CS-UY 4563: Introduction to Machine Learning
Section A
Final Project Written Report
Parkinson’s Disease Progression Classification
April 30, 2025
Professor Linda N. Sellie
Nelson Jiang, David Zeng

I. Introduction

This machine learning project focuses on predicting the various stages of Parkinson's diseases in patients. The dataset is composed of 500 training samples and each sample is composed of ten features and one label. The ten features are **age, gender, years since diagnosis, UPDRS score (assesses Parkinson's symptoms), tremor severity, motor function, speech difficulty, balance issues, exercise level** and **medication** while the **label** is the **stage of Parkinson disease denoted 1 through 3** with 3 being the most advanced stage. This project is a multiclass categorization problem and we implemented three supervised models, **logistic regression (i.e. softmax regression)**, **SVM**, and **neural networks** and one unsupervised model **K-Clustering** to tackle this problem. Scikit-learn libraries were used for modeling, while Numpy, Pandas, Scipy, and Matplotlib were used for data manipulation and visualization.

II. Data Preparation

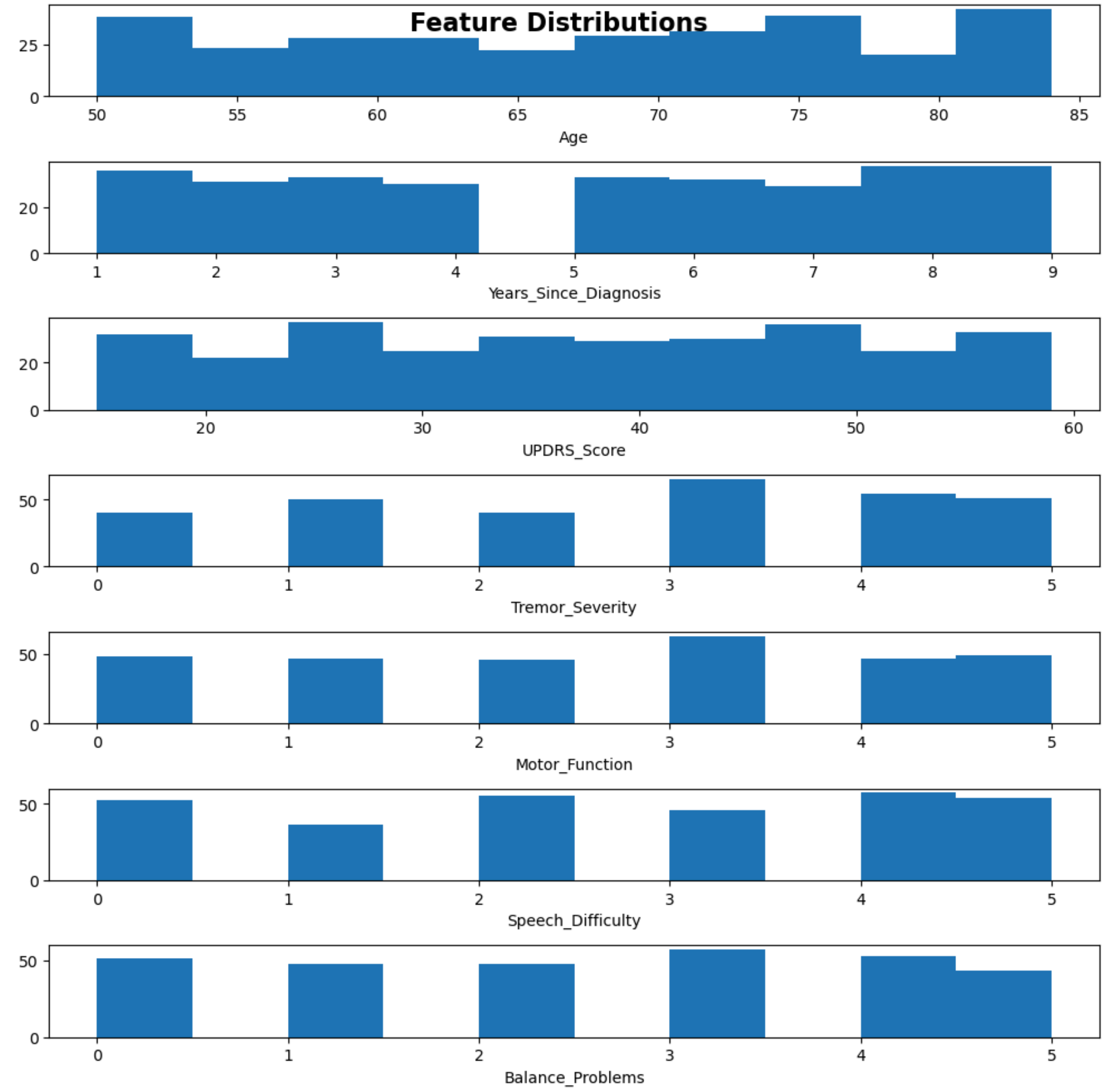
Prior to any training, we polished our dataset to enhance the performance of our machine learning models. We implemented different encoding strategies on different features. For instance, on the 'age', "years since diagnosis', and 'UDPRS' features, we **standardized** the values, on the 'gender' and 'medication' features we implemented **dummy variable encoding**, on the 'tremor's', 'balance', 'motor', 'speech' and 'exercise' features we implemented **ordinal encoding**. Finally, we eliminated the PatientID column as it did not provide any relevant information to our machine learning models.

Our entire dataset consists of 500 samples and we decided upon a **60-20-20 random split** of our dataset meaning we use 60% of our dataset (**300 sample points**) as training data, and 20% of our dataset as validation (**100 sample points**) and testing data (**100 sample points**) each. We later tried a round of training using a 70-15-15 random split of the dataset into training, validation, and testing sets; since the performance of the models suffered with this reallocation of data, the previous 60-20-20 split of the data was restored.

