CS-UY 4563 Intro to Machine Learning

Final Project Write-Up

Orange Quality Classification

Yirong Wang and Helen Yuan

Introduction

In the pursuit of helping people find the best worth buying orange, our Machine Learning final presentation focuses on the classification of orange quality. This project uses a dataset comprising 241 samples of oranges, categorized by 10 distinct input features to predict the overall quality on a scale from 1 to 5. Employing three machine learning techniques in binary classification, including Logistic Regression, SVM, and Neural Network, this study seeks to discern high-quality oranges, defined as those with a quality rating of 4 or higher, thereby providing valuable insights that could influence consumer purchasing decisions.

Data Preparation

For our project on classifying good or bad oranges, the data preparation phase was carefully designed to optimize the processing and handling of the orange quality dataset. Initially, our dataset comprised 241 samples, each characterized by 10 input features that are instrumental in determining the quality of the oranges. These features were meticulously encoded using one-hot encoding techniques, expanding the original 10 input features to 48 derivative features. This encoding was essential to transform categorical data, which included attributes like color variety and surface blemishes, into a numerical format suitable for machine learning algorithms. The transformation aids in enhancing the interpretability of the model by allowing algorithms to more effectively weigh the importance of different categories in predicting orange quality.

As the project focused on classifying orange quality, a critical step involved in preparing our dataset was the binarization of the output feature. Originally, the quality of the oranges was rated on a scale from 1 to 5. To simplify our classification model and make it more effective at predicting whether an orange is worth buying, we transformed this multiclass system into a binary one. This binarization process involved categorizing oranges with a quality rating of 4, 4.5, or 5 as '1' (indicating good quality and worth buying), and those rated between 1 and 3 as '0' (indicating lesser quality).

To ensure the robustness and accuracy of our model, we adopted a systematic approach to data splitting—a critical step in data preparation aimed at evaluating the model's performance under various scenarios. The dataset was divided into three subsets: 60% was used for training, 20% for validation, and the remaining 20% for testing. This division allows for the comprehensive training of the model while also providing a mechanism to fine-tune the parameters and prevent overfitting. Overfitting occurs when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. By having separate validation and testing subsets, we can iteratively improve our model's settings during the validation phase before finally evaluating its generalized performance on unseen test data. Such meticulous data management ensures that the model remains predictive and reliable when applied to real-world scenarios outside the scope of the training dataset.

Logistic Regression

Our model used on approaching the orange classification problem is Logistic Regression which operates by estimating probabilities using a logistic function. The Scikit-Learn python library was used to perform the training. We first trained the model with different strength c values (0.001, 0.01, 0.1, 0.5, 1, 5, 10) separately for L1 and L2 regularization. In L1 regularization we found the best strength value is 5, having the highest validation F1 value 0.8615384615384615 (Figure 1). In L2 regularization the best strength is 0.5, having the best validation F1 value 0.911764705882353 (Figure 2). Therefore, we picked L2 regularization with strength 5 for our final model.

Table 1: Train/Validation F1 Score for Different L1 Regularization Strength (Logistic Regression)

Regularization Strength	Train F1 Score	Validation F1 Score
0.001	0.7848101265822784	0.8
0.01	0.7848101265822784	0.8
0.1	0.8659793814432989	0.8358208955223881
0.5	0.9042553191489361	0.8615384615384615
1	0.925531914893617	0.8615384615384615
5	**0.9361702127659575**	**0.8615384615384615**
10	0.9312169312169313	0.8615384615384615

