EE769

Intro. To Machine Learning

Assignment Report

Shreyas Chaudhari
160100009

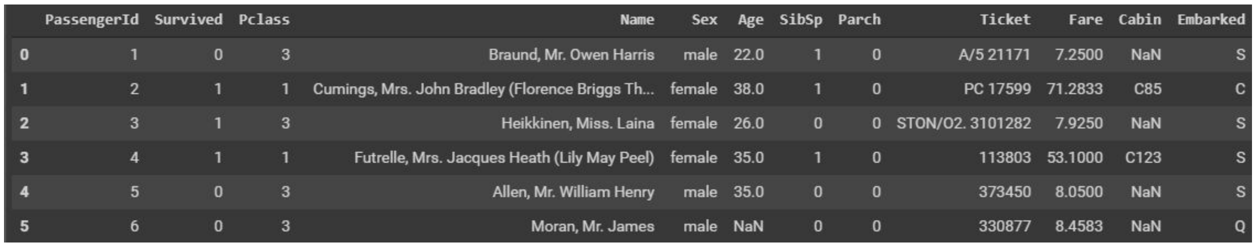
Tree Based models

Tree based models are used mostly for classification rather than regression. It uses comparators to predict the target value. From first node to leaf node data is compared and passed to the next children node accordingly. Last layer predicts the output of the data.

Multiple tree models are stacked parallelly to make more accurate predictions, This model is called random forest. In this model there are multiple outputs from every single tree, And final decision is made after voting or averaging. There are many variations of this model like Bagging ,Boosting, Bootstrap etc.

Here we are using Random Forest for classification of titanic dataset to predict if a person survived or not.

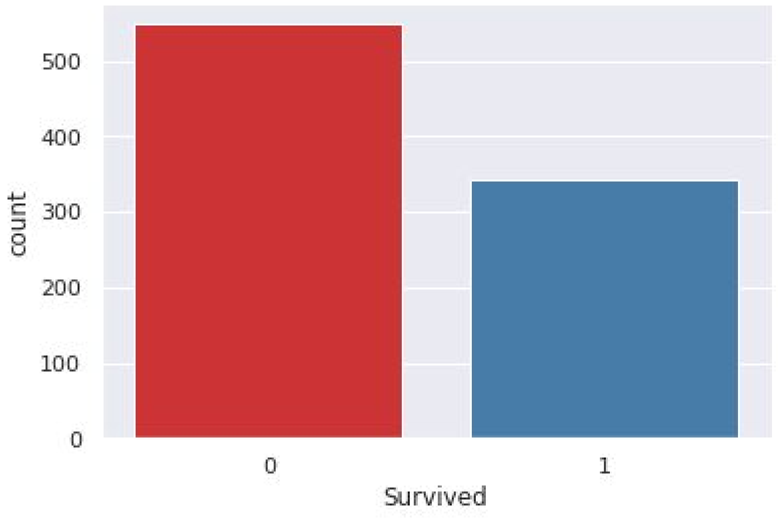
Data analysis and visualization:



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q

Fig1.1. Titanic dataset

As you can see that there are many redundant features of the data like PassengerId, Cabin, Name, Ticket. Though we can extract some common information from them also but for simplicity these features are not used. Object data is converted into one hot encoding.



Data is almost balanced (both classes are enough) so we don't need to worry about this. But there are some null values in the age data.

Fig1.2. distribution of the classes

For missing data we can use another model to predict the missing value. As shown in the

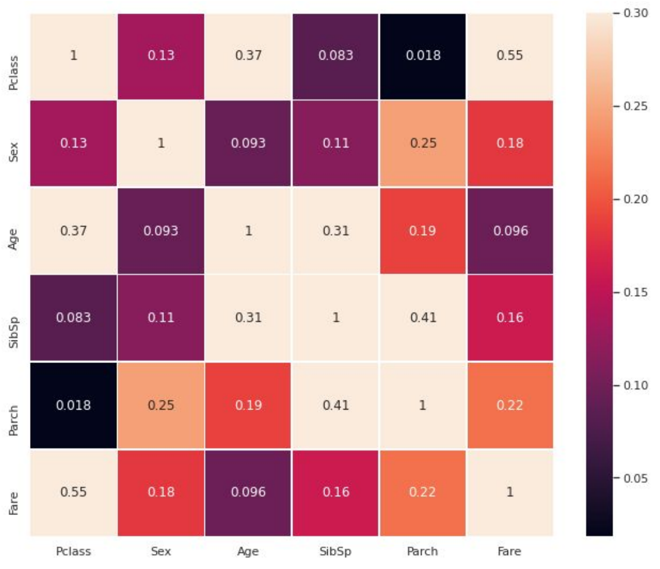


Figure1.3 Age is somewhat related to other features. Here the KNN regression model is used to predict the null age of the data. Also this creates some extra relation in the dataset or creates bias in the data, but the amount is small so we can ignore that. Another method to fill null data is mean value but this creates more bias.

