 2 59 1 0 0 0 0 0 0
0 0 8 61 3 0 0 0 0 0
0 0 0 2 54 1 0 0 0 0
0 0 0 0 7 53 7 0 0 0
0 0 0 0 0 9 43 2 0 0
0 0 0 0 0 0 4 35 13 0
0 0 0 0 0 0 0 7 58 9
0 0 0 0 0 0 0 0 9 54
50 40 30 20 10 0


Confusion Matrix: PCA Dataset
True label
Predicted label
0 1 2 3 4 5 6 7 8 9
62 0 0 0 0 0 0 0 0 0
8 50 2 0 0 0 0 0 0 0
0 0 64 0 0 0 0 0 0 0
0 0 5 62 2 0 0 0 0 0
0 0 0 0 61 0 0 0 0 0
0 0 0 0 2 63 0 0 0 0
0 0 0 0 0 1 59 1 0 0
0 0 0 0 0 0 0 52 6 0
0 0 0 0 0 0 0 8 59 0
0 0 0 0 0 0 0 0 3 60
60 50 40 30 20 10 0



FNN Accuracy



Analysis

Numeric Dataset

- The numeric dataset performed the worst due to largely noisy features. The high dimensionality made it harder for the model to generalize
- The features in the numeric dataset would have large scales making the data extremely noisy

PCA Dataset

- The PCA dataset performed the best
- Dimensionality reduction allowed the most significant features to be considered. This allowed the model to have less noisy data since only features that captured a large amount of variance in the data would be considered. This also meant that the model could focus on the most uncorrelated data to find patterns.