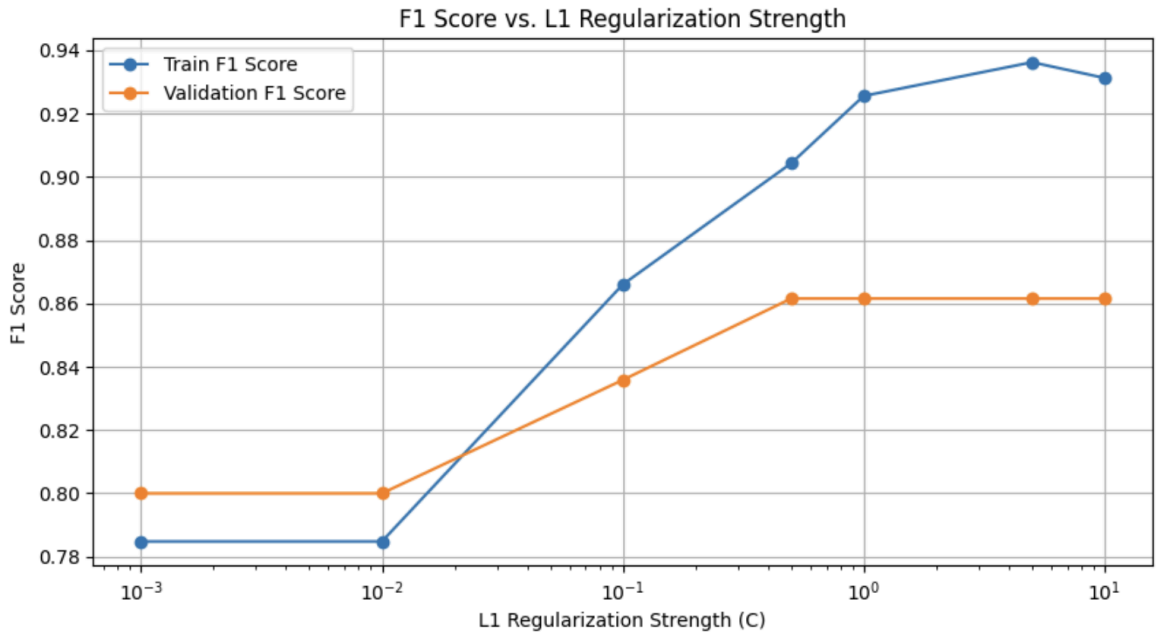

Figure 1: F1 Score vs. L1 Regularization Strength

Table 2: Train/Validation F1 Score for Different L2 Regularization Strength (Logistic Regression)

Regularization Strength	Train F1 Score	Validation F1 Score
0.001	0.7848101265822784	0.8
0.01	0.8910891089108912	0.8378378378378379
0.1	0.9042553191489361	0.8823529411764706
0.5	**0.9206349206349206**	**0.911764705882353**
1	0.9304812834224598	0.8955223880597014
5	0.9312169312169313	0.8615384615384615
10	0.9312169312169313	0.8615384615384615

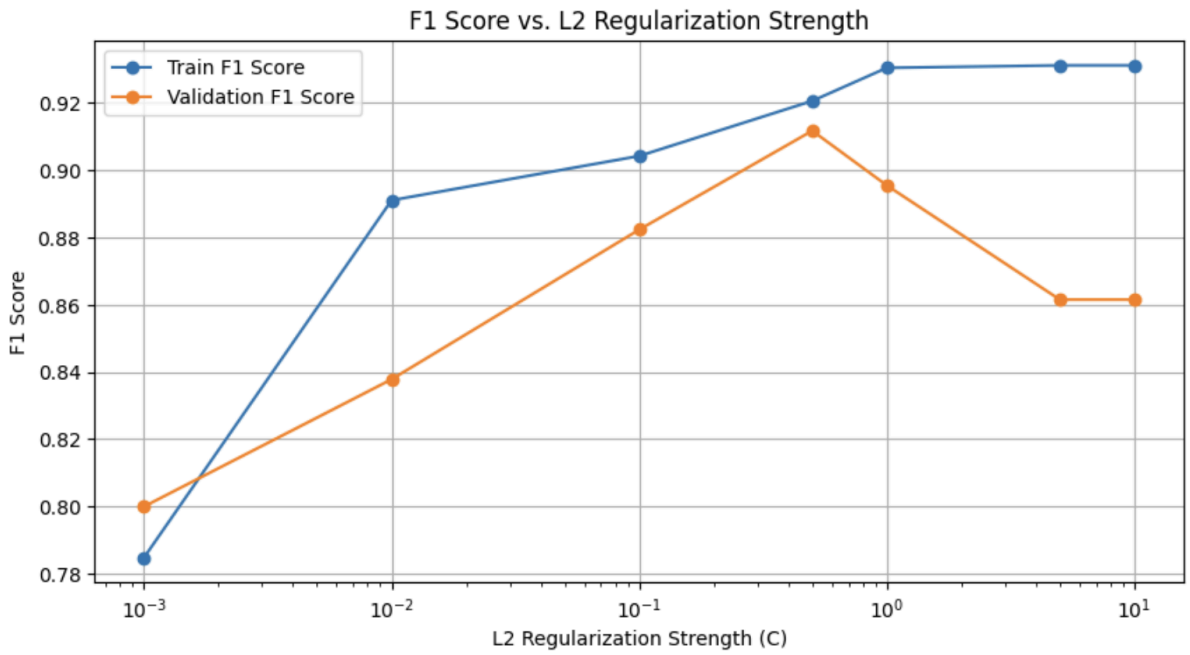

Figure 2: F1 Score vs. L2 Regularization Strength

In the context of our logistic regression model for classifying orange quality, selecting an appropriate threshold is crucial for determining the final classification of an orange as either high-quality (worth buying) or not. In training L1 and L2 regularization for logistic regression, we used 0.5 threshold to determine the output prediction. Since this may not be the best performance threshold, the next step we trained on threshold value. We trained on threshold values from 0.1 to 0.9 incremented by 0.05 each time. It turns out that 0.5 is the best value for this orange classification with the highest validation value 0.8985507246376812 (Figure 3). Thus, we picked the probability threshold value as 0.5 for our final model.