Fig1.3. correlation matrix of the features

Training:

For Training, 50 estimators are used in Random forest with depth of 10 layers. Dataset of 891 samples is split for 20% testing data and trained on 80% data.

Results:

Confusion matrix	Predicted not survived	Predicted survived
Not Survived (0)	108	10
survived (1)	14	47

Precision based on survived class = 0.82

Recall based on survived class = 0.77

Roc curve: higher AUC value shows good performance of the model

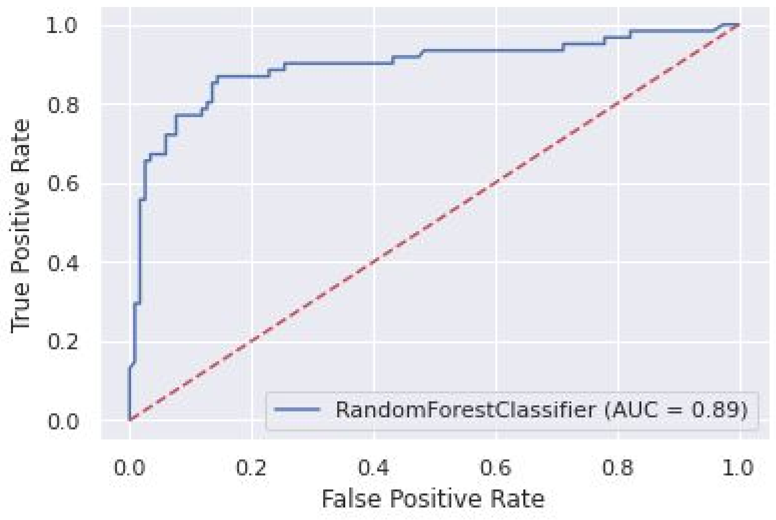


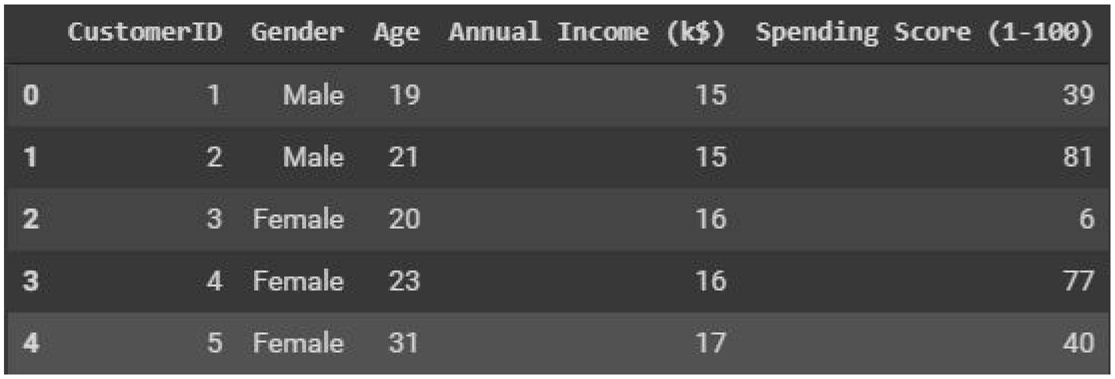
Fig1.4. ROC curve

Clustering

Clustering is a method used to find groups in the data, how data points are gathered in a space. This technique is useful to understand behaviour of certain groups and how they are different from each other. Here we have super market data of different customers, which is clustered using various clustering techniques (kmeans, agglomerative clustering, dbscan).

Data analysis and visualization:

Supermarket data has the following features. From which “CustomerID” is a redundant feature.”Gender” is a categorical feature which is converted to a binary feature.



	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Fig2.1. supermarket customers data

Range of features are vering too much, so data is standardized. This data has four features left. In this dendrogram the first blue line of the tree separates male and female, so we can use a separate clustering algorithm for male and female data separately.