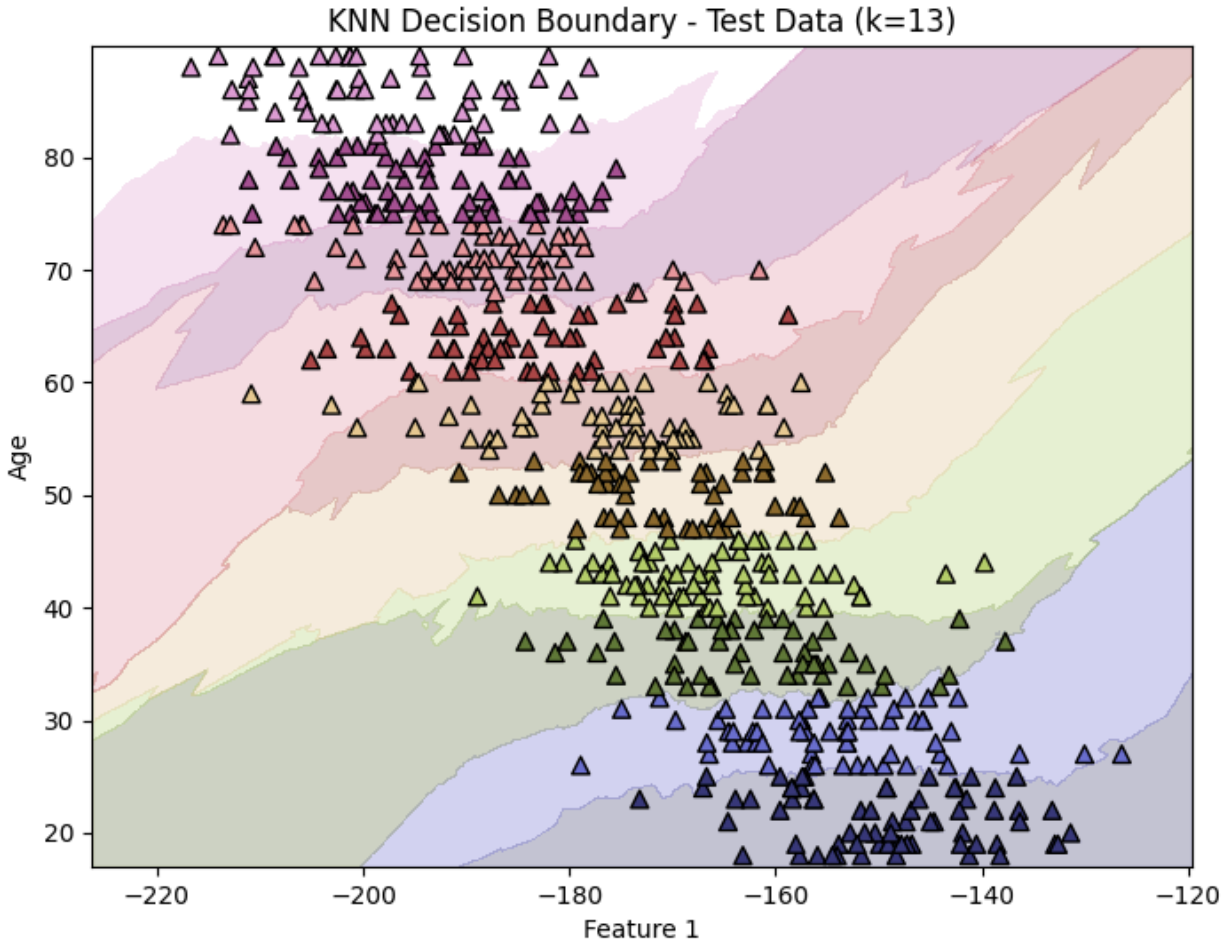
Correlation Filtered Dataset

- The correlation filtered dataset benefitted from feature selection
- Correlation filtering removed the features with low predictive power/high redundancy. This improved the model's performance compared to the numeric dataset, but may still leave

