CP322: Mini project 1

Alexandros Ioannou, Duc Nguyen Minh, Florian Novak, Saumya Patel

I. Abstract

In this group project, we have developed two classification algorithms, Logistic Regression and K-Nearest Neighbors (KNN), and used it on four benchmark datasets: Ionosphere, Adult, US Cars and Iris dataset. The main focus of the project was to evaluate and compare the performance of these classification algorithms in terms of accuracy and training efficiency. Before creating the models, cleaning, tidying and analyzing of the datasets was performed in order to have a better understanding of the features within each dataset. In the next step, for implementing the two algorithms, the optimal hyperparameters were chosen after the data processing and model implementation. Based on our experiments, the logistic regression (a=0.01) achieved accuracies of approximately 75.89% on the Adult dataset, 98.59% on the Ionosphere dataset, 95% on the US cars dataset, and struggled with the Iris dataset with an accuracy close to 0%. On the other hand, KNN (k=5) achieved accuracies of approximately 80.06% on the Adult dataset, 97.18% on the Ionosphere dataset and 80% on the Iris dataset. For the US cars dataset, KNN showed an optimal accuracy of 97%.

II. Introduction

Our project can be divided into four different stages of work for every dataset plus the development of the classification algorithms in general. First, we managed to load all four different datasets into Kaggle with the help of the pandas library. In the next step, every dataset had to be checked and cleaned. Moreover, for each dataset, some specific cleaning and transforming processes were necessary to be able to fully work with the data. Next, we created the basic statistics for every dataset to be able to better understand all the features and to have a good understanding of the classification problem before developing the models. For the Adult dataset, our task was to predict whether a person makes over 50k a year. For the Ionosphere dataset, the goal was to predict whether a radar return from the ionosphere is ‘good’ or ‘bad. For the Iris dataset, our task was to predict which type of Iris Species the given flower’s information. For the US Cars dataset, our task was to predict whether a car has a clean title or has been under salvage insurance. After all the preparation, we started to develop our two classification algorithms: the Logistic regression function and the KNN. After finishing the two algorithms, we split up every dataset into a training and testing set.

III. Datasets

- For the Ionosphere dataset, The target variable, 'label,' was mapped to binary values. Moreover, an exploratory data analysis of the data revealed that the second column contained constant values, making it redundant. This column was consequently removed.
- For the Adult dataset missing values denoted by '?' were replaced with NaN and subsequently dropped. After thorough feature selection, we have decided to use Age,

Work Class, Education, Occupation, Race, Sex and Marital Status. Our primary target was the ‘income’ column, indicating whether an individual earns more than 50k a year.

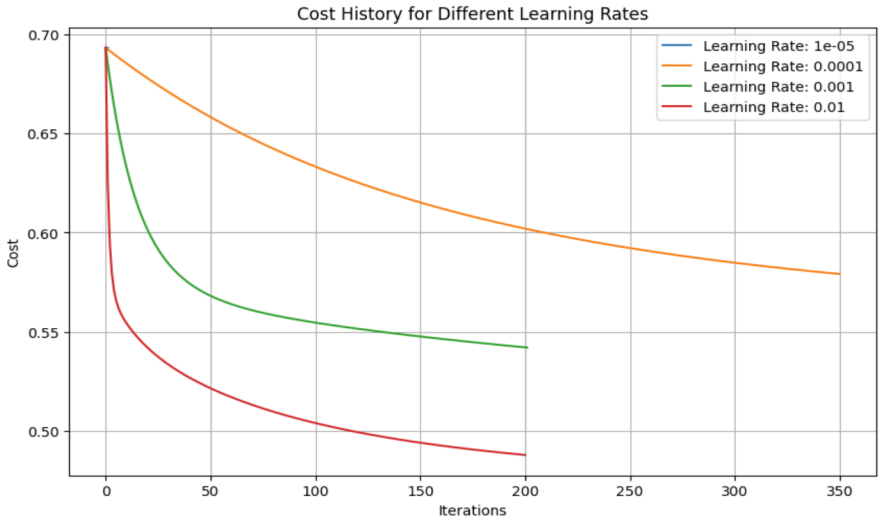
- The Iris dataset required minimal preprocessing, as it already possesses numerical attributes. Exploratory analysis involved computing basic statistics, such as mean and standard deviation, to understand the feature characteristics better.
- The US Cars dataset cleansing involved removing an unnecessary 'Unnamed: 0' column. Furthermore, the 'mileage' attribute was cast as an integer. A key feature of this dataset was the binary classification task for 'title_status,' classifying titles as 'clean vehicle' (1) or 'salvage insurance' (0). The dataset, therefore, demanded the encoding of this categorical feature. Moreover, some of the data was drop such as (‘brand’, ‘model’, ‘color’, ‘state’, ‘condition’)

