STAT 542 / CS 598

Project-2

Team

Jingwei Li - jl96

Pushpit Saxena - pushpit2 **(Team Lead)**

Harish Chandramohan - harishc4

Project Description

The goal of this project is to build classification models for identifying malignant moles based on skin lesion images. The dataset contains 300 images of skin moles, of which, 150 images are of benign and 150 images are of malignant skin moles.

The first part of the project is to build 3 different classification models that are trained using the image pixel data. The image files are of inconsistent sizes, so the images were resized to a consistent format before using it to develop the models. The processing done on the images will be detailed in the later sections. Data from all 3 channels (RBG color) from the resized image will be used to develop the models.

The second part involves feature engineering to extract new features from the images, so that the features are more interpretable to the user. Research papers and existing literature in the field of skin cancer detection were reviewed to learn about clinically relevant features that are useful in detection of malignant moles. The new features will be used to train 2 new models, so that the models are more accurate, interpretable and explainable to the user (medical doctor).

Results:

Model	Accuracy
Question 1: Pixel based SVM	0.7
Question 1: Pixel based RandomForest	0.733
Question 1: Pixel based LDA	0.683
Question 2: Image based features SVM	0.867
Question 2: Image based features XGBoost	0.767

STAT 542 / CS 598

Project-2

Data Processing

Image resizing (*image_processor.py for code***)**

The images are of inconsistent size and hence needs to be resized into a consistent format to be useful in building the model. On inspection of the images, all the images seemed to maintain a 4:3 aspect ratio with 3 channels of colors – Red, Green and Blue. The images were resized, using area interpolation method, to 600 wide x 450 high x 3 channel format to maintain the original aspect ratio and the smallest image in the dataset is 600 x 450.

PrincipleComponent (*Project2_question1 notebook for code***)**

As the images we have after resizing on flattening generated vectors with 810000 features which leads to higher model training time as well as there is collinearity also in this feature space. So, we also performed PCA to reduce the feature space. We used 0.998 as n_components to select number of components such that amount of variance that needs to be explained is greater than this percentage. This gives us 300 vectors (150 benign, 150 malignant) with 263 components.

```
X = np.concatenate((X_benign, X_malinant), axis=0)
Y = np.concatenate((np.zeros(X_benign.shape[0]), np.ones(X_malinant.shape[0])))
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)
pca = PCA(.998)
pca.fit(X_scaled)
X_pca = pca.transform(X_scaled)
```

STAT 542 / CS 598

Project-2

Classification models based on pixels (*Project2_question1 notebook for code)*:
We have taken the processed data (as described previously) and split it into Train Set (80%) and HoldOut Set (20%). Then we build different classification models using 10-fold cross-validation, to get a base-line idea of how each model is performing in general on the given dataset.

As can be seen from the box plot above the models which are generally performing better (with default hyperparameters) are SVM, LDA and RandomForest (we have also taken into consideration that our models are sufficiently different, hence we didn't choose XGBoost for this question).
We explored these models further and performed hyperparameter tuning to generate the best model for each of those model types. Please note that all these models are built on the processed data (i.e. image resized to 600x450 and PCA is applied, see data processing section).

Classification Model 1: SVM

Hyperparameter tuning:

We leveraged the GridSearchCV in sklearn library to perform the hyperparameter tuning.
Following parameters are tuned:

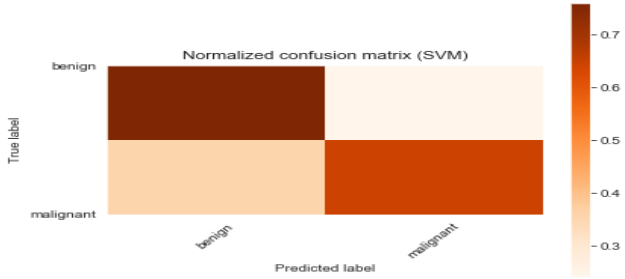
- **C** → [0.1, 1, 10, 100, 1000]
- **Gamma** → [1, 0.1, 0.01, 0.001, 0.0001]
- **Kernel** → ['rbf', 'linear', 'poly']

```
1. grid = GridSearchCV(svm.SVC(), param_grid, refit = True, verbose = 3, cv=KFold(n_splits
   =10), n_jobs=5, scoring="accuracy")
2. grid.fit(X_train, Y_train)
```

Best parameters: `{'C': 0.1, 'gamma': 1, 'kernel': 'poly'}`
Hold-out set accuracy score (for the best estimator): 0.7
Precision/Recall metrics for SVM(best estimator):