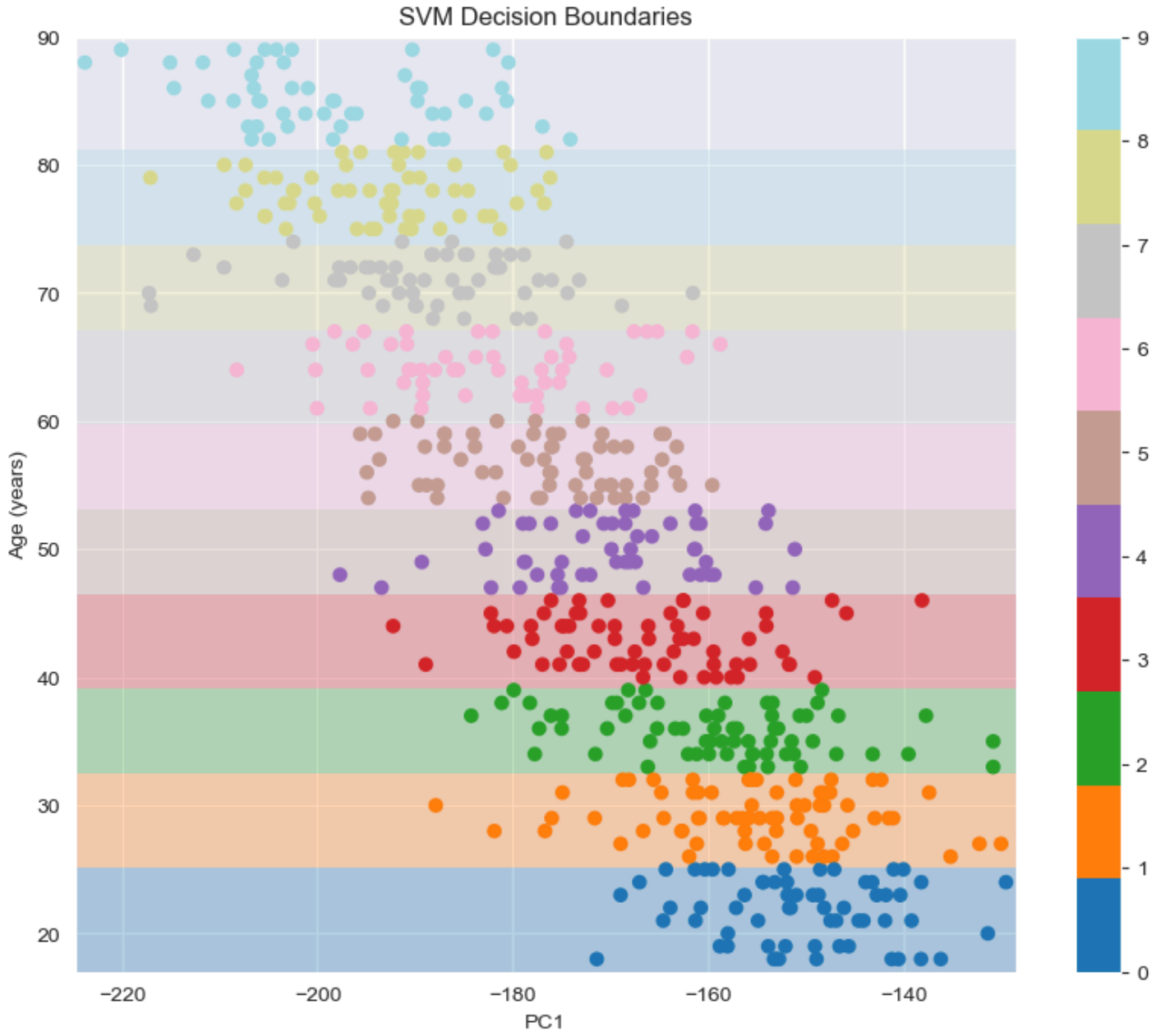
more room for noise than PCA

Comparison

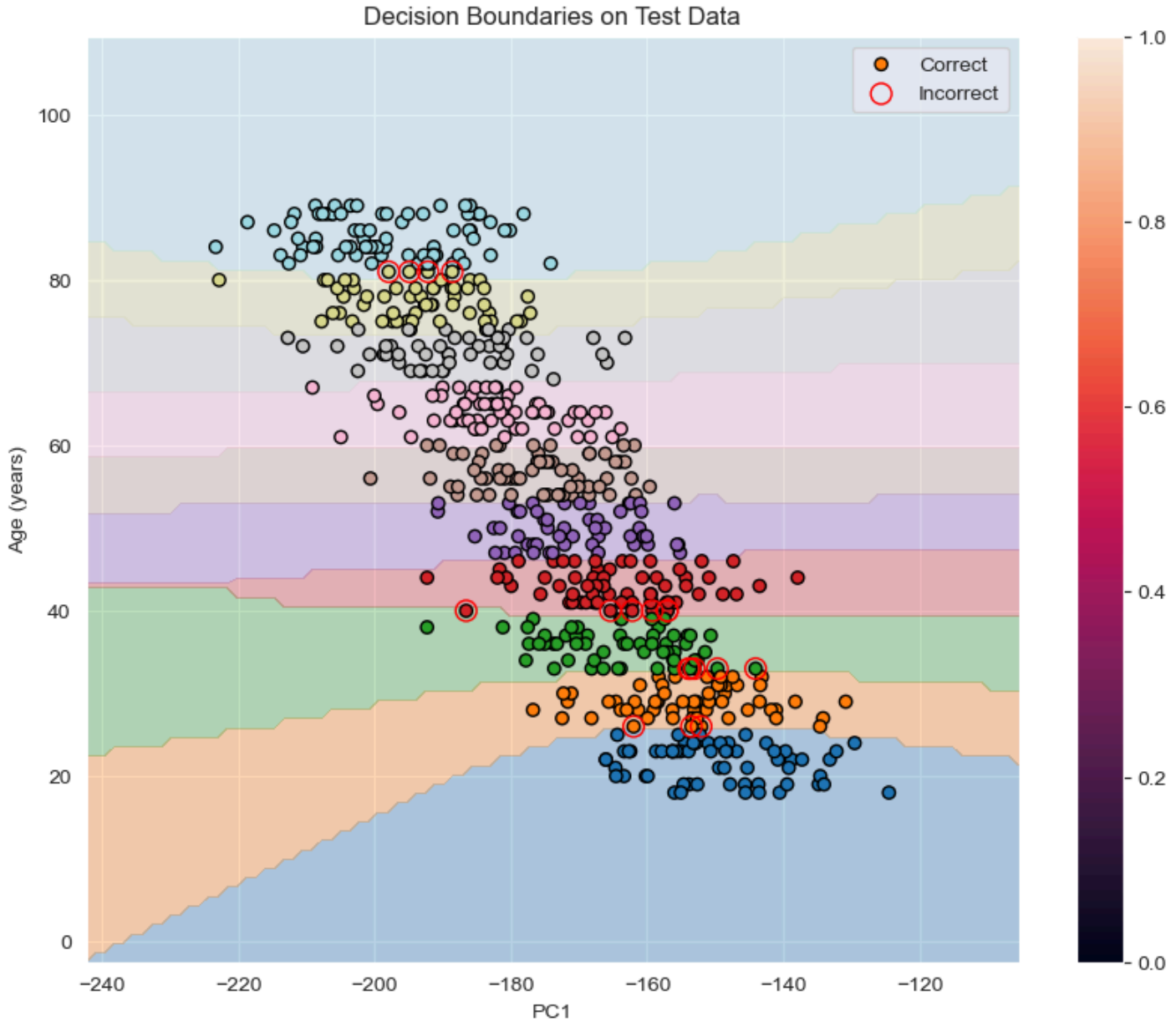
KNN



SVM


SVM Decision Boundaries
Age (years)
PC1
90
80
70
60
50
40
30
20
−220
−200
−180
−160
−140
9
8
7
6
5
4
3
2
1
0

FNN



Conclusion

- SVM with linear kernel perform best. As stated before, there is no real explanation for why the linear kernel performed the best aside, from the fact that some features of the data may be linearly separable, and the linear kernel is able to effectively classify the data, more than KNN and FNN.

Next step

- If continue, our group would like to implement more models or testing our current models with different datasets.