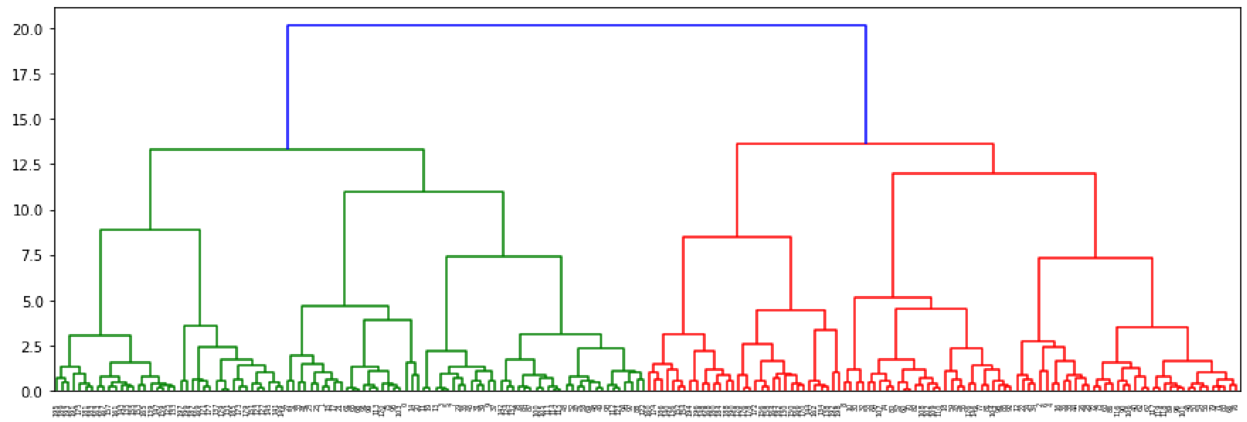
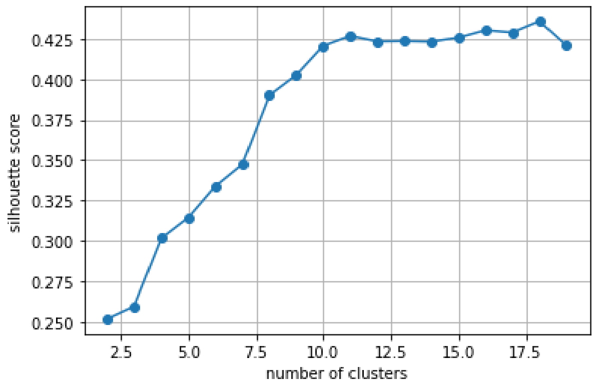


Fig2.2. dendrogram of supermarket data

Kmeans:

First collectively clusters are formed using a sklearn library using all four features. Number of clusters are chosen based on ‘silhouette score’. Silhouette score is a measure of similarity of own cluster and separation to other clusters. Higher scores means well separated clusters are formed.



From the graph we can see that after 11 clusters the graph is almost flat. So 11 clusters are chosen. From the dendrogram figure2.2 we can also see that 10-11 clusters are visible clearly.

Fig2.3. silhouette score with different clusters

After 11 clusters data is plotted in 3D, where we can also see that gender data is clearly separable.

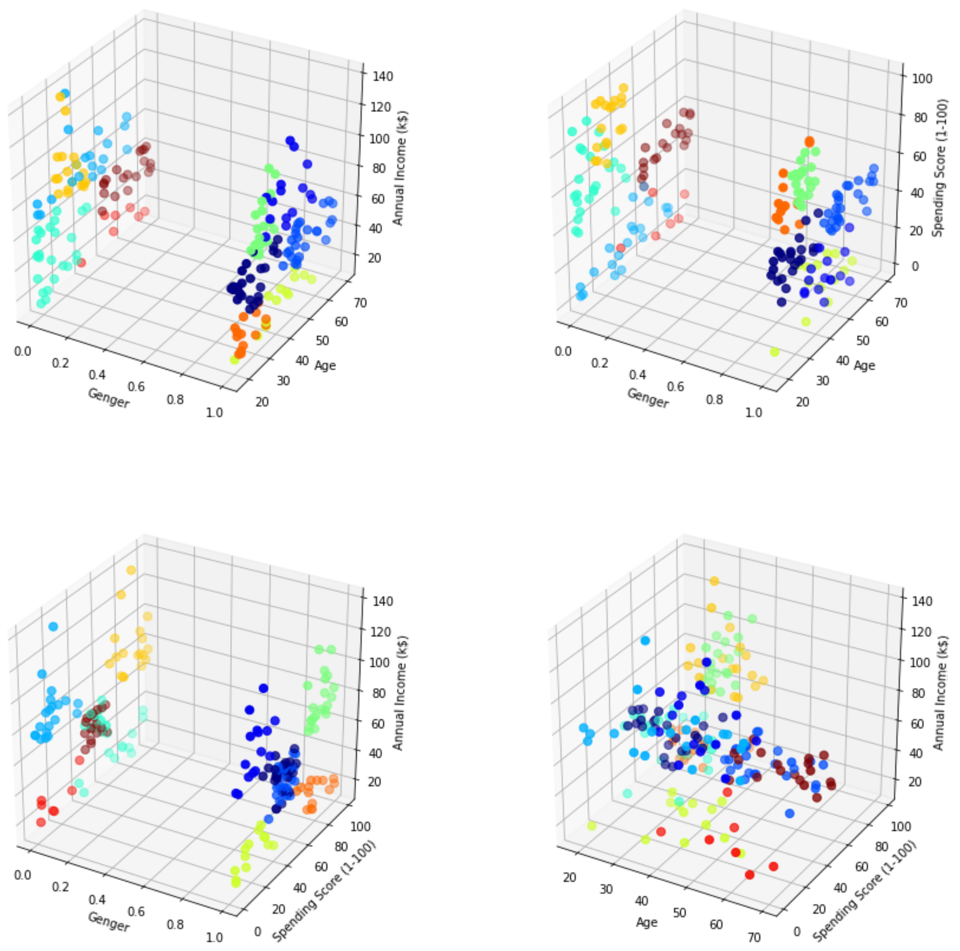


Fig2.4. kmeans clustering using 11 clusters

To visualize data clearly we plot the male and female data separately.