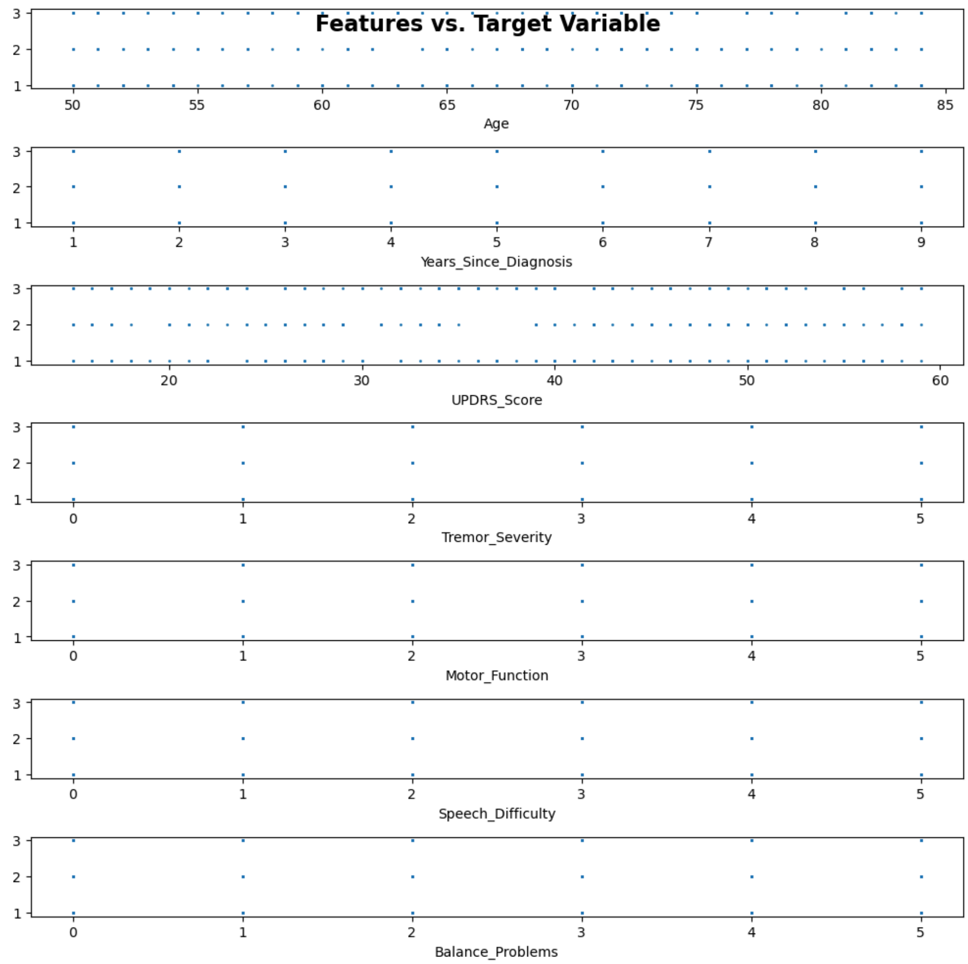
III. Unsupervised Analysis

Data Visualization

The feature distributions for the seven features that were originally numeric were plotted as histograms as shown below. The distributions for all of these variables are roughly uniform: the gap in the Years_Since_Diagnosis feature is due to the empty bin not aligning with any integer values.

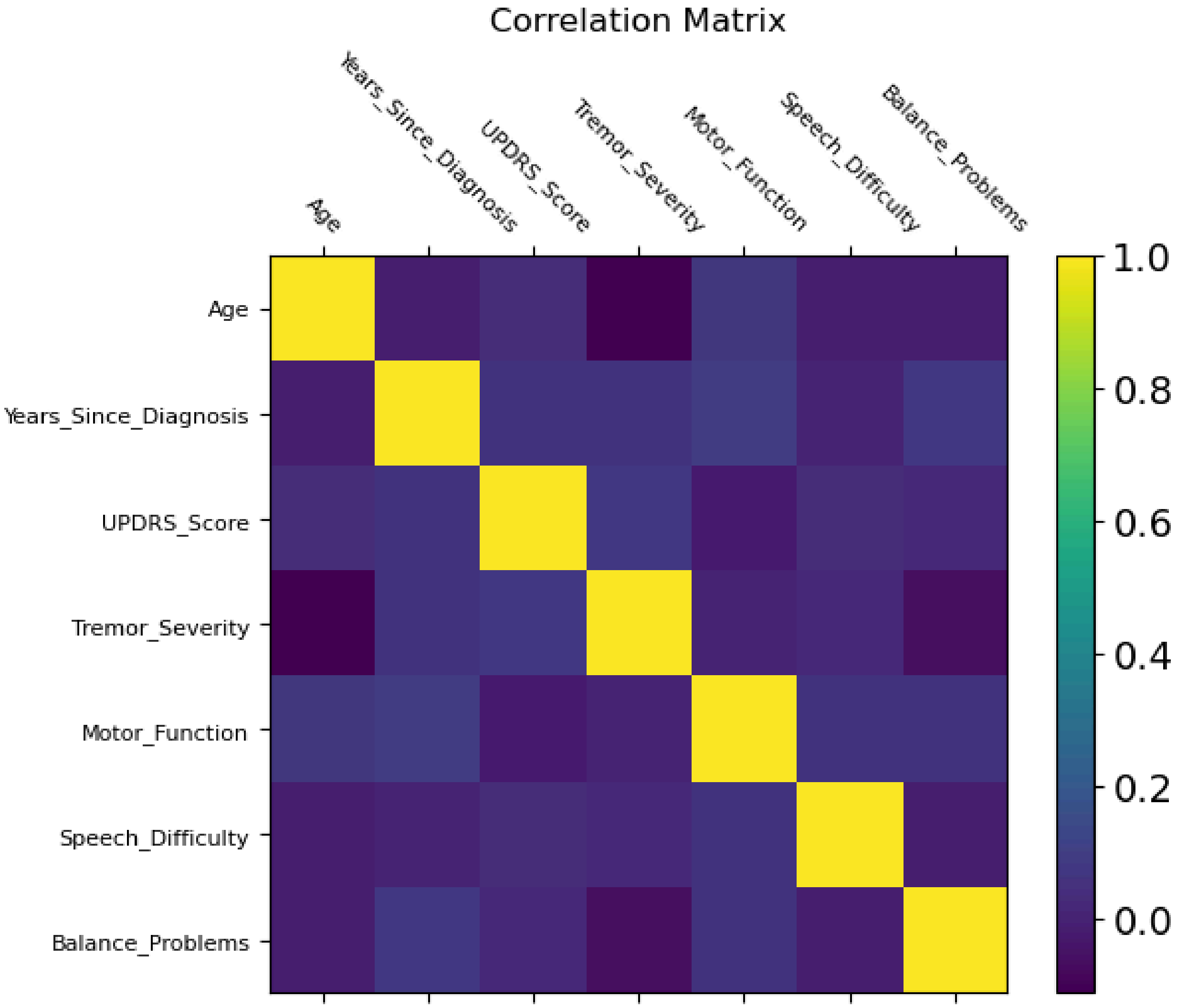


The numeric features were also plotted against the target variable, Parkinson's disease stage, as shown below on the left. The scatterplot for these plots is composed of rows with discrete dots due to the low number of possible values for many of these features and the target variable. No clear trends are observable using these plots: almost all values of each plotted feature correspond to every possible label. The standard correlation coefficients for each numeric feature with the target variable are shown in the table below: any correlations that may exist are extremely weak with the absolute value of the correlation coefficients never exceeding 0.07.



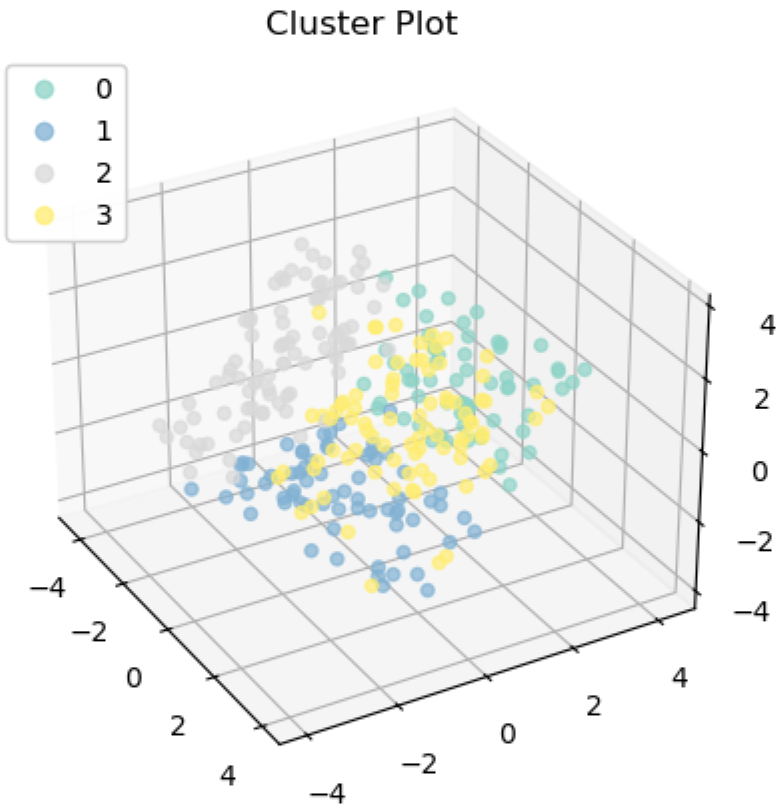
Standard Correlation Coefficients of Feature with Parkinson’s Disease Stage	
Years Since Diagnosis	0.046
Speech Difficulty	0.014
Motor Function	0.0089
Balance Problems	-0.0096
Tremor Severity	-0.040
Age	-0.056
UPDRS Score	-0.067

A correlation matrix was created to analyze relationships between different numeric features, as shown below. The key on the right of the matrix identifies which color squares correspond to which correlation coefficients. The highest positive or negative correlation between different features has a correlation coefficient with an absolute value of 0.111, which is extremely low. Therefore, there are no apparent relationships between any of the numeric features.