IV. Results

Adult Dataset

A comparison of the two models reveals that the KNN model, with an optimal k value, performed slightly better than logistic regression in terms of accuracy on the specific dataset.
Logistic Regression:

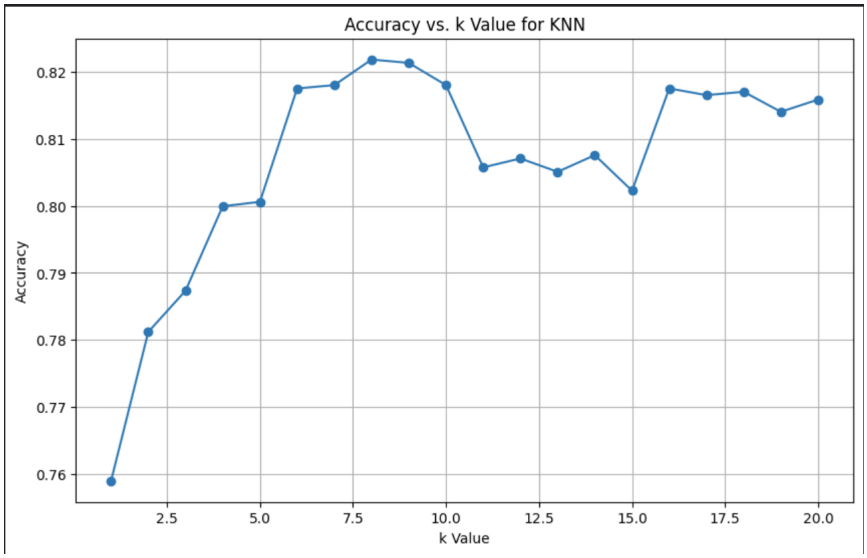
- Achieved an accuracy of approximately 75.89% on the test set
- The average accuracy from k-fold cross-validation was approximately 75.39%



- Here a learning rate of 0.001 (green curve) seems optimal as it decreases rapidly without diverging. The red curve (0.01) shows an even steeper decrease in cost and seems to be diverging. Orange and blue curves show slow convergence due to small learning rates.

KNN:

- Achieved an accuracy of approximately 80.06% on the test set
- For k-fold cross-validation we tested different k values (k=1 to 10) and the model performed best with k=9, with an accuracy of approximately 80.40%
- For the accuracy we have also tested different k values (k=1 to 20):

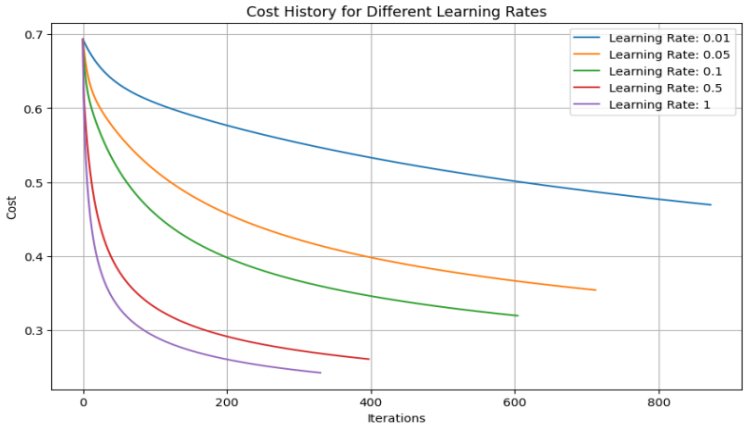


- The accuracy seems to be increasing steadily until the 9th k-value and then from there it dips approximately 1-2% until the 15th k-value and then increases from there on. While increasing k initially leads to a more stable model by reducing sensitivity to noise, there's a tipping point beyond which the model starts losing accuracy. It appears that the optimal k value is 9.

Ionosphere Dataset

Both models worked very well for this dataset. Logistic Regression slightly edging out KNN.
Logistic Regression:

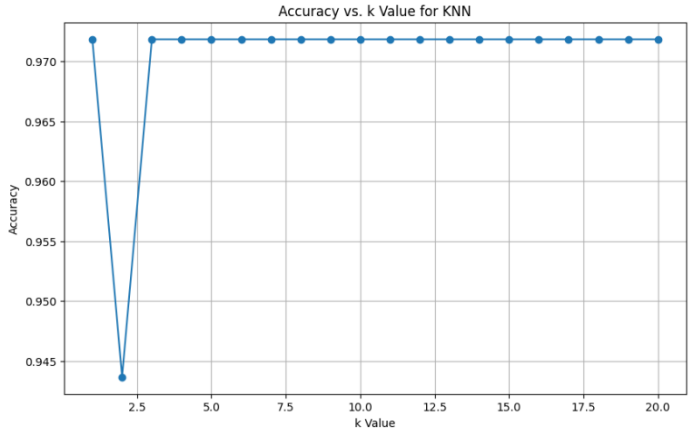
- Achieved an accuracy of about 98.59%.
- The average k-fold cross validation was around 81.48%.