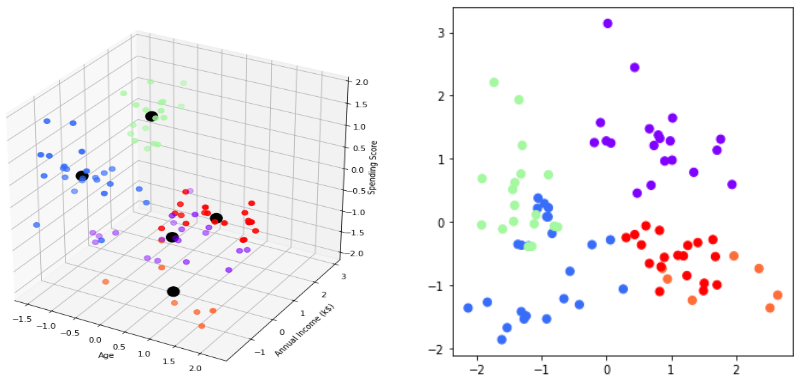


Fig2.5 (a) male data with 5 clusters. (b) data represented in 2D plane using PCA transformation

As we can see in male dataset, blue and green clusters are the customers whose spending rate is higher and age is lower.There is low income and older people with low spending shown in orange cluster. Red and purple clusters are middle age and average rating customers.

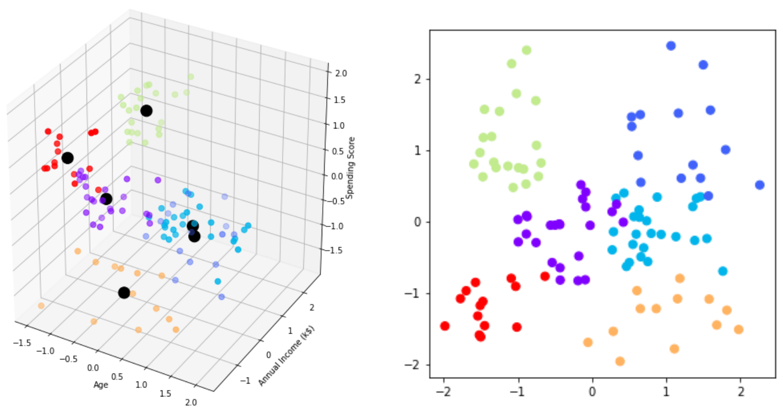


Fig2.6 (a) female data with 6 clusters. (b) data represented in 2D plane using PCA transformation

In this female data some of the clusters show similar trends as male clusters, like green and red clusters here are similar to green and blue clusters in male datasets. Here orange and blue clusters are middle aged female and give lower ratings. Average income and average scores clusters are purple and light blue.

As shown in PCA clusters some of them are overlapping, because in higher dimensions distances are different than reduced projected dimensions.

DBSCAN:

Dbscan is another method for clustering, and its clusters are not radial uniform clusters. Here again male and female datasets are clustered separately. Parameters are chosen such a way that unlabeled data is minimum. Here red points are unlabeled data. Both datasets have four clusters as shown in fig2.7. By increasing min_samples there can be too many clusters and more data is unlabeled also. Similarly a large bounding radius leads to all datasets to a single cluster. In male datasets clusters are well separated while in female datasets there is a large green region which may be the union of two small clusters.