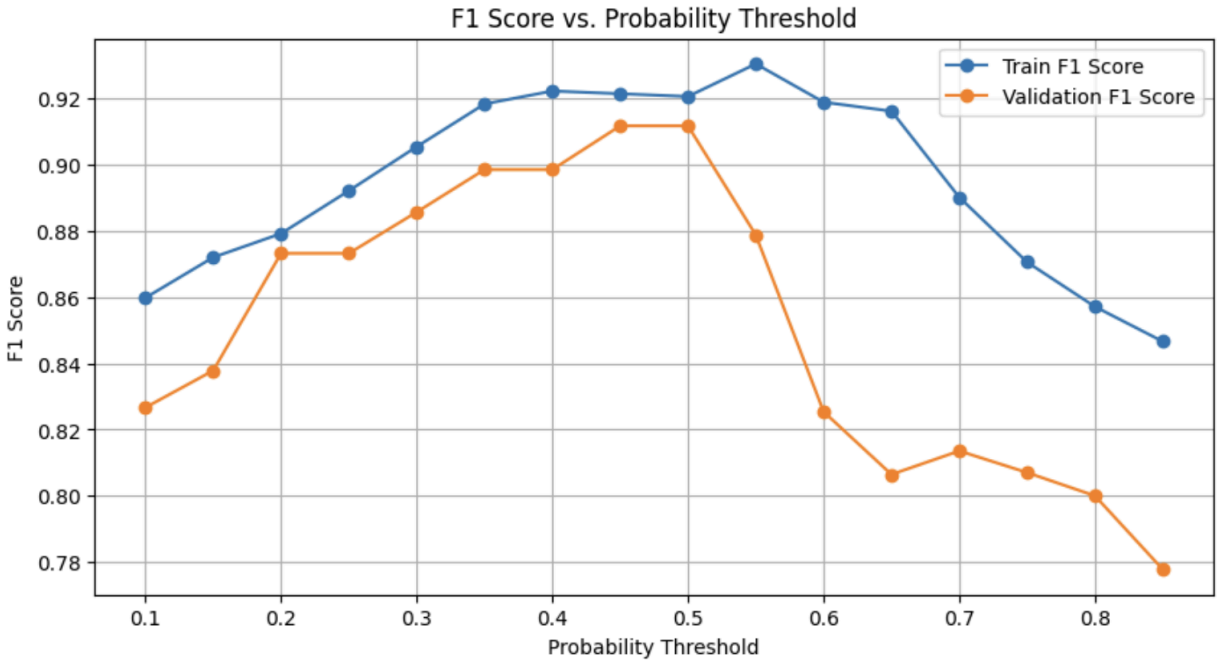

Figure 3: F1 Score vs. Probability Threshold Value

In our logistic regression model for classifying orange quality, feature transformation played a pivotal role in enhancing the model's predictive power and accuracy. For our project, we implemented a second-degree polynomial transformation on the features, a technique that allows the model to consider not only the linear relationships but also the interactions between different features. We tested the performance of the transformation on both training set and validation set. Compared with F1 score before transformation, the model worked better having the second-degree polynomial feature transformation (Table 1). Thus, we picked the second-degree transformed features for the final model.

Table 3: w/wo Transformation vs. Training and Validation F1 Score (Logistic Regression)

	With Transformation	Without Transformation
Training set F1 Score	0.9946524064171123	0.9206349206349206
Validation set F1 Score	**0.9253731343283582**	0.911764705882353

Finally, to test on how well our model predicted, we combined all best performing parameters together to find F1 score for the test set. Using L2 regularization with strength value 0.5, second-degree feature transformation, and probability threshold value 0.5, we got the test set F1 Score: 0.84375. This F1 score indicated a good performance of our logistic regression model.

Support Vector Machine

The second model we used to approach the orange classification problem is Support Vector Machines, which operates by finding the optimal hyperplane that best separates different classes in the feature space, maximizing the margin between the closest data points of different classes. We use the Scikit-Learn python library to apply and experiment the Support Vector Machines with regularizations on both the linear and the RBF kernel.

We first trained the SVM model with different regularization strengths (0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000) using the linear kernel. As shown in Figure 4 below, we found the best regularization strength of 0.1 when applying on the linear kernel, with the highest validation F1 score of 0.8888888888888888 as shown in Table 4.

