K-Nearest Neighbors, Support-Vector Machine, Logistic Regression, and Convolutional Neural Network on Fashion MNIST

Adilkhan Bakridenov, Moldir Berkaliyeva, and Darya Taratynova

***Abstract*—Online Fashion Market considered to be one of the instantly growing online industries and a great area for implementing artificial intelligence tools. This tools able to understand online customers' behavior in online-store and also consider their preferences, and with this data online-stores can to improve user experience and sales. The aim of our research article is to design 4 various machine learning models and compare their performance on image classification of clothes. In this Research paper the Fashion-MNIST was used as a dataset, while K-Nearest Neighbors, Support-Vector Machine, Logistic Regression, and Convolutional Neural Network(CNN) were used as machine learning algorithms.The difference between CNN and 3 other models is that CNN is a large multi layered Neural Network model that requires significant amount of data and powerful GPUs to train itself, however these models achieve considerably high accuracy at some problems. After conducting the data pre-processing and model training parts, we started to compare the test accuracy of each model. Analyzing the results, we can conclude that CNN model has the best performance on classifying the Fashion-MNIST with almost 0.93 accuracy.**

I. Introduction

We can apply different machine learning algorithms to multi-class classification problems, including standard algorithms and neural networks. One neural network used in the multi-class classification of clothes is Convolutional Neural Network (CNN). CNN is a powerful tool in image and pattern recognition, as it was shown [1], using a back-propagation algorithm. Fashion MNIST is served as a benchmark for testing algorithms. The difficulty of this problem is related to the main objects of data - clothes. Some of the types of clothes have similar features. This makes the problem of multi-class classification deep to test machine learning algorithms. Fashion MNIST dataset contains 70 000 images (28x28 pixels), with ten classes: T-shirt/Top, Trousers, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle Boot [2]

In this paper, we perform model selection and investigate the difference in generalization performance of CNN and standard algorithms, SVM, K-Nearest Neighbors, and Logistic Regression.

This report is organized as follows: Section 2 gives an overview of the dataset and used methods, Section 3 presents our results, and Section 4 concludes our results.

II. Methods

A. Principal Component Analysis

The main reason behind using Principal Component Analysis (PCA) during SVM training is to reduce dimensionality, losing the least possible amount of information. In this data exploratory data analysis, we choose a few principal components. In our case, out of 784 dimensions, we take the first 50 principal components. From Fig 1, we see that the optimal number of principal components is about 25. We will use it in the training of SVM in pipelining.

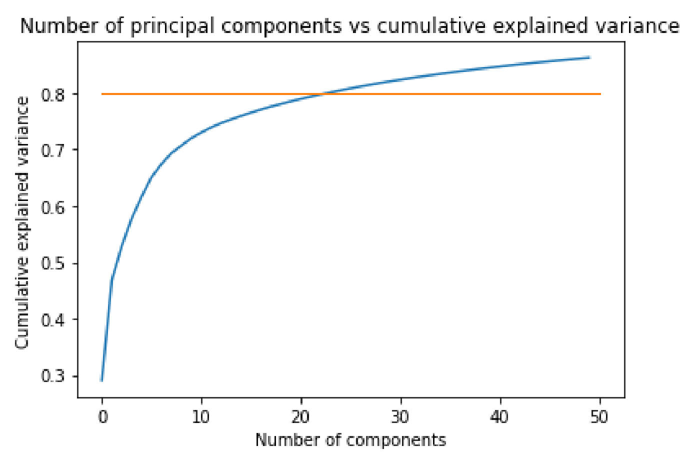

Fig. 1. Figure 1.Graph of number of components vs cumulative explained variance. Optimal number of principle components is 25

B. K-Nearest Neighbors

K-Nearest Neighbors (KNN) clusters points together based on distance metric: Manhattan, Euclidean, and Minkowski distance [3]. We set the value of K to 12 to avoid overfitting. Then we run the algorithm using three different metrics. The choice of best is primarily based on the accuracy of the model. Also, we decided to use a confusion matrix plotted with seaborn to look at two additional metrics: Precision and Recall, because accuracy is not always the best indicator of a good model. Data are split to 60000 training and 10000 testing.