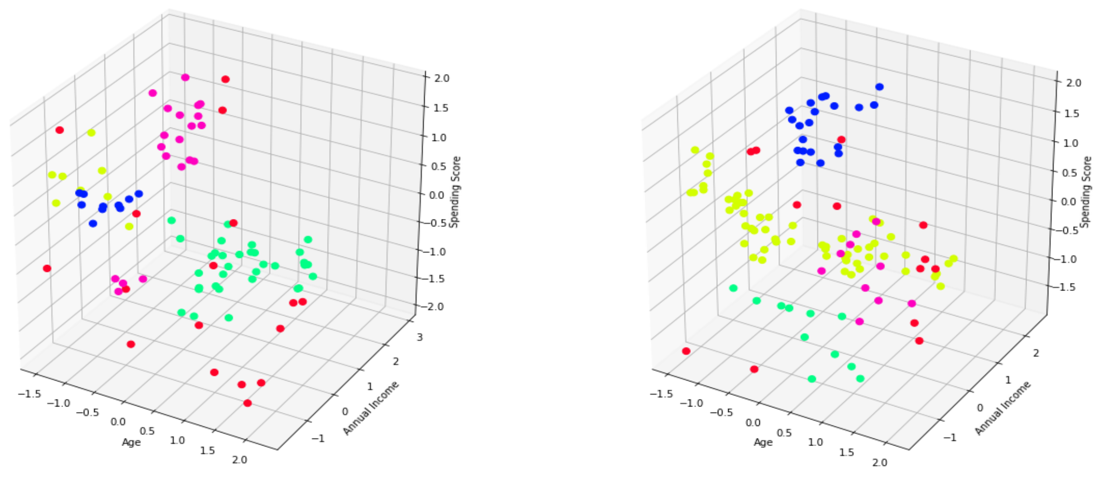


Fig2.7 (a)Dbscan on male dataset (b)Dbscan of female datasets

Result and Conclusion:

- In this datasets kmeans and agglomerative clustering is a good clustering algorithm and Dbscan is not useful for this datasets.
- We can first classify these datasets by male and female, then there are low income and high spending rate customers in both classes. Another cluster is old customers with low spending rates. There is high income, high spending rate and young people cluster. Also there are customers with high income and middle aged males have lower spending rate. Similar in middle income and middle aged females with low spending rates.
- Here we can give some incentives to different groups of people to improve marketing. We can put some conditional scheme for average rating customers to increase their spending rate.For high income young customers who already have higher spending rate, we can provide more variety of goods and new products or advertising may help.
- Clustering also helps with how much portion of the data is in a particular cluster,which is useful to decide the importance of that group. Working on a large cluster will have more impact.

CNN Classification

CNN is mostly used for computer vision problems. CNN captures patterns in the image irrespective of the location in the image, thus it requires less amount of parameters than dense neural network. There are many variations of this algorithm in many applications like Object Detection, Segmentation, Neural Style transfer, Face Recognition etc.

Here we have MNIST data for classification. Digit dataset and fashion dataset are classified using LeNet5 architecture. For fashion dataset image augmentation was performed to increase accuracy.

Data Processing:

MNIST dataset is grayscale images of handwritten digits from 0 to 9. All images have 28x28 pixels and are nicely distributed over each class. Now all pixel values are mapped to [0,1] range so weights remain in similar range. Output classes are one hot encoded. Now this data is ready for training

Fashion MNIST dataset is also grayscale with the same size of image and same 10 classes. But it contains fashion items as shown in figure3.1, which has a lot more complex features than handwritten digits. This dataset is also scaled in range [0,1] and output is one hot encoded.

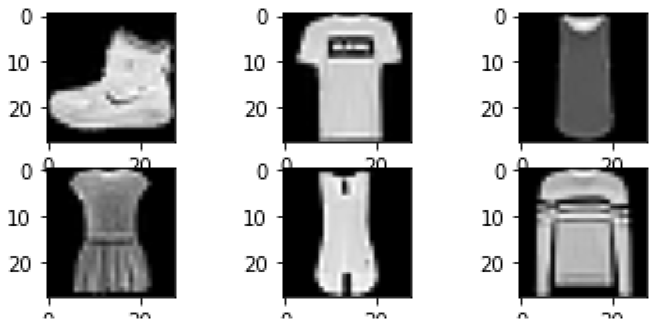


Fig3.1. sample of Fashion MNIST dataset

Data augmentation is used to generate dataset using fewer dataset. When there is an overfitting problem this technique is very useful. It reduces time for manual labeling. There are many operations for data augmentation, some of them used for training are as follows.

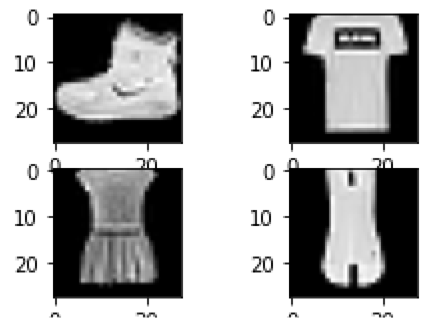


Fig3.2. Random crop and resize image

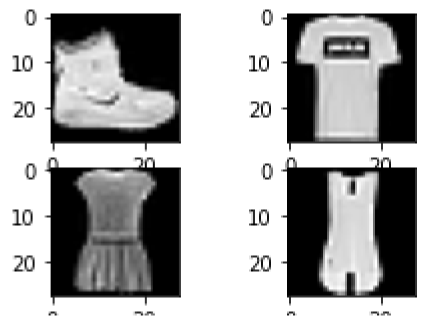


Fig3.3. Flipping image horizontally

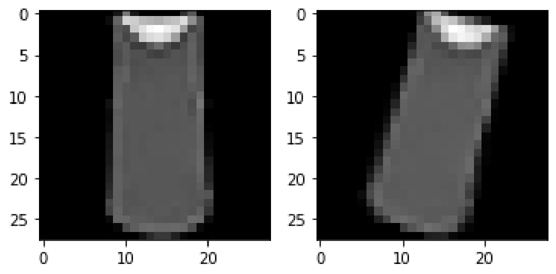


Fig3.4. Random angle rotation

We can manipulate the images manually before training and also there is a library available in keras which augments the data during training.