References

[1] Alan Le Goallec, S. Collin, M. Jabri, S. Diai, T. Vincent, and C. J. Patel, “Machine learning approaches to predict age from accelerometer records of physical activity at biobank scale,” PLOS Digital Health, vol. 2, no. 1, pp. e0000176–e0000176, Jan. 2023, doi: https://doi.org/10.1371/journal.pdig.0000176.

[2] C. Wang et al., “A machine learning–based biological aging prediction and its associations with healthy lifestyles: the Dongfeng–Tongji cohort,” Annals of the New York Academy of Sciences, vol. 1507, no. 1, pp. 108–120, Sep. 2021, doi: https://doi.org/10.1111/nyas.14685.

[3] M. abdullah, “Human Age Prediction Synthetic Dataset,” Kaggle.com, 2024. https://www.kaggle.com/datasets/abdullah0a/human-age-prediction-synthetic-dataset?select=Train.csv.

[4] C.-Y. Bae et al., “Comparison of Biological Age Prediction Models Using Clinical Biomarkers Commonly Measured in Clinical Practice Settings: AI Techniques Vs. Traditional Statistical Methods,” Frontiers in Analytical Science, vol. 1, Dec. 2021, doi: https://doi.org/10.3389/frans.2021.709589.

[5] “sklearn.cluster.DBSCAN — scikit-learn 0.22 documentation,” Scikit-learn.org, 2017. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

[6] “SMOTE — Version 0.9.0,” imbalanced-learn.org. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

[7] “PCA,” scikit-learn, 2024. https://scikit-learn.org/dev/modules/generated/sklearn.decomposition.PCA.html

[8] J. T. Hancock and T. M. Khoshgoftaar, “Survey on categorical data for neural networks,” Journal of Big Data, vol. 7, no. 1, Apr. 2020, doi: https://doi.org/10.1186/s40537-020-00305-w.

[9] “KNeighborsClassifier,” scikit-learn, 2024. https://scikit-learn.org/dev/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[10] “1.4. Support Vector Machines — scikit-learn 0.20.3 documentation,” Scikit-learn.org, 2018. https://scikit-learn.org/stable/modules/svm.html

[11] M. Roser, E. Ortiz-Ospina, H. Ritchie, S. Dattani, and L. Rodés-Guirao, “Life Expectancy,” Our World in Data, 2023. https://ourworldindata.org/life-expectancy

Explanation of Files

There are four main directories that support this project. `/data` which contains all pre-processing methods and derived datasets, `/knn` which contains a jupyter notebook regarding to applying

KNN to solve the problme, `/svm` which contains a jupyter notebook regarding applying SVM to solve the problem, and `/docs/assets/` which contains the visual assets that support this `README.md` file.

- `/data`
  - `/data/data.csv` contains the original dataset sourced from Kaggle.
  - `/data/eda.ipynb` contains the jupyter notebook used to initialize and visualize metrics regarding the original dataset. Furthermore it contains pre-processing methods to generate three new datasets (numerically encoded: `/data/numeric/data.csv`, correlation filtered: `/data/correlation_filtered/data.csv`, pca variation thresholded: `/data/pca/data.csv`, pca derived visualization dataset: `/data/pca/vis_data.csv`).
  - `/data/data.py` contains utilities to enable better data loading for ScikitLearn and Pytorch models, as well as a numerical categorization feature to convert ages in years to classifiable categories which is required for the classification task.
- `/knn`
  - `/knn/knn.ipynb` contains the jupyter notebook used to perform KNN via Scikit Learn.
  - All other files in this directory are exported images that take the outputted plots for training and testing and save them as `.png` s.
- `/svm`
  - `/svm/svm.ipynb` contains the jupyter notebook used to perform SVM via Scikit Learn.
- `/nn`
  - `/nn/nn.ipynb` contains the jupyter notebook used to train and evaluate the Neural Network via PyTorch.
- `/docs/assets`
  - `/docs/assets/data` contains image outputs of the `~/eda.ipynb` file.
  - `/docs/assets/knn` contains image outputs of the `~/knn.ipynb` file.
  - `/docs/assets/svm` contains image outputs of the `~/svm.ipynb` file.
  - `/docs/assets/nn` contains image outputs of the `~/nn.ipynb` file.

Gantt Chart

https://docs.google.com/spreadsheets/d/1GZS4CvdPtn4_A_fZCol2bqostQoGsKuJm5mnv2ZrFKU/edit?usp=sharing

Contributions

Name	Contributions
Daniel Park	FNN Model
Keerthi Sridhar Kaashyap	Data Pre-Processing and SVM Model
Matthew Tang Zhong	FNN Model
Ngoc Truong Tran	Final Report, GitHub Pages, and Video Presentation
Richard George Borowski	KNN Model