Table 4: Train/Validation F1 Score for Different Regularization Strength (Linear Kernel)

Regularization Strength	Train F1 Score	Validation F1 Score
0.0001	0.7848101265822784	0.8
0.001	0.7848101265822784	0.8
0.01	0.8994708994708994	0.8611111111111113

0.1	**0.9197860962566845**	**0.8888888888888888**
1	0.9297297297297297	0.8571428571428571
10	0.9368421052631579	0.8852459016393444
100	0.9417989417989417	0.870967741935484
1000	0.9417989417989417	0.870967741935484
10000	0.9417989417989417	0.870967741935484

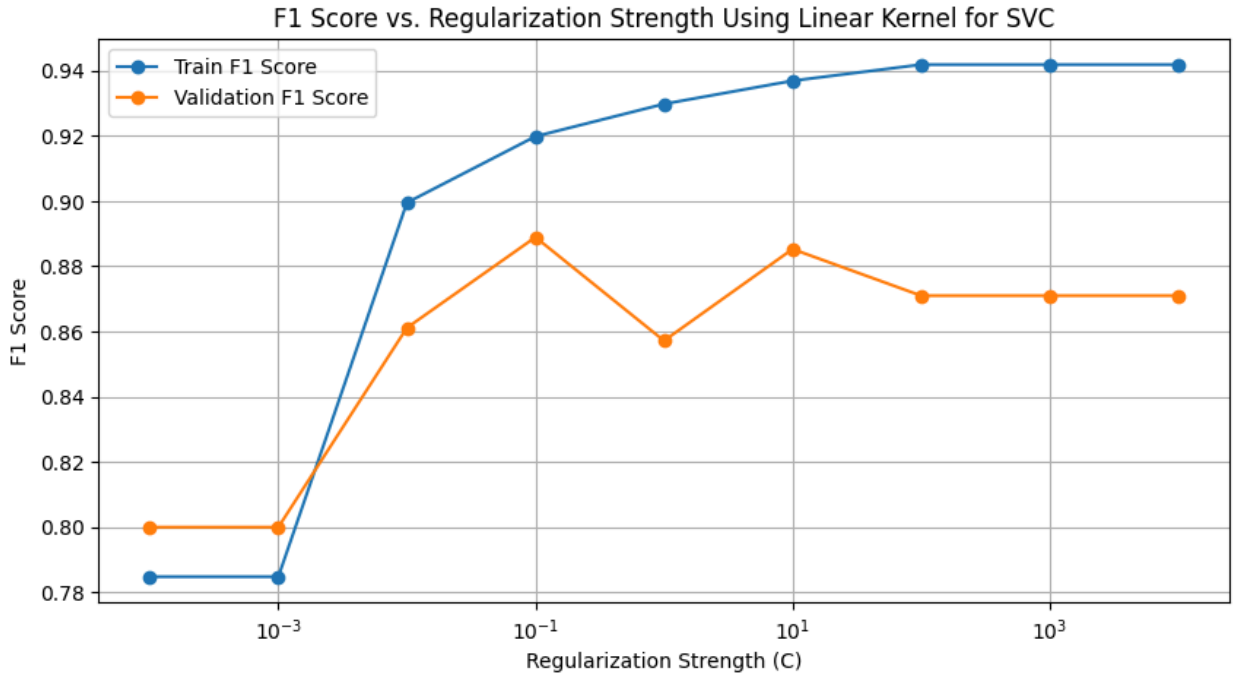

Figure 4: F1 Score vs. Regularization Strength Using Linear Kernel

We then trained the SVM model with different regularization strengths (0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000) using the RBF kernel. As shown in Figure 5 below, we found the best regularization strength of 10 when applying on the RBF kernel, with the highest validation F1 score of 0.8823529411764706 as shown in Table 5.

Table 5: Train/Validation F1 Score for Different Regularization Strength (RBF Kernel)

Regularization Strength	Train F1 Score	Validation F1 Score
0.0001	0.7848101265822784	0.8
0.001	0.7848101265822784	0.8
0.01	0.7848101265822784	0.8
0.1	0.7848101265822784	0.8
1	0.9417989417989417	0.8611111111111113
10	**1.0**	**0.8823529411764706**
100	1.0	0.8823529411764706
1000	1.0	0.8823529411764706
10000	1.0	0.8823529411764706