Clustering and Dimensionality

Clustering and dimensionality analysis using principal component analysis (PCA) were both harnessed to explore the structure of the data. For clustering analysis, the features were reduced to three components using PCA first to make the clusters visualizable in a 3D plot. The elbow method was used to minimize the within-clusters sum of squares by choosing a k-value at the point where the rate of decrease in variance decreased significantly, which resulted in a k-value of 4 being chosen. K-means++ clustering was performed for four clusters, with the clusters shown below. The clusters were relatively indistinct with blended boundaries and no separation between clusters, indicating that the data did not naturally fall into sharply defined clusters.



Since the 3-component PCA previously performed may not have used the optimal number of components, PCA was run again using Minka's maximum likelihood estimation (MLE) method to estimate the optimal number of components. Using Minka's MLE, the optimal number of components was determined to be 11, which was a reduction of 1 dimension compared to the original 12 features, including encoded features. Based on the minimal dimensionality reduction from PCA, the existing features were already relatively independent of each other: a quality which is supported by the lack of correlations between features in the correlation matrix.

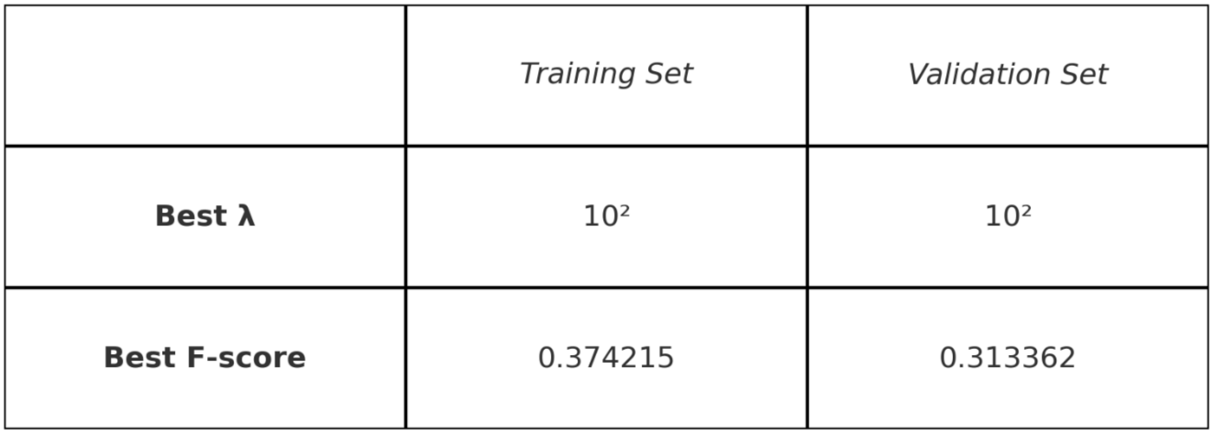
Based on the unsupervised analysis of the features and target variable, the features seem to be relatively uniformly distributed with almost nonexistent correlations between each other and the target variable, while lacking any identifiable structure. The lack of imbalance will help with training, but the lack of any identified relationships or structure means data reduction is less efficient and trying to find any true relationships to use in a predictor will be relatively difficult.

The data was relatively balanced between classes, with each Parkinson's disease progression stage having a roughly equal number of members. Therefore, macro-averaged metrics were used for model analysis as the classes were balanced and each class was of equal importance. The F1 score, the harmonic mean of precision and recall, was used as the primary metric for model analysis in this multiclass classification problem as a balance between precision and recall, which were deemed to be of equal importance.

IV. Supervised Learning Model: Logistic Regression

For Logistic Regression, we implemented four polynomial transformations of our features (one untransformed and three transformations) and for each of the transformations we tested eight L2 regularization hyperparameters (i.e. lambda values 10^1 through 10^8). We chose these large regularization hyperparameters to try to drive down any overfitting that might occur. In our logistic regression model, we used a macro-averaged (i.e. computing the f-score for each class and then averaging those computed f-scores) f-score as our primary measurement of performance. We decided upon this metric due to the multiclass nature of this problem and since all classes of labels are equally important in this dataset.

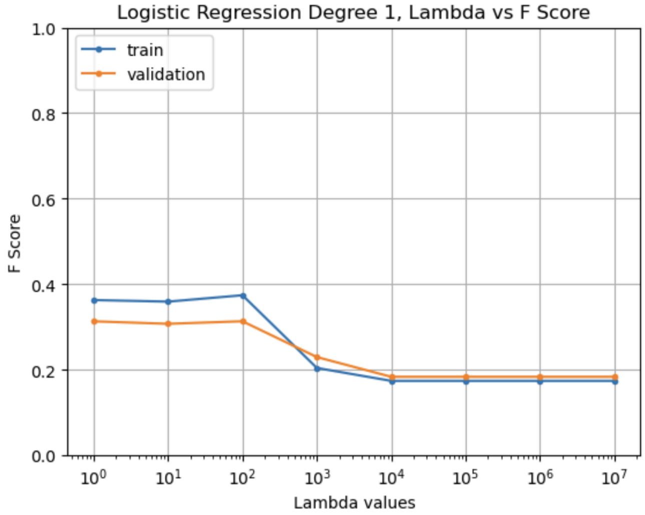
Firstly, we used an untransformed, original dataset which is a polynomial feature transformation of 1. We fit our model on our test set, our validation set and training set and reached the following conclusions summarized in the table:



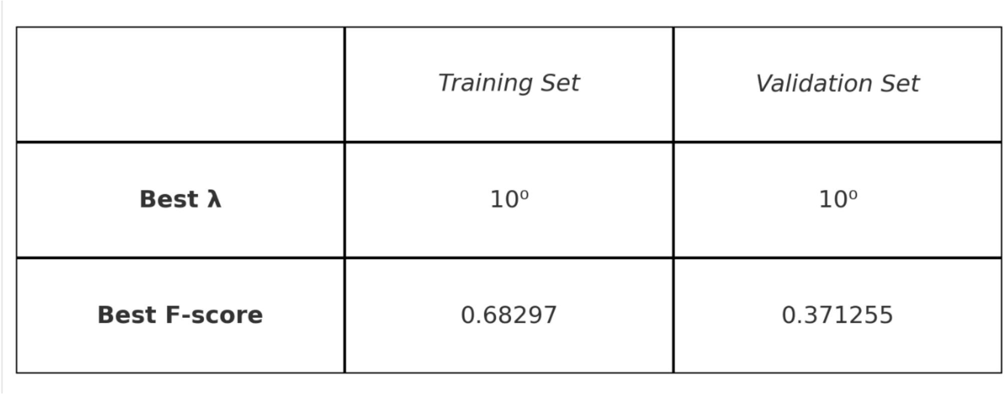
	Training Set	*Validation Set*
Best λ	10^2	10^2
Best F-score	0.374215	0.313362

When fitting on the training set the best hyperparameter was 10^2 which obtained an f-score of 0.374215 and working on the validation set to select the best regularization hyperparameter, we concluded that a lambda of 10^2 produced the best f-score of

0.313362. Furthermore, across all of the regularization hyperparameters that were tested, we graphed the results as shown in the following plot