C. Support-Vector Machine

Support-Vector Machine (SVM) works by defining a separating hyperplane. As a Regularization parameter (C), we have chosen the middle one, not to overfit or underfit our data, C = 100, and gamma = 0.1 for Radial Basis Function (RBF). In the case with the polynomial kernel, kernel trick, we have tested two variants with higher degree = 5 (higher complexity and more chances for overfitting) and with average degree = 3. Data are split to 60000 training and 10000 testing.

D. Logistic Regression

In the Logistic Regression algorithm, we divide data 80 percent for training and 20 percent for testing. Then we use Keras Sequential model with softmax activation function, stochastic gradient descent as an optimizer, categorical cross-entropy as loss function, and accuracy as metrics. We also use the loss function, cross-entropy, because we want our algorithm to predict probabilities. It should help us in the backpropagation step because we need to minimize the loss value.

E. CNN

It can be said that CNN model extracts the feature of image and then convert it into lower dimension avoiding the loss of main characteristics.During the designing the CNN model we need keep the representation power alive. In other words, when we reduce the image dimension, we should increase the number of filters by the same multiplier.For example, if we have 64 filters for a layer reducing 112x112 feature map to 56x56, we need to use 128 filters for the forthcoming layer.By this way we will increase the number of filters in Conv2D layers. Convolution layers contains ReLU activation to make all negative value to zero. At the last layer of our CNN model, there will be 10 neurons for our 10 classes with ”Softmax ”activation fucntion. Because softmax function is used for the multiclass logistic regression, while the sigmoid function is used for the two-class logistic regression. For loss function we used categorical cross-entropy, while for optimization algorithm we used adaptive learning rate optimization(Adam).Accuracy was our main metrics.In CNN training, we track test loss and test accuracy and decide further manipulations with layers accordingly.

F. Choosing one model

After working on each model independently, we compare their accuracy and decide which is better to use as final. Then we compare our results with top results on the Kaggle competition, and if there are results better (more than 1 percent), we revisit our algorithm. We will continue all steps until we get the best-performing model.

III. Results

Model Performance		
Model Type	Test Accuracy	Test Loss
12-nearest neighbors with manhattan metrics	0.8568	-
12-nearest neighbors with euclidean metrics	0.8471	-
12-nearest neighbors with minkowski metrics	0.8471	-
SVM with rbf	0.8638	-
SVM with poly and degree 3	0.8801	-
SVM with poly and degree 5	0.8687	-
Logistic Regression	0.8576	0.4197
CNN	0.9290	0.2039

Table 1: Results of models on Test Part of Fashion MNIST

From the table above, we can see that CNN model has the highest accuracy from the chosen models.However we should also mention the “SVM with rbf” with relatively good results, even thought it is not a DNN model.

In case of KNN, we can analyze the confusion matrix with 3 different metrics(Fig 2-4). It can be said that in all three metrics KNN underperforms on predicting class4 (Coat) and class6 (Shirt).The same pattern of underperforming we can observe in CNN model.

In the Figure 5,we can see that our CNN model is underperforming for class 6(Shirt) in both precision and recall. In case of class 2(Pullover),precision parameter is low, while for class 4(Coat) recall parameter is low. We can assume that presence of similar patterns on multiple classes negatively effect the performance of KNN and CNN.

In the Figure 6, classification report of Logistic model shows that model underperforms on predicting class0, class2, class4, class6.So we can state almost all machine learning models have a poor performance on predicting Pullover, Coat, Shirt.