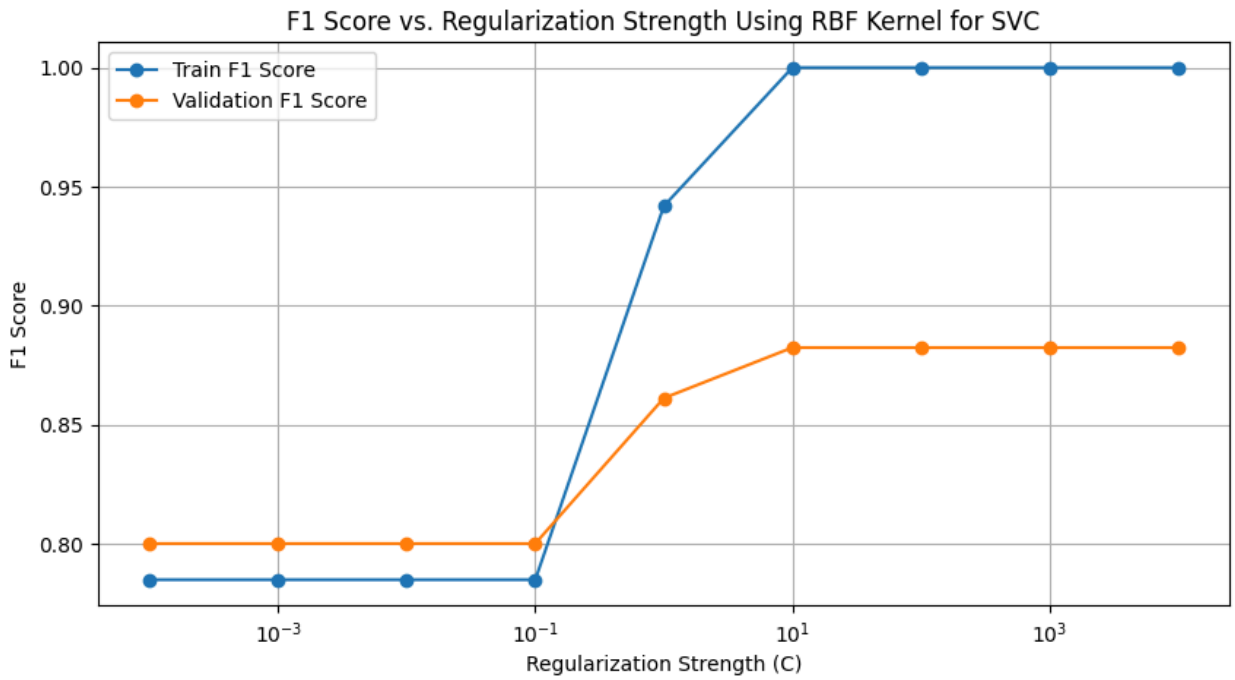

Figure 5: F1 Score vs. Regularization Strength Using RBF Kernel

According to the best performing F1 Score shown above, we select the regularization strength of 0.1 with the linear kernel to test how well our model performed on the test set. With this best parameter setting, we got a test set F1 score of 0.923076923076923.

Neural Network

The third model we used to approach the orange classification problem is Neural Network, which utilizes interconnected layers of nodes, or neurons, to learn complex patterns and relationships in the input data, ultimately providing a nonlinear decision boundary for classification. We use the Keras library from Tensorflow to apply a 7 layer neural network to our data set. Our neural network has an input layer with 144 neurons, 5 hidden layers with 128, 64, 32, 16, and 8 neurons, and finally an output layer with 1 neuron. For the 5 hidden layers, we choose to use the ELU activation function, and for the output layer, we choose to use the Sigmoid activation function. We choose the ELU activation function for the 5 hidden layers because ELU activation functions have nonzero gradients for all inputs, unlike Sigmoid activation functions, the vanishing gradient problem is less likely to occur when using ELU activation function. Since ELU activation function allows negative values, using ELU for our hidden layers also ensures that all our neurons remain active and continue to learn (contribute to updating the weight). On the other hand, we choose the sigmoid activation function for our final output layer because it well suits our needs of binary classification, where it outputs a probability within the range of 0 and 1.

Table 6: Neural Network Structure

	# Of Neurons	Activation Function

Hidden Layer 1	128	ELU
Hidden Layer 2	64	ELU
Hidden Layer 3	32	ELU
Hidden Layer 4	16	ELU
Hidden Layer 5	8	ELU
Output Layer	1	Sigmoid

We first train the Neural Network with a different learning rate (0.0001, 0.001, 0.005, 0.01, 0.1, 0.5) using L2 regularization with a strength value of 0.001. As shown in Figure 6 below, we found the best learning rate of 0.001 when applying the same L2 regularization, with the highest validation F1 score of 0.8985507246376812 as shown in Table 7.