Model training:

LeNet5 architecture is designed using keras layers for CNN. This model has five convolution layers and two dense layers. Total number of parameters are 557,328 and all are trainable. We are using categorical cross-entropy loss, which is commonly used for classification problems. Adam optimizer is used for model training.

Results:

First model is trained on the MNIST digit dataset. Which has very high accuracy.Training and validation graphs are almost similar.

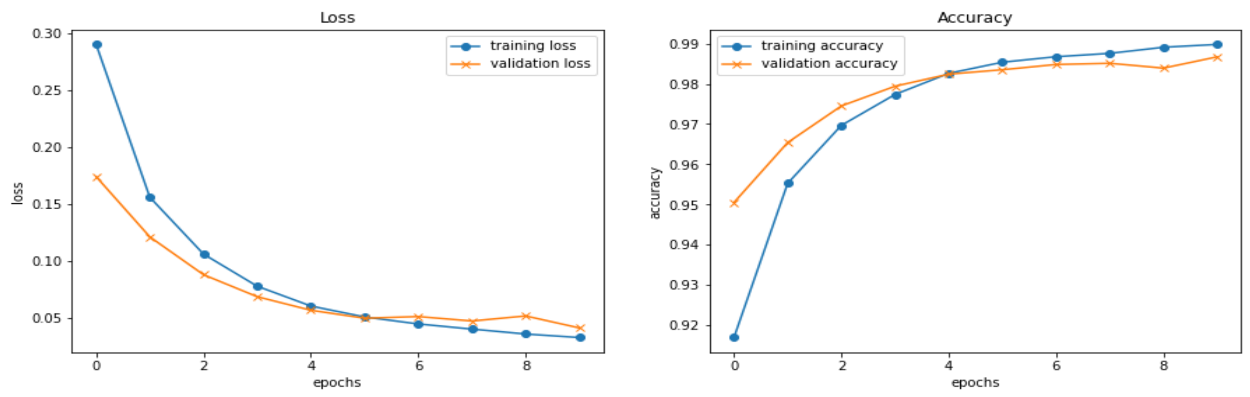


Fig3.5.MNIST digit model (a)loss (b)accuracy

Now the model is trained on the Fashion MNIST dataset. Which shows lesser accuracy and training and validation graphs are different after 10 epochs.

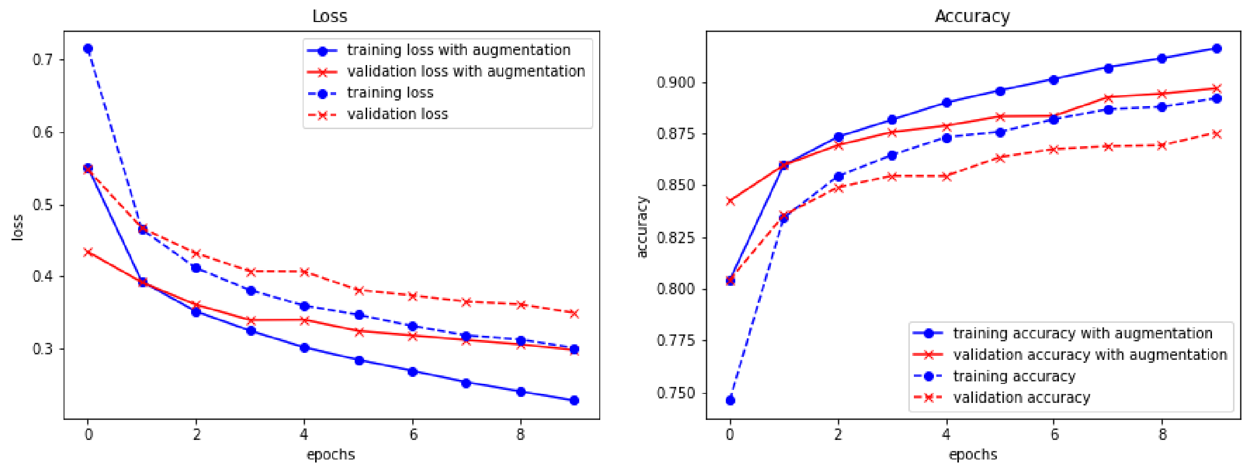


Fig3.6.Fashion MNIST model (a)loss (b)accuracy

As we can observe accuracy is increased after augmentation accuracy higher than original dataset. But still training and validation graphs are not on similar lines. Sometimes augmented data can be different from validation sets which may cause poor performance.