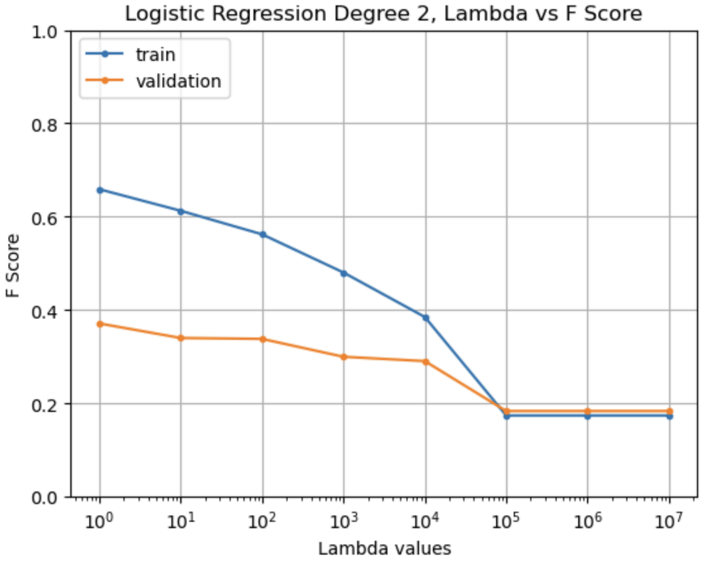


The second transformation performed was a polynomial transformation of degree 2. When we fitted our model on the training set, we discovered that the best regularization hyperparameter was 10^0 which produced an f-score of 0.68297 and when selecting for the optimal model, we tested on our validation set which concluded that a lambda of 10^0 produced the best f-score of 0.371255. The conclusions are summarized in the following table:



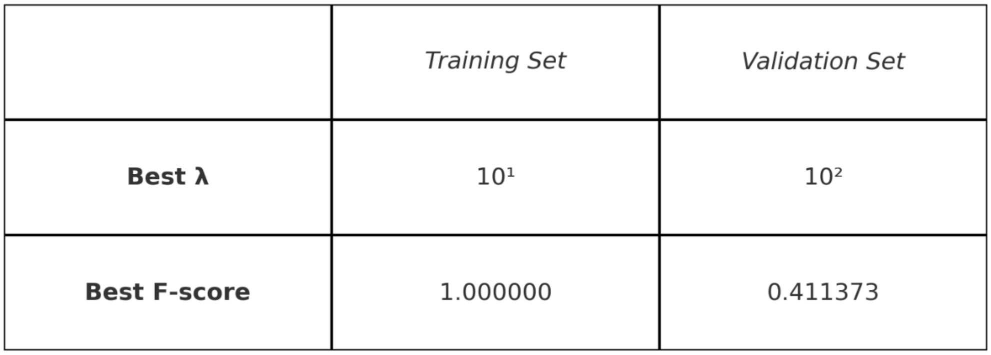
	Training Set	*Validation Set*
Best λ	10^0	10^0
Best F-score	0.68297	0.371255

Furthermore, across all regularization hyperparameters, we graphed the results in the following plot



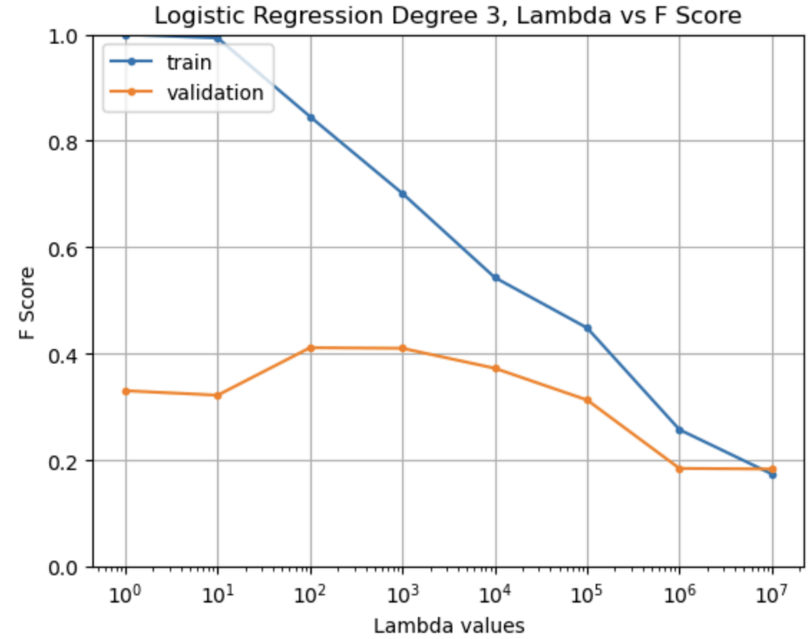
We can observe a trend that as when we perform a degree 2 polynomial transformation, our f-score is dropping as lambda increases until it plateaus when lambda is 10^5.

The third transformation performed was a polynomial transformation of degree 3. We fit our model on our training set and obtained the best f-score of 1 using the hyperparameter of 10^1 then when selecting the optimal model we tested on our validation set and concluded that the best f-score produced is 0.411373 using lambda of 10^2 . The results are summarized in the following table:



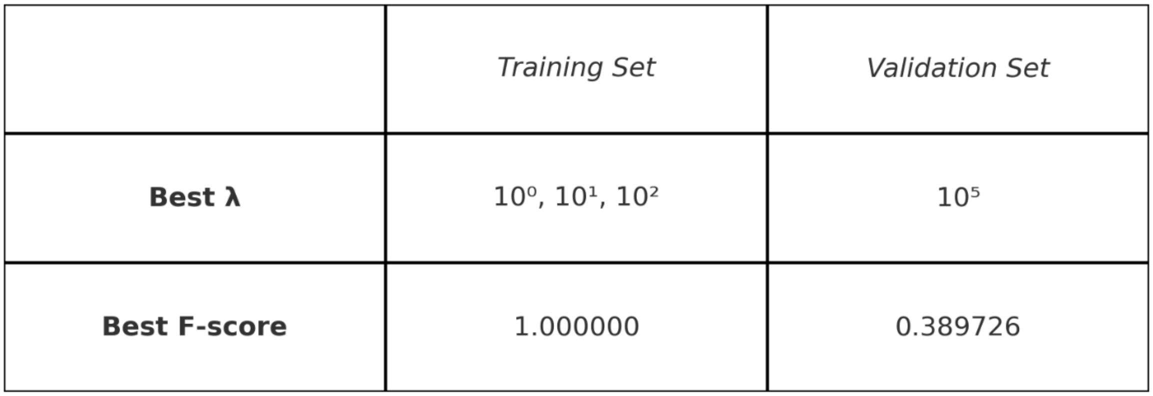
	Training Set	*Validation Set*
Best λ	10^1	10^2
Best F-score	1.000000	0.411373

Furthermore, across all regularization hyperparameters, we graphed the results in the following plot:



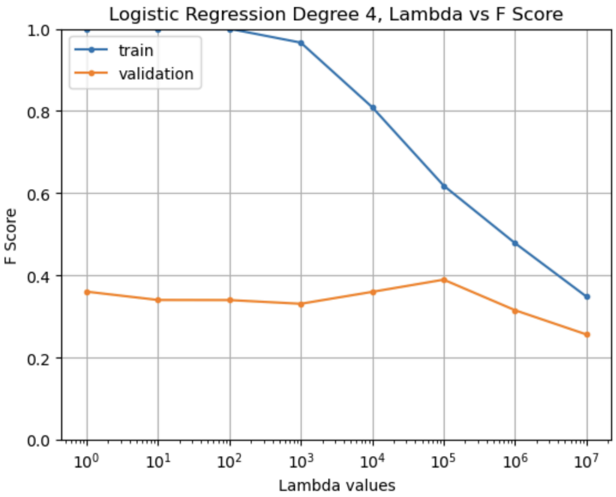
This transformation is similar to the degree 2 transformation in that as lambda increases, the f-score for the training set is plummeting; however, the f-score for the validation set is acting in a more erratic manner compared to how it behaved in the degree 2 transformation since for this transformation, the f-score for the validation set is rising and then dropping.