Table 7: Train/Validation F1 Score for Different Learning Rate

Learning Rate	Train F1 Score	Validation F1 Score
0.0001	0.7759562841530053	0.6774193548387096
0.001	**0.9021739130434783**	**0.8985507246376812**
0.005	0.9732620320855615	0.8955223880597014
0.01	0.9891304347826088	0.8484848484848485
0.1	0.7848101265822784	0.8
0.5	0.0	0.0

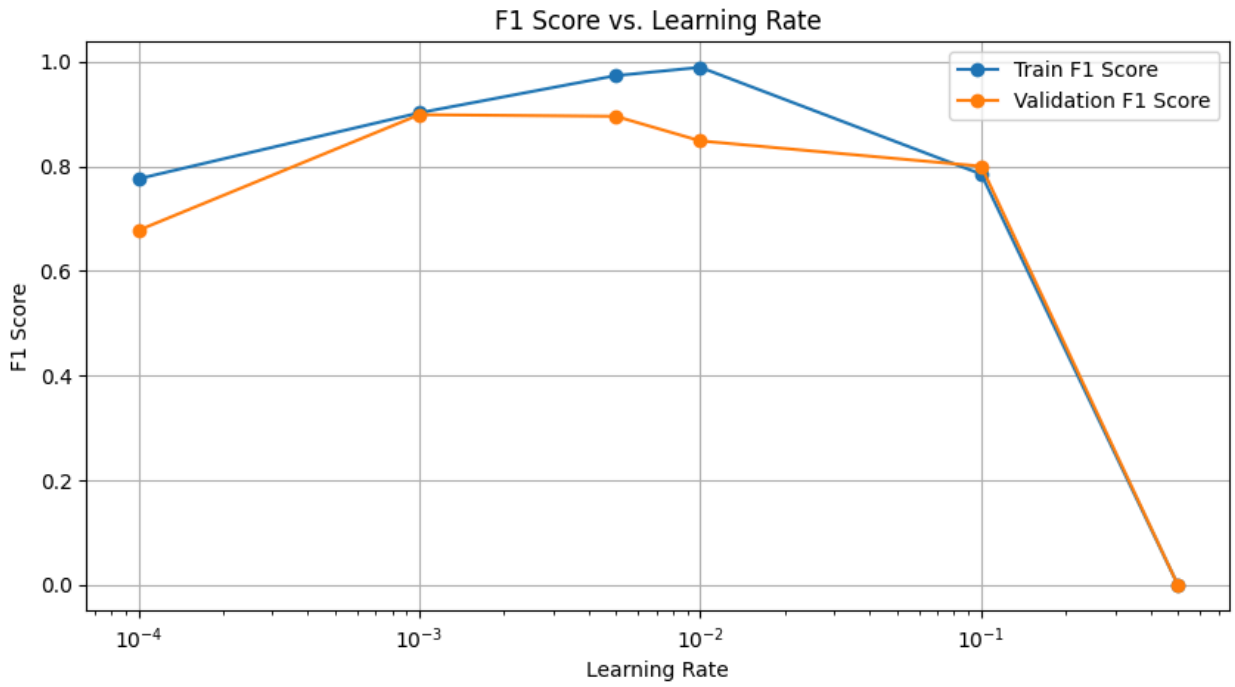

Figure 6: F1 Score vs. Learning Rate for Neural Network

We then train the Neural Network with different L2 regularization strength values (0.0001, 0.001, 0.005, 0.01, 0.1, 0.5) with the best learning rate of 0.001 we got above. As shown in Figure 7 below, we found that the best strength value for L2 regularization is 0.001, with the highest validation F1 score of 0.8857142857142857 as shown in Table 8.

Table 8: Train/Validation F1 Score for Different L2 Regularization Strength

Regularization Strength	Train F1 Score	Validation F1 Score
0.0001	0.9081081081081082	0.8405797101449275
0.001	**0.925531914893617**	**0.8857142857142857**
0.005	0.8936170212765957	0.8181818181818182
0.01	0.9090909090909091	0.8358208955223881
0.1	0.9411764705882354	0.823529411764706