Image Segmentation

Image segmentation is used to classify each pixel of the image. So this model can recognize different object classes in the image. This algorithm uses CNN for pattern extraction which is followed by upsampling to map these patterns on image. This process is known as encoding and then decoding. It has many applications in medical-images, autonomous vehicles etc.

Here we have a CamVid dataset of street images which has 12 classes like car,road,people etc. two models were used for segmentation.

Dataset visualization and processing:

Dataset contains images of size (480x360) and labels of 12 classes. Different classes are shown in different colours. Black colour is for unlabeled classes. Labels have the same dimensions as images with one channel representing the class of each pixel.

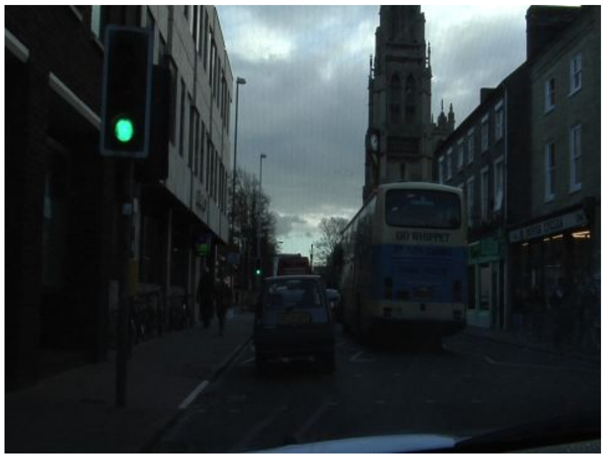


Fig4.1. image

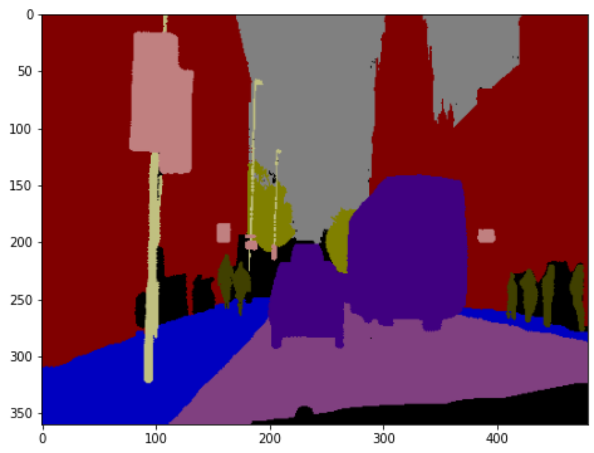


Fig4.2. image labels

Dataset has 367 training examples and 101 validation examples. Image size is reduced to avoid GPU errors (may be overflow) and increase computation speed. For model images of size 224x160 were used. Image data is normalized and some of the dataset are horizontally flipped also.

During encoding part image dimensions are reduced by a factor of two and there may be odd length dimensions in between also. Problem arises during upsampling , because dimensions are doubled every time which are always even. And these layers are not able to concatenate with odd layers from encoding.

UNet Model training:

Unet model is copied from the blog link provided in the assignment. This model has four blocks of encoding and four blocks of decoding and concatenation. 'He_initilizer' is used for weight initialization. This model has 31,032,870 trainable weight parameters. This model gives output of 12 classes for each pixel and after softmax activation we get the predicted class for that pixel.

For training, a minibatch of 8 samples and "sparse categorical cross entropy loss" is used. This loss is the same as cross entropy loss and calculated pixelwise on the entire minibatch.

This model gives very less pixel wise acuray around 41%. Most of the time the model predicts only one 'road' class in the entire image, this is because weights are stuck in local minima and also the amount of road pixels are very high around 28% in the dataset. This loss function can't handle bias in classes.

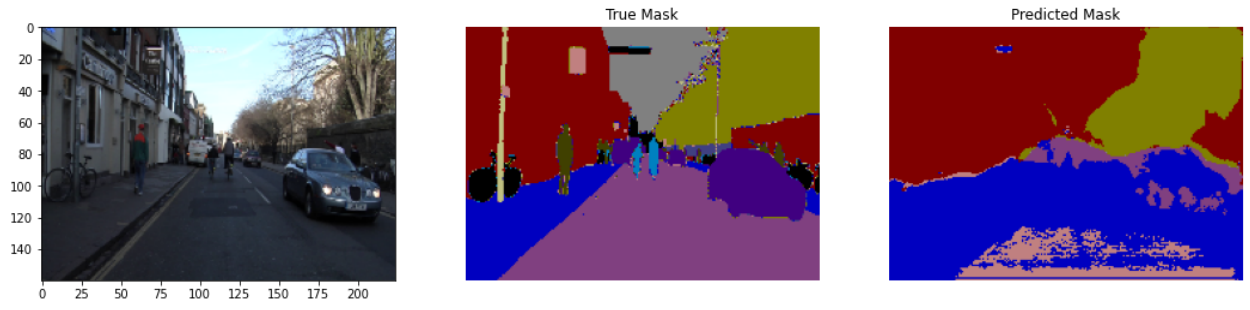


Fig4.3. Unet model with 10 epochs

UNet Model training with dice loss:

Same Unet model is trained using dice loss and cross entropy loss. Dice loss uses the 'IoU' analogy as a loss function. 'IoU' is the amount of overlap between two classes. High overlap gives low dice loss value.