Overall our CNN model outperforms the following kaggle notebook: https://www.kaggle.com/pavansanagapati/a-simple-cnn-model-beginner-guide

The model of this notebook yields the following results:

- Test Loss : 0.2606
- Test Accuracy : 0.9089

Our CNN model Results:

- Test loss: 0.20388315618038177
- Test Accuracy: 0.9290000200271606

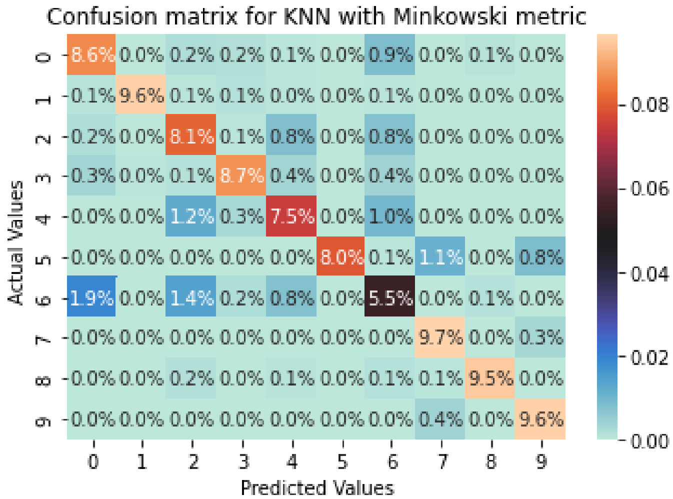

Fig. 2. Confusion Matrix for KNN with Minkowski metric

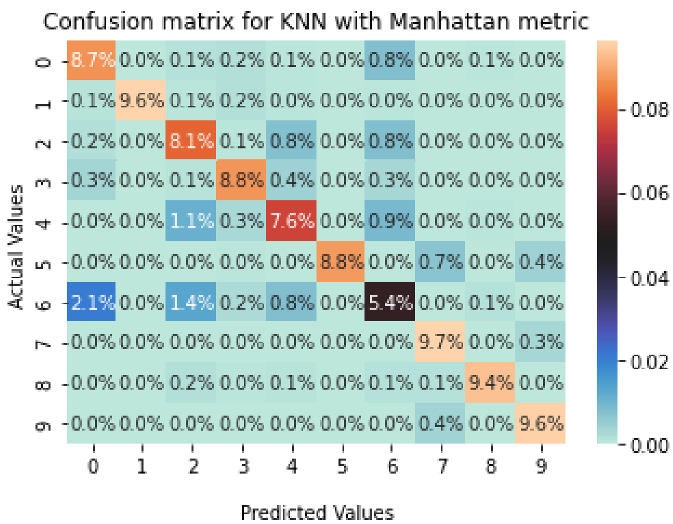

Fig. 3. Confusion Matrix for KNN with Manhattan metric

IV. Conclusion

Obtained results from 4 models evidence that CNN classification on clothes dataset is the most accurate comparing to 3 other models which were K nearest neighbors, SVM, and Logistic Regression. Limitations of each method are as follows:

Logistic Regression was tested on grayscale photos that were centered and rotated regularly, with a lot of blank space. It might not work with more complicated photos, though.

While K nearest neighbors produced good results, they still outperform CNNs because they don't function in the immediate vicinity of each individual feature, but centroids fail because they can't distinguish between similar-looking items (e.g. pullover versus t-shirt/top).