0.5	0.9239130434782609	0.8656716417910447

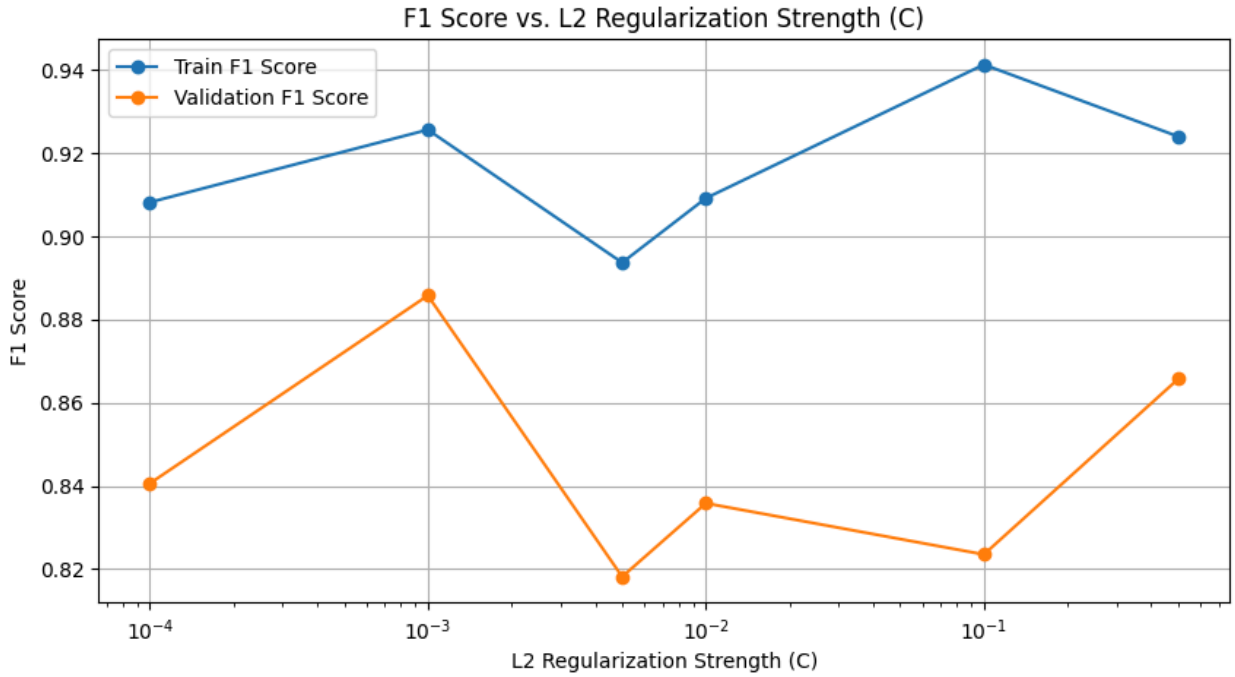

Figure 7: F1 Score vs. L2 Regularization Strength for Neural Network

Lastly, we then train the Neural Network with different L1 regularization strength values (0.0001, 0.001, 0.005, 0.01, 0.1, 0.5) with the best learning rate of 0.001 we got above. As shown in Figure 8 below, we found that the best strength value for L1 regularization is 0.1, with the highest validation F1 score of 0.8985507246376812 as shown in Table 9.

Table 9: Train/Validation F1 Score for Different L1 Regularization Strength

Regularization Strength	Train F1 Score	Validation F1 Score
0.0001	0.9312169312169313	0.8823529411764706
0.001	0.9042553191489361	0.8695652173913043
0.005	0.9139784946236559	0.8695652173913043

0.01	0.9239130434782609	0.8823529411764706
0.1	**0.9139784946236559**	**0.8985507246376812**
0.5	0.9304812834224598	0.8857142857142857