Lastly, our fourth transformation is a degree four polynomial transformation. We fit our model on our training set and test on our validation to select the ideal model and reached the following conclusions summarized in our table:

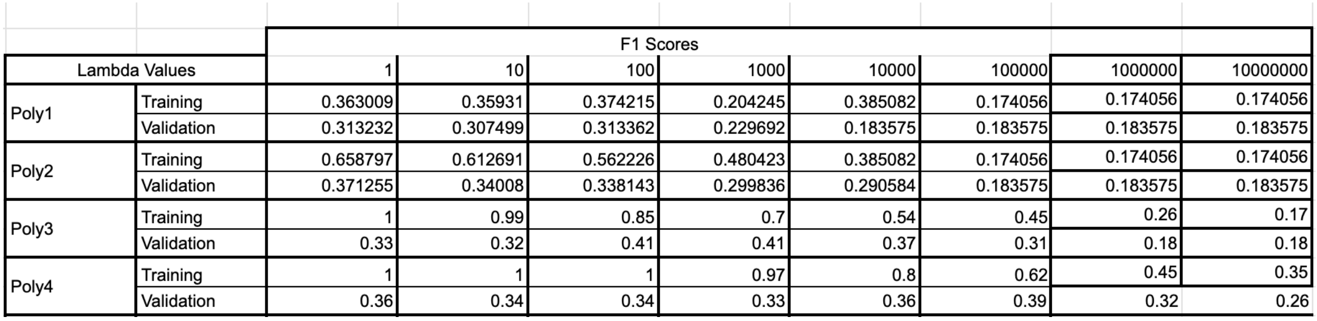


	Training Set	*Validation Set*
Best λ	10^0, 10^1, 10^2	10^5
Best F-score	1.000000	0.389726

As indicated in the table, as we performed a degree four transformation on our features, our training f-score achieved the maximum score of 1.0 when using lambdas 10^0 ,10^1 and 10^2 while on the validation set, the optimal f-score is 0.389726 achieved with a lambda of 10^5. Furthermore, across all regularization hyperparameters, we graphed the results in the following plot:



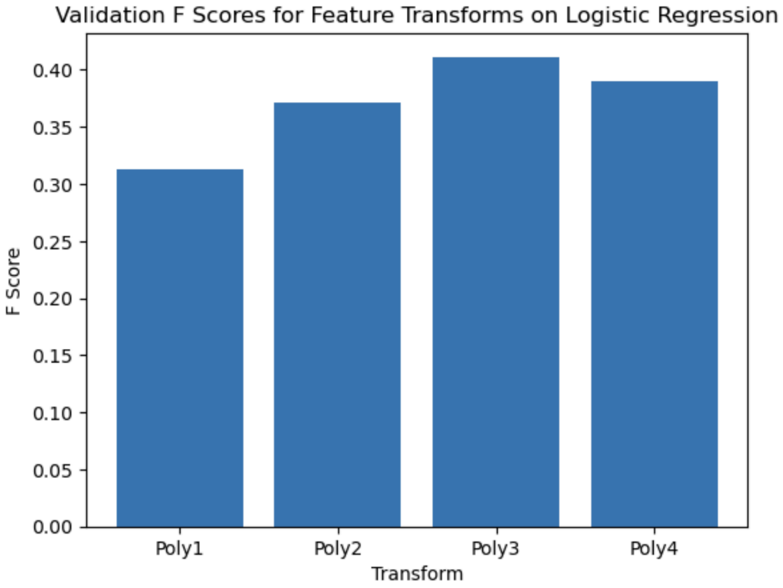
Notice in the graph that as our lambda increases, our training f-score is dropping rapidly (and not plateauing like the previous transformations) which is expected behavior since increasing regularization will drive down the training set's accuracy as the difference between training set's accuracy and validation set's accuracy shrinks. Below we include a table with all of the data:



		F1 Scores							
Lambda Values		1	10	100	1000	10000	100000	1000000	10000000
Poly1	Training	0.363009	0.35931	0.374215	0.204245	0.385082	0.174056	0.174056	0.174056
	Validation	0.313232	0.307499	0.313362	0.229692	0.183575	0.183575	0.183575	0.183575
Poly2	Training	0.658797	0.612691	0.562226	0.480423	0.385082	0.174056	0.174056	0.174056
	Validation	0.371255	0.34008	0.338143	0.299836	0.290584	0.183575	0.183575	0.183575
Poly3	Training	1	0.99	0.85	0.7	0.54	0.45	0.26	0.17
	Validation	0.33	0.32	0.41	0.41	0.37	0.31	0.18	0.18
Poly4	Training	1	1	1	0.97	0.8	0.62	0.45	0.35
	Validation	0.36	0.34	0.34	0.33	0.36	0.39	0.32	0.26

Note that the higher the transformation didn't necessarily lead to the best performance on the validation set.

In conclusion, for the logistic regression class of models, we should select the model that uses a **degree 3 polynomial transformation and with a regularization hyperparameter of 10^2** to model against our Test set since amongst the testing on validation sets, this produced the highest f-score.



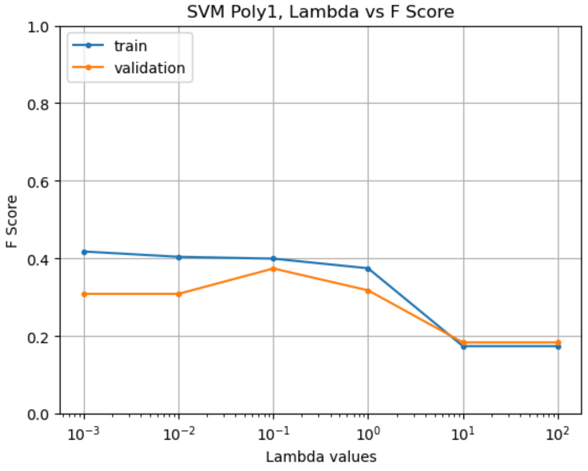
V. Supervised Learning Model: SVM

We also trained support vector machines (SVM) to predict the stage of Parkinson's disease progression. In addition to the use of untransformed data (Poly1), we also tried a polynomial kernel with degrees 2 and 3 (Poly2, Poly3), along with a radial bias function (RBF) kernel and a feature transformation using PCA. For the PCA transformation, 11 components were used as previously identified by Minka's MLE. Polynomial kernels of increasing degrees were chosen to examine the impact of increasing model complexity on the model predictions. The RBF kernel was chosen as a general-purpose kernel and a PCA transformation was chosen to reduce data interdependency and increase the speed of model training. However, the poor correlations between features and the low amount of dimensionality reduction achieved using PCA suggest that runtime benefits for PCA would be limited.

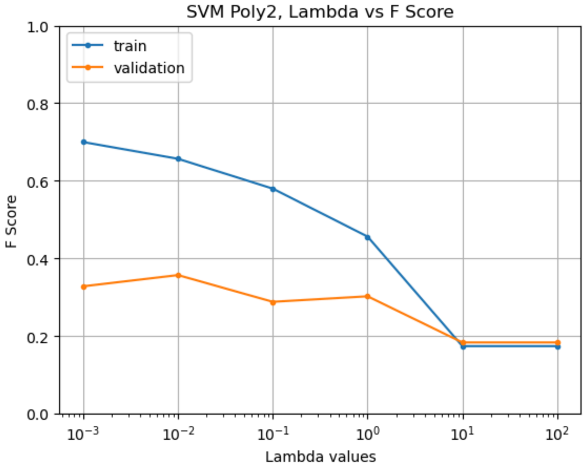
For each model class, six regularization values were tested as powers of 10, ranging from 10^{-3} to 10^{2}. The training and validation F1 scores for each regularization value of each feature transformation / kernel for SVM are in the table below.