Weighted loss is used for training which gives better results. Without this minor classes have less losses which remains untrained in the model. Sometimes weighted loss creates problems if minor class loss is amplified much, it creates too many fluctuations in the model. There are many ways we can use a combination of these losses with weighted and unweighted losses.

In modified Unet model weighted dice loss and softly weighted cross entropy loss is used with 1:0.8 ratio. Which gives accuracy of 62% and this model rarely saturates at local minima.

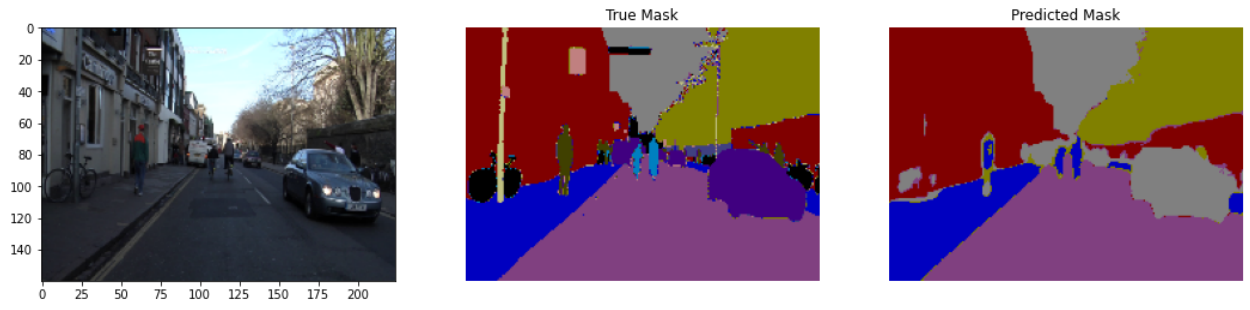


Fig4.4. UNet model with modified loss function with 50 epochs

Transfer Learning:

Transfer learning is a widely used and effective algorithm in many applications of computer vision, NLP etc. There are many trained open source models available on the internet, and their hyper parameters are tuned very well. We have to just train the last few layers according to our application, because the front layers extract the details very efficiently.

MobileNetV2 is used as an encoding block of the model. Five layers are extracted from this model which are concatenated to decoder parts. This model is slightly different from the UNet which we implemented. Decoder part uses pix2pix layers which is shown below.

Pix2pix : (upsampling=>conv2D=>batchNorm=>instanceNorm=>LeakyRelu=>conv2D)

Total parameters are 6,517,196 from which we freeze MobileNetV2 parameters and trainable parameters are 4,673,292 These are from the decoding part of the model. Here again we use only categorical cross entropy loss.

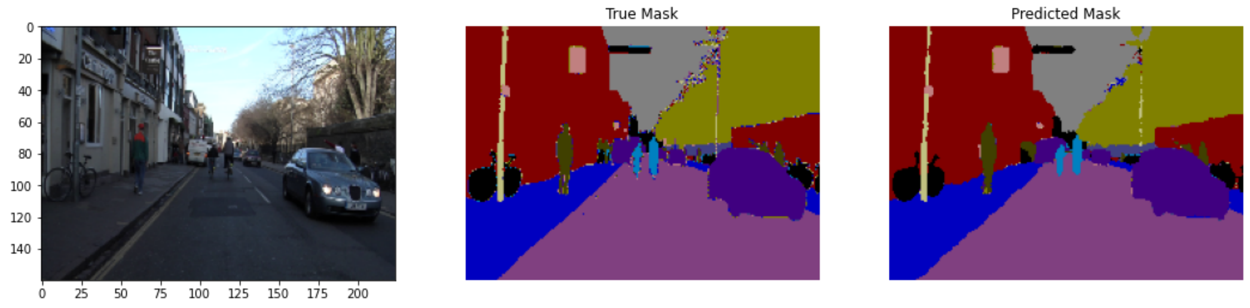


Fig4.5. Transfer learning UNet model with 50 epochs

This model gives very good accuracy around 85% with very less parameters. As shown in the figure4.5, the true mask and predicted mask are very similar with very fine details.

Conclusion:

- The UNet model performs very poorly on the dataset. And also this is prone to local minima. It is able to classify very few classes with unclear boundaries.
- After modifying the loss function UNet is able to achieve higher accuracy. This model can classify more classes but still detailing of the image is not achieved.
- Transfer learning gives very satisfactory performance on the dataset. It is giving better details with less number of parameters. Batch normalization improve performance significantly.