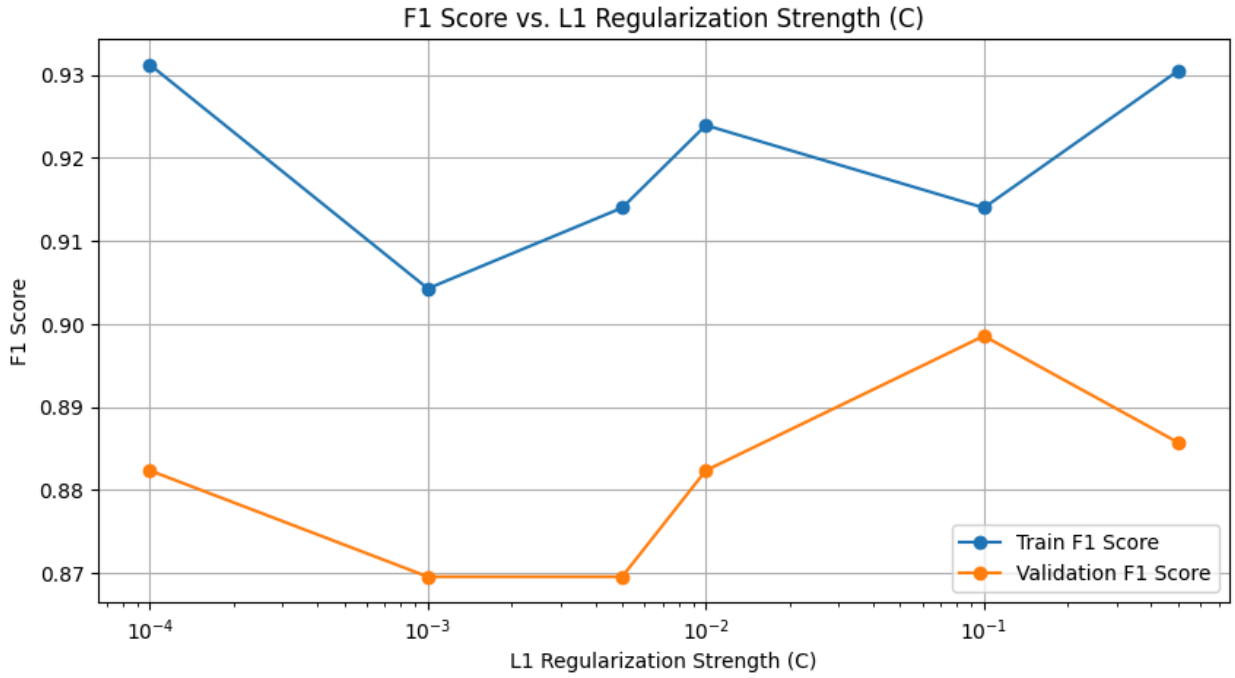

Figure 8: F1 Score vs. L1 Regularization Strength for Neural Network

After we have trained and selected the best parameter settings using our validation set, we also want to see if implementing a second-degree polynomial feature transformation will help improve our Neural Network model, just like how it helped in Logistic Regression. We train the Neural Network on our data both with and without second-degree transformation, using the best parameter settings with learning rate of 0.001 and L1 regularization with strength value of 0.1. According to the F1 Score we got shown in Table 10 below, the second-degree polynomial feature transformation did not

help in the case of Neural Network, and our model actually performed better with a higher validation F1 score of 0.8695652173913043 without transformation.

Table 10: w/wo Transformation vs. Training and Validation F1 Score (Neural Network)

	With Transformation	Without Transformation
Training Set F1 Score	0.9304812834224598	0.9247311827956989
Validation Set F1 Score	0.8169014084507042	**0.8695652173913043**

Going through the training and model selection process using our training and validation set, we select the L1 regularization with strength of 0.1 and the learning rate of 0.001 to test how well our model performed on the test set. With the best parameter settings for our Neural Network, we obtained a test set F1 score of 0.9253731343283582.

Conclusion

During the data preparation stage, we divided the dataset into three subsets: training, validation, and testing, allowing us to monitor overfitting and underfitting effectively. In the Logistic Regression model, extensive regularization tuning demonstrated how stronger regularization (L2 with lambda 0.5) mitigated overfitting, leading to a balanced validation F1 score. Meanwhile, the Support Vector Machine (SVM) model showcased a similar balance when we identified an optimal regularization strength of 0.1 for the linear kernel and 10 for the RBF kernel. The neural network model, however, showed sensitivity to learning rates and regularization strengths, demonstrating overfitting tendencies at higher values, which were countered through validation monitoring.

Variance and bias were evident in our models' performances. The logistic regression model's performance variations across different thresholds highlighted the need for balanced parameter tuning to avoid high variance. The SVM model's results underscored this balance as well, with linear kernel regularization at 0.1 offering consistent results, indicating lower variance and bias. For the neural network model, a mix of high variance and low bias was balanced through L1 regularization with lambda 0.1, yielding a reliable test set F1 score.

The inclusion of a second-degree polynomial transformation in the Logistic Regression model proved beneficial, indicating interactions among features and reducing underfitting by allowing models to capture more complex relationships. In contrast, this transformation didn't yield similar benefits for the neural network, reinforcing the importance of evaluating feature transformations' impact case by case.

In conclusion, our project successfully implemented three machine learning models (Logistic Regression, Support Vector Machines (SVM), and Neural Network) and achieved notable success in classifying orange quality. These models, through extensive parameter tuning and feature transformation, were able to distinguish between high-quality and low-quality oranges with reasonable performance, as demonstrated by the F1 scores for each model (Table 11). The SVM and neural network models showed consistent high performance, both with F1 scores exceeding 0.92, particularly when balanced between variance and bias through regularization and learning rate tuning. The logistic regression model's performance benefited from a

second-degree polynomial transformation, illustrating how feature transformations can enhance model effectiveness in specific cases.

Table 11: Test Set F1 Score Across Models

Logistic Regression	SVM	**Neural Network**
0.84375	0.923076923076923	**0.9253731343283582**

Works Cited

Original Orange Dataset:

https://www.kaggle.com/datasets/shruthiiiee/orange-quality/data

Scikit-Learn Logistic Regression Documentation:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Scikit-Learn Support Vector Classification Documentation:

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Documentation for Tensorflow:

https://www.tensorflow.org/tutorials/quickstart/beginner