		F1 Scores					
Lambda Values		0.001	0.01	0.1	1	10	100
Poly1	Training	0.417	0.404	0.399	0.375	0.174	0.174
	Validation	0.309	0.309	0.374	0.318	0.184	0.184
Poly2	Training	0.7	0.657	0.58	0.456	0.174	0.174
	Validation	0.328	0.357	0.288	0.302	0.184	0.184
Poly3	Training	1	0.936	0.739	0.533	0.267	0.174
	Validation	0.342	0.334	0.343	0.369	0.208	0.184
RBF	Training	1	0.99	0.751	0.472	0.174	0.174
	Validation	0.41	0.4	0.42	0.325	0.184	0.184
PCA	Training	0.36	0.359	0.352	0.377	0.174	0.174
	Validation	0.311	0.311	0.312	0.335	0.184	0.184

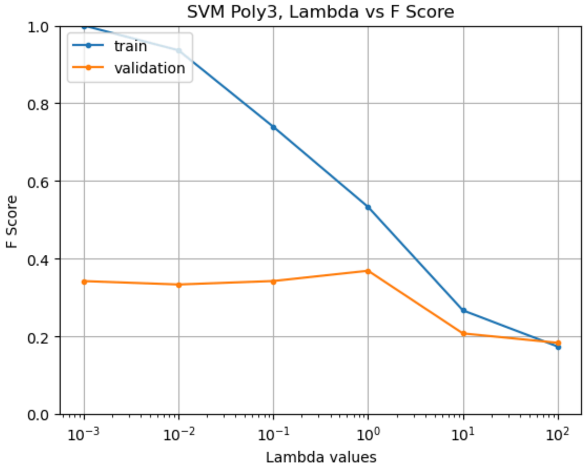
The plots of training and validation F1 scores against different regularization values for each feature transformation / kernel are in the graphs below. As expected, higher levels of regularization lowered training F1 scores by restricting the degree to which the models could fit to the training set. Validation F1 scores saw minor initial improvements with higher regularization, marking the region where regularization helped combat model overfitting, followed by sharp drops with even higher regularization due to underfitting.



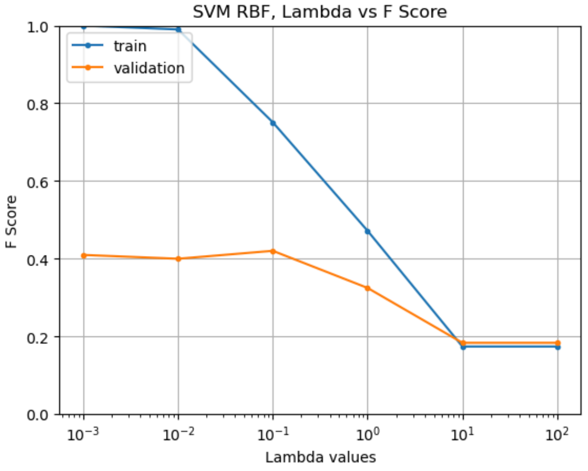
SVM Poly1, Lambda vs F Score



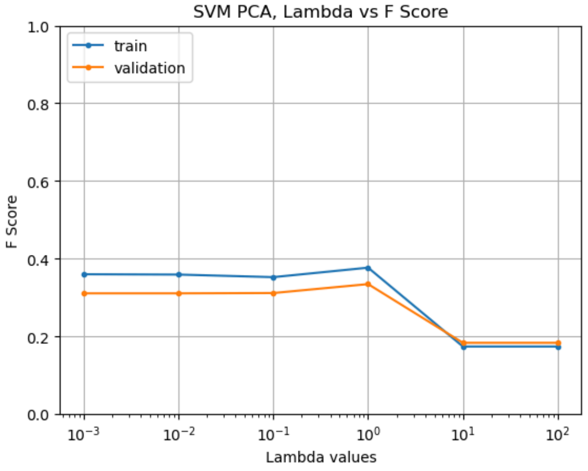
SVM Poly2, Lambda vs F Score



SVM Poly3, Lambda vs F Score



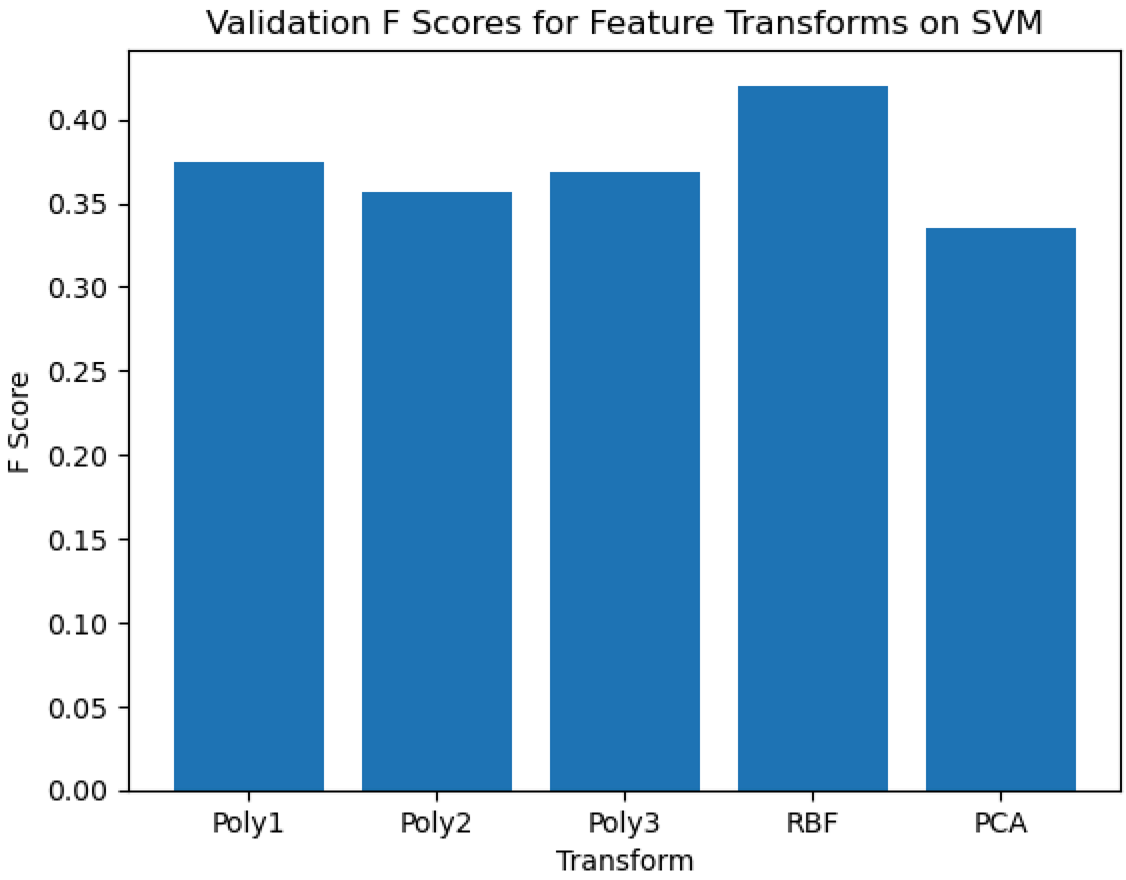
SVM RBF, Lambda vs F Score



SVM PCA, Lambda vs F Score

For higher regularization values (10 - 100 or greater), the training and validation F1 scores for all feature transformations / kernels all converged to the same range, roughly 0.38. This behavior is due to all models making predictions of Parkinson's disease stage 3 for every example regardless of its features when high regularization values were used. Since all models made the same predictions with high regularization values, the F1 scores converged for all models.