- Plotting the cost history out shows us that at a learning rate of 1, the model converges the quickest and stabilizes while the learning rate of 0.01 is the worst.

KNN:

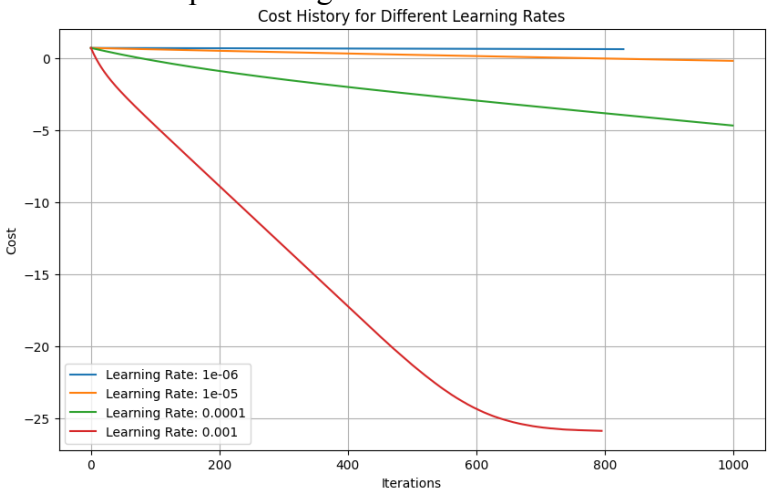
- Achieved an accuracy of approximately 97.18%.
- For k-fold cross validation the k value of 2 gave us the highest accuracy of about 90%.



- This plot here shows the accuracy of the corresponding k value. It shows that for all values of k except 1, the accuracy is approximately just above 97%, let's say 97.18%, which is the approximate accuracy for k = 5. This is obviously not correct, as the accuracy can't stay the same for more than one k value. There will be a difference. Thus, this dataset is better suited for logistic regression.

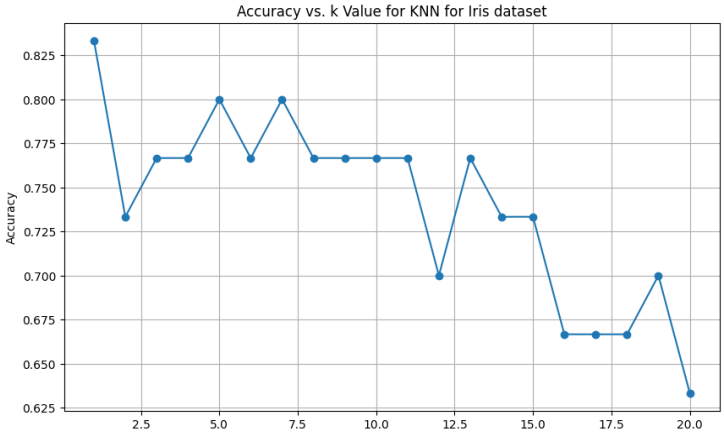
Iris Dataset

- Logistic Regression:
  - Having the graph for cost history plotted out, we could see that at a learning rate of 0.001, the model is learning effectively at the start. However, the stabilization of the cost would imply that the model has reached its optimal performance for this learning rate or may have settled in a local minimum
  - Given that the accuracy is 0%, it's clear that the logistic regression model is not performing well on this dataset.




```
predictions:  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Accuracy:  0.0
The average accuracy from k-fold cross-validation is: 0.3333333333333333
```