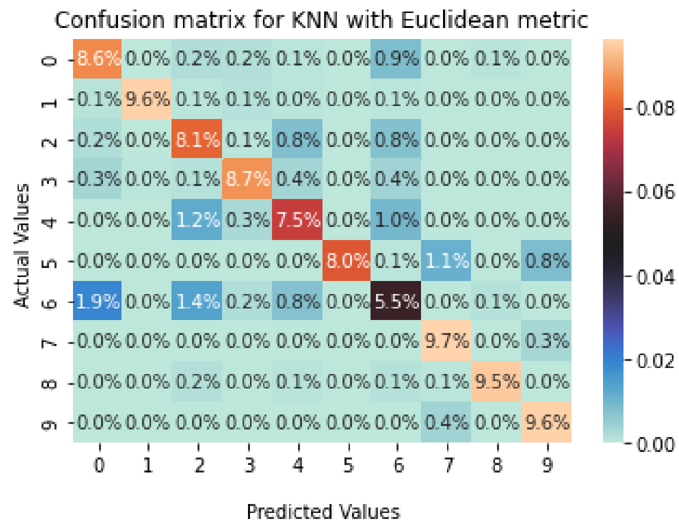

Fig. 4. Confusion Matrix for KNN with Euclidean metric

	precision	recall	f1-score	support
Class 0 (T-shirt/top) :	0.89	0.88	0.88	1000
Class 1 (Trouser) :	0.99	0.99	0.99	1000
Class 2 (Pullover) :	0.86	0.92	0.89	1000
Class 3 (Dress) :	0.92	0.96	0.94	1000
Class 4 (Coat) :	0.92	0.85	0.89	1000
Class 5 (Sandal) :	0.99	0.98	0.99	1000
Class 6 (Shirt) :	0.80	0.77	0.79	1000
Class 7 (Sneaker) :	0.96	0.97	0.97	1000
Class 8 (Bag) :	0.97	0.99	0.98	1000
Class 9 (Ankle Boot) :	0.97	0.97	0.97	1000
accuracy			0.93	10000
macro avg	0.93	0.93	0.93	10000
weighted avg	0.93	0.93	0.93	10000

Fig. 5. Classification Report for CNN

Despite the fact that image classification is not SVM's strong suit, it is extremely useful for other binary classification tasks. The most significant disadvantage is that it requires feature selection, which reduces accuracy, and without it, SVM can be computationally expensive.

Analyzing the incorrect predictions, it is possible to conclude that the diversity of similar patterns found across multiple classes has a negative impact on the CNN model's performance. Long sleeves, buttons, and a similar form characterize Class 2 (Pullover), Class 4 (Coat), and Class 6 (Shirt). This can be credited to the comparatively poor results (precision/recall) in predicting the above-mentioned classes.

As it was mentioned in the Results section, our CNN model performed better than the one from top of Kaggle's board.

And, while the other models do not produce as good results as CNN on this dataset, they are still used for other image processing tasks (sharpening, smoothing etc.).

About our future goals, we could compare obtained results with other models that were not covered in this project such as Random Forest, Fine Tuning DARTS, etc.

	precision	recall	f1-score	support
Class 0 (T-shirt/top) :	0.79	0.83	0.81	1000
Class 1 (Trouser) :	0.97	0.98	0.97	1000
Class 2 (Pullover) :	0.77	0.78	0.77	1000
Class 3 (Dress) :	0.87	0.89	0.88	1000
Class 4 (Coat) :	0.77	0.81	0.79	1000
Class 5 (Sandal) :	0.93	0.92	0.93	1000
Class 6 (Shirt) :	0.67	0.57	0.62	1000
Class 7 (Sneaker) :	0.91	0.90	0.91	1000
Class 8 (Bag) :	0.95	0.95	0.95	1000
Class 9 (Ankle Boot) :	0.93	0.95	0.94	1000
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

Fig. 6. Classification Report for Logistic Regression

REFERENCES

[1] Datt A. (2017). GitHub. https://github.com/anujdutt9/Fashion-MNIST/blob/master/knn.py. Accessed November 30, 2021.

[2] Fiori L. (2020). Medium. https://medium.com/@luigi.fiori.lf0303/distance-metrics-and-k-nearest-neighbor-knn-1b840969c0f4. Accessed December 2, 2021.

[3] Jianming, Z. H. A. N. G., Zhicai, Z. H. A. N., Keyang, C. H. E. N. G., Yongzhao, Z. (2015). Review on development of deep learning. Journal of Jiangsu university: natural science editions, 36(2), 191-200.

[4] Sanagapati P. (2020). Kaggle. https://www.kaggle.com/pavansanagapati/a-simple-cnn-model-beginner-guide. Accessed December 1, 2021.

[5] Techtangent Community. (2020). Medium. https://medium.com/@techtangent.in/image-classification-on-fashion-mnist-dataset-using-logistic-regression-c079fec56712. Accessed December 1, 2021.

[6] Zailando Research. (2017). Kaggle. https://www.kaggle.com/zalando-research/fashionmnist

CONTRIBUTION

All work is distributed evenly among the members of the group. We helped each other during work on this report.
Approximate work distribution:
Adilkhan Bakridenov: *CNN, Abstract and Results sections*
Moldir Berkaliyeva: *Logistic Regression, SVM, Conlusion and References sections*
Darya Taratynova: *KNN, SVM, Introduction and Methods sections*