For each feature transformation, the regularization value (lambda) with the best validation F1 score was chosen: regularization values of 0.1, 0.01, 1, 0.1, and 1 were chosen for the Poly1, Poly2, Poly3, RBF, and PCA transformations / kernels, respectively. The different feature transformations / kernels and their best validation F1 scores given optimized hyperparameters are shown in the graph below.



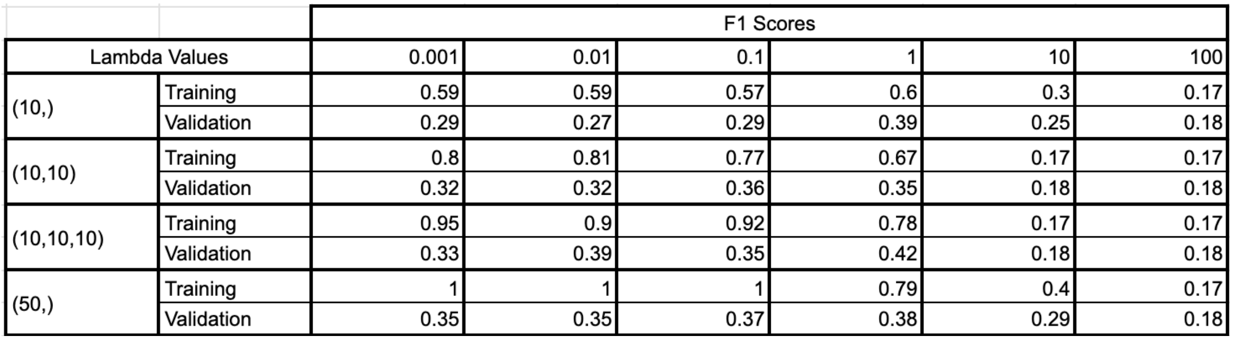
Higher degree polynomial kernels had better initial training F1 scores but showed no clear trends regarding the optimal regularization values or validation F1 scores. The PCA transform behaved similarly to the linear untransformed features but had consistently lower training and validation F1 scores. The RBF kernel had the highest training and validation F1 scores across multiple regularization values compared to the other feature transformations / kernels.

Given these validation metrics for the different feature transformations / kernels, the RBF kernel with a regularization value of 0.1 was chosen as the SVM model with the best performance against the validation set.

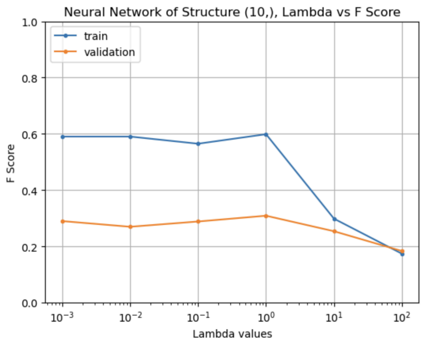
VI. Supervised Learning Model: Neural Network

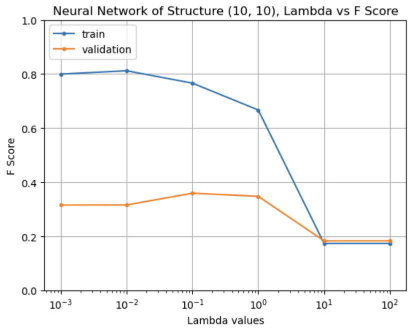
For the Neural Network models we implemented four architectures. The first architecture was the original architecture with 10 hidden layers, the second architecture was a network with two hidden layers of each being ten nodes , the third was three hidden layers with ten nodes each layer and the fourth architecture was a single hidden layer of 50 nodes. On all of our architectures, the input layer had twelve nodes since each data point had twelve features and the output layer had three nodes since we had three labels (Stage 1, 2, 3 of Parkinson disease) and for the activation function, we selected ReLU due to its quicker and more lightweight nature as compared to sigmoid. For the performance metric, we decided that the macro-averaged f-score will be the most reliable metric to measure.

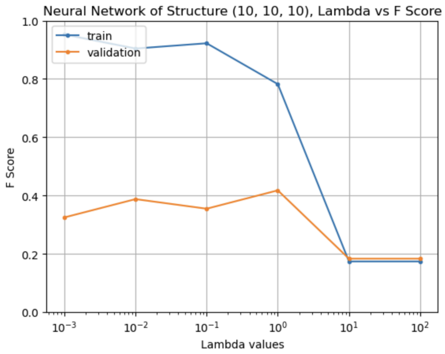
For each network architecture, we trained on six regularization hyperparameters (lambdas of 0.001,0.01,0.1,1,10,100) using the training set and then we tested on the validation set. The results of this is summarized in the table and graphs below:

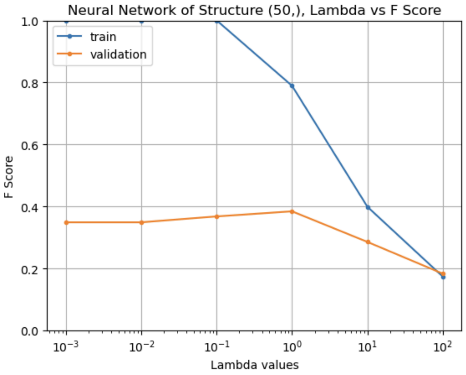


		F1 Scores					
Lambda Values		0.001	0.01	0.1	1	10	100
(10,)	Training	0.59	0.59	0.57	0.6	0.3	0.17
	Validation	0.29	0.27	0.29	0.39	0.25	0.18
(10,10)	Training	0.8	0.81	0.77	0.67	0.17	0.17
	Validation	0.32	0.32	0.36	0.35	0.18	0.18
(10,10,10)	Training	0.95	0.9	0.92	0.78	0.17	0.17
	Validation	0.33	0.39	0.35	0.42	0.18	0.18
(50,)	Training	1	1	1	0.79	0.4	0.17
	Validation	0.35	0.35	0.37	0.38	0.29	0.18

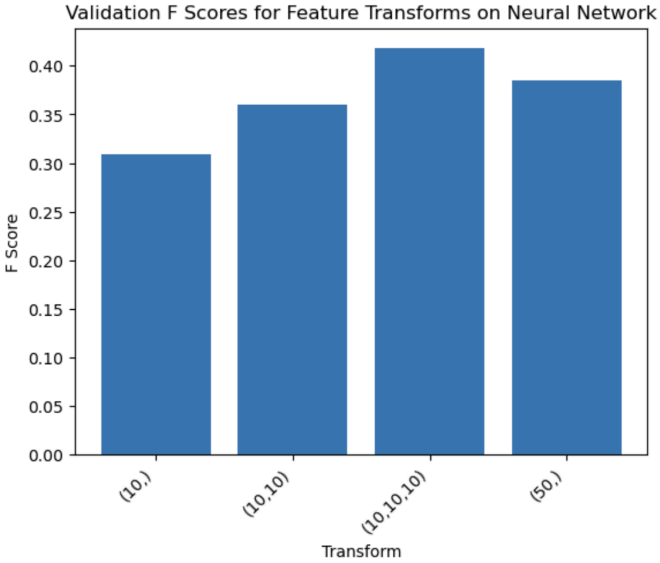








For our base architecture of (10,), we achieved the highest training training f-score of 0.6 using the regularization hyperparameter of 1 and achieved the highest validation f-score of 0.39 using the same regularization hyperparameter. For the architecture of (10,10), the optimal f-score on the training set was 0.81 using a lambda of 0.01 and for the validation set, the optimal f-score of 0.36 was achieved through a lambda of 0.1. For the architecture of (10,10,10), the best training f-score was 0.81 achieved using a lambda of 0.01 and the best validation f-score was 42 achieved using a lambda of 1. Finally, for the (50,) architecture, the best training f-score was 1 obtained using lambdas of 0.001, 0.001, 0.01 and the best validation f-score was of 0.38 obtained using lambda of 1. A pattern to observe is that as we implement more complex neural network architectures, the performance difference between training set and validation set is large when the regularization hyperparameter is small but as we increase hyperparameters, performance difference converges together.