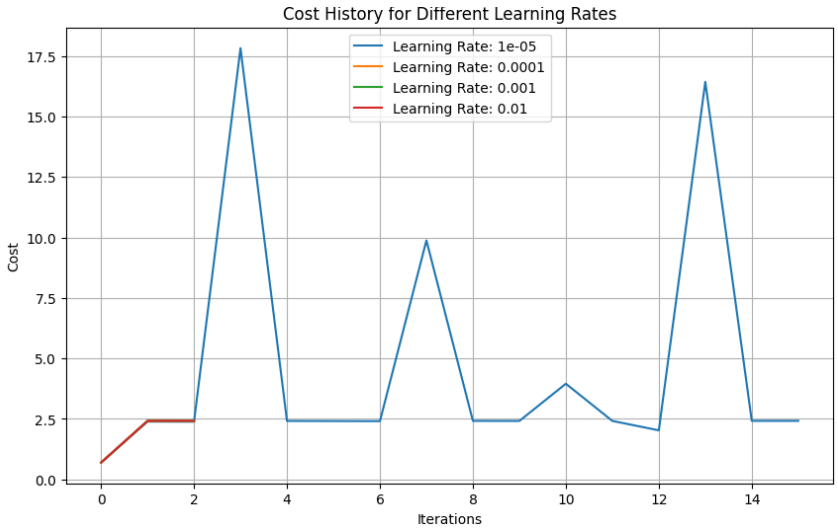
- KNN classification:
    - Our observations show that the accuracy peaks at k-values of approximately 4 and 7, reaching up to 80%. The k-fold test yielded an accuracy of nearly 97%. Based on these results, we believe that this model is particularly well-suited for the Iris dataset.



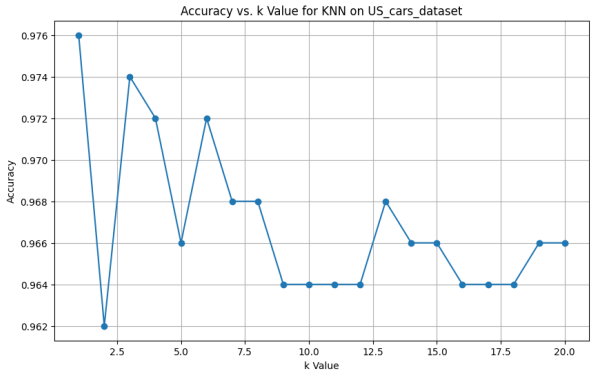

```
KNN Accuracy:  0.8
CV Accuracy for k=1: 0.9666666666666666
CV Accuracy for k=2: 0.9666666666666668
CV Accuracy for k=3: 0.975
CV Accuracy for k=4: 0.9583333333333334
CV Accuracy for k=5: 0.975
```

US Cars Dataset

- Logistic Regression:
    - We experimented with various learning rates, specifically {0.00001, 0.0001, 0.001, 0.1}. At a learning rate of 0.00001, we observed a training accuracy of 95% but a k-fold accuracy of 65%. This difference highlights potential overfitting. Given these findings, we think that Logistic Regression might not be the best choice for this data.



- KNN classification:
    - We evaluated the KNN model's performance using various values of K. We found that the model achieved optimal results with K=1 and K=3. However, I'd lean towards selecting K=3 to avoid potential overfitting that can occur with K=1. With this choice, we attained an accuracy of 97% and the k-fold is roughly 95%




```
KNN Accuracy: 97.39999999999999%
Accuracy for k= 94.99799919967987%
Accuracy for k= 95.7983193277311%
Accuracy for k= 95.71828731492596%
Accuracy for k= 95.6782713085234%
Accuracy for k= 95.2781112444978%
Accuracy for k= 95.19807923169269%
```

V. Discussion and Conclusion

One key takeaway from our analysis was the influence of the learning rate on Logistic Regression's performance. The learning rate played a pivotal role in affecting the convergence speed of the model. By carefully selecting the learning rate, we were able to achieve faster convergence while maintaining stability during training, highlighting the critical role of hyperparameter tuning in optimizing Logistic Regression. In the case of the KNN algorithm, we performed an exhaustive search to identify the optimal value of k, which represents the number of nearest neighbors to consider.

Our experiments across the datasets varied in performance for both algorithms. For the Ionosphere dataset, logistic regression slightly outperformed KNN. For the Adult dataset, KNN had a higher accuracy than logistic regression. The Iris dataset had some challenges for logistic regression, but KNN managed to achieve a high performance. For the US cars dataset, KNN had a slightly higher accuracy than logistic regression.

VI. Statement of Contributions

- **Alexandros Ioannou:** Assisted with the cleaning and tidying of the Adult Dataset. Implemented the KNN algorithm and assisted with the Logistic Regression algorithm. Implemented training and testing of the Adult dataset. Contributed to the relevant information in the writeup regarding the Adult dataset.
- **Duc Nguyen Minh:** Assisted and implemented K-fold tests for Logistic Regression and KNN algorithm, helped clean the adult datasets and US cars, evaluated the accuracy and prediction for Iris and US car dataset, and contributed to the project write up
- **Florian Novak:** Contributed to Logistic Regression. Responsible for project writeup.
- **Saumya Patel:** Implemented log regg, cleaned and split Ionosphere dataset, ran statistics for the datasets, assisted in cleaning, testing, and plotting the datasets, and writeup.