In conclusion, we should select the neural network with the architecture of **(10,10,10) with a lambda of 1** as it produced the highest f-score when tested against the validation set and use this model to test against the Test set. We observe that as we include more layers in the neural network our performance increases.

VII. Conclusion

With the optimal feature transformations and hyperparameters chosen for each model type, each tuned model was tested against the test set to determine final F1 scores for model evaluation. The validation set and test set F1 scores for each model are shown in the table below.

	Logistic Regression	SVM	Neural Network
Feature Transform	Poly3	RBF	(10, 10, 10)
Lambda	100	0.1	1
Validation F1 Score	0.411	0.420	0.418
Test F1 Score	0.346	0.310	0.357

The F1 scores for the tuned models of each type were similar for the validation set, but the tuned neural network performed better than the tuned logistic regression and SVM models against the test set, with the highest F1 score against the test set. Based on the models' relative performance against the test set, the tuned neural network is the superior model. However, all models performed poorly as predictors: the best performing model mustered a precision of 0.360, recall of 0.357, and F1 score of 0.357 against the test set, all of which are barely above the expected values of ⅓ for all three metrics for random guessing given an infinite set of examples.

For all logistic regression, SVM, and neural network models, increasing the regularization value decreased the training F1 score, which can be explained as the increased regularization restricting the ability of the model to fit to the training data. Based on these results, higher regularization increases bias by reducing the ability of the model to fit the training data, while reducing variance by reducing the impact of a particular training set on the model weights. Higher-complexity feature transforms such as higher-degree polynomial transforms, RBF kernels, and larger neural networks had higher training F1 scores compared to lower-complexity feature transforms when not using high amounts of regularization, which can be attributed to overfitting by the models to the training set causing improved performance on the training set.

The validation F1 score for each model increased to a point followed by a marked decrease with increased regularization, as shown in the previous graphs of validation F1 scores vs. regularization values. The region of increasing validation F1 scores corresponds to overfitting by the model to the training data where higher regularization reins in the overfitting, while the region of decreasing validation F1 scores corresponds to underfitting by the model where higher regularization further inhibits the ability of the model to fit the data. The optimal regularization value for each model class lay between the regions of overfitting and underfitting, where the peak validation F1 score was observed. At this optimal point, the rising bias and falling variance associated with higher regularization balance each other out to minimize the out-of-sample error of the model. The regularization value associated with this point therefore produces the model which is most generalizable to data outside the training set, such as the validation and test sets.

The regularization value corresponding to the best validation F1 score for each model class depended on the feature transform used. Generally, higher degree polynomial transforms required higher regularization values to reach their optimal performance on the validation set, but the trend here was not very clear-cut. This trend would be reasonable based on models with higher complexity being more prone to overfitting, necessitating higher regularization values to reach the optimal balance between bias and variance for generalizability.

More complex feature transformations generally produced better classifiers for all three model types, but only to a point. For logistic regression, higher degree polynomial transformations boosted performance up to a degree of 3, but using a degree 4 polynomial transformation saw a small loss in performance. For SVM, higher degree polynomial transformations saw mixed results but RBF, as a highly flexible and non-linear kernel, saw the best results. PCA, which reduces model complexity as part of feature reduction, had the worst results for all SVM model classes. For neural networks, deeper networks and networks with larger layers had improved performance, with the deepest neural network model having the best performance among all trained models. The improved performance for complex models is reasonable based on the highly unstructured data with little in the way of linear correlations or separability: only complex models would be able to find variations for the purpose of classification.

For all model types and classes, extremely high regularization values resulted in the models only predicting a Parkinson's disease stage of 3 regardless of the example used. This behavior was universal among models but the threshold where larger label predictions became predominant varied between models, with less complex models generally reaching this point at lower regularization values. Due to this behavior, the training and validation F1 scores for all models reached the same values of roughly 0.18 with high regularization due to this being the F1 score for predictions of only 3 against the training and validation set.

Since Scikit-learn optimizers run until convergence, the maximum number of iterations was increased for each optimizer until convergence was reached for each model. Additional hyperparameters and optimizers were tested in addition to the ones listed above, but were not used in the final analysis due to not providing better performance. For logistic regression, the limited-memory BFGS algorithm was ultimately used with an L2 norm for regularization, The Newton-CG algorithm with an L2 norm was also tested using the same regularization values, but produced similar results and was not used. The SAGA (Stochastic Average Gradient Accelerated) algorithm with L1 and L2 norms was tested briefly, but produced worse results and ran slowly: since SAGA is optimized for large datasets and our dataset contains only 500 examples, SAGA was ignored.

For neural network training, ReLU was ultimately used as the activation function due to having the best performance and making actual predictions. Tanh performed better for shallow networks but worse in deeper networks against the validation set and a tuned tanh-based model did not perform better than a tuned ReLU-based model. The sigmoid activation performed slightly better for a neural network with one hidden layer of 10 neurons, but performed worse with larger hidden layers and exclusively predicted 3 as the target variable for any model with more than one hidden layer, possibly due to saturation issues with multiple hidden layers. The performance of a tuned sigmoid-based model was substantially worse than the performance using other tuned models.

The Adam (Adaptive Moment Estimation) optimizer was used for neural network training. The limited-memory BFGS optimizer was also tested but had generally worse performance, including on a tuned model. The learning rate was also adjusted from its default value of 0.001, but increasing and decreasing the learning rate both had negative effects on its performance. Finally, all four neural network architectures were rerun with the features derived from PCA used as an input. However, performance with PCA was reduced, which likely was due to the independence of the original features mitigating the benefits of PCA and the loss of information from dimensionality reduction harming optimization.

Among all three models, neural networks yielded the best test F1 score: logistic regression models yielded a slightly lower score and SVMs yielded the worst score by a noticeable margin. The prediction F1 scores were low for all models, which could be explained by the overall lack of structure and near nonexistent correlations between the individual features and the target variable making it difficult to identify characteristics representative of each class for identification. Viewed in this light, the ability of the models to improve upon random chance in their predictions could be viewed as a limited success. Potential improvements to the model include running more neural network architectures, including combinations of deeper networks and larger hidden layers, since the best results from the architectures tested came from the deeper and larger networks. Another improvement would be to generate synthetic data to improve the limited size of the dataset and allow for more training: the lack of relationships between the features would reduce the risk of distortions from using synthetic data. Tuning of the gamma hyperparameter for the RBF kernel is another potential improvement for SVM, especially since the RBF kernel produced the best results out of all trained SVM model classes.

Works Cited

Accuracy, precision, and recall in multi-class classification:

https://www.evidentlyai.com/classification-metrics/multi-class-metrics

Parkinson’s Disease Progression Dataset:

https://www.kaggle.com/datasets/aniruddhawankhede/parkinsons-disease-progression-dataset/data

Scikit-learn User Guide:

https://scikit-learn.org/stable/user_guide.html

Scikit-learn API Reference:

https://scikit-learn.org/stable/api/index.html