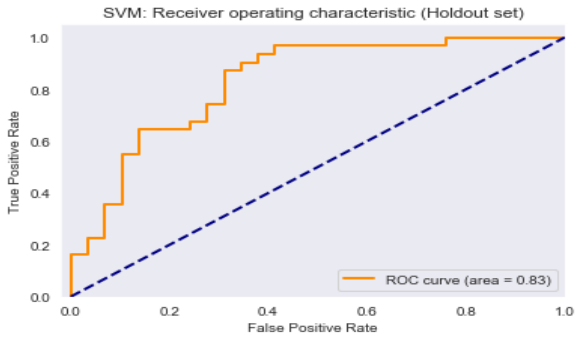
Class	Precision	Recall	F1-Score	Support
benign	0.67	0.76	0.71	29
malignant	0.74	0.65	0.69	31
accuracy			0.7	60
macro avg	0.7	0.7	0.7	60
weighted avg	0.7	0.7	0.7	60

Confusion matrix:



ROC Curve for SVM (best estimator):

```
plot_roc_auc(Y_test,grid.decision_function(X_test), 'SVM: Receiver operating characteristic (Holdout set)')
```



Classification Model 2: RandomForest

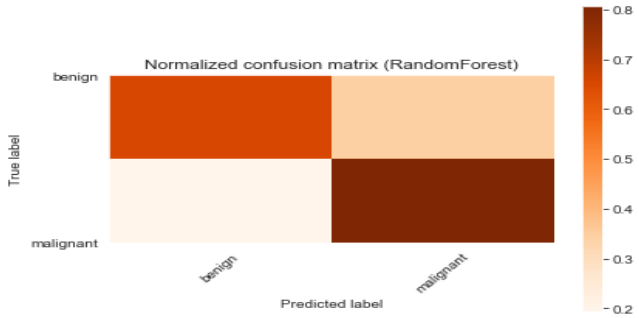
Hyperparameter tuning: Again, we used GridSearchCV to tune the hyperparameters.

```
params= {
    'n_estimators': list(range(10,500,10))
     'max_features': list(range(6,32,5))
}
rf_grid = GridSearchCV(RandomForestClassifier(random_state=seed), params, refit=True, cv=KFold(n_splits=10), verbose=3,scoring="accuracy", n_jobs=5)
rf_grid.fit(X_train, Y_train)
```

Best Parameters: `{'max_features': 16, 'n_estimators': 420}`
Hold-out set accuracy score (for the best estimator): 0.733
Precision/Recall metrics for RandomForest(best estimator):

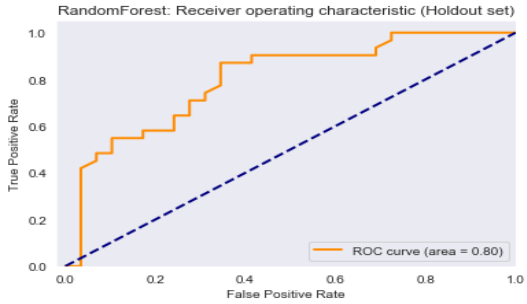
Class/Metric	Precision	Recall	F1-Score	support
benign	0.76	0.66	0.7	29
malignant	0.71	0.81	0.76	31
accuracy			0.73	60
macro avg	0.74	0.73	0.73	60
weighted avg	0.74	0.73	0.73	60

Confusion matrix:



ROC Curve for RandomForest (best estimator):

```
plot_roc_auc(Y_test,rf_grid.predict_proba(X_test)[:,1], 'RandomForest: Receiver operating characteristic (Holdout set)')
```



Classification Model 3: LinearDiscriminantAnalysis
Hyperparameter Tuning:

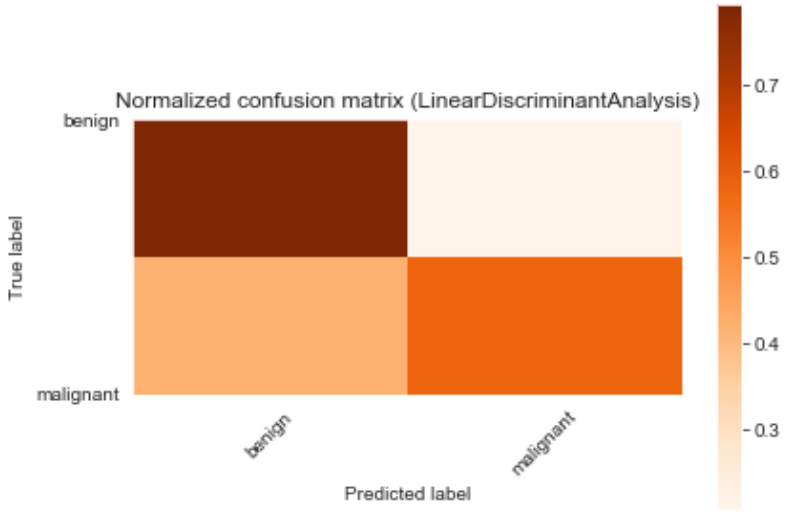
```
1. params = {
2.     'shrinkage' : np.arange(0, 1.1, 0.1)
3. }
4. lda_grid =  GridSearchCV(LinearDiscriminantAnalysis(solver='lsqr'), params, refit=True,
    cv=KFold(n_splits=10), verbose=3,scoring="accuracy", n_jobs=5)
5. lda_grid.fit(X_train, Y_train)
```

Best Parameters: `{'shrinkage': 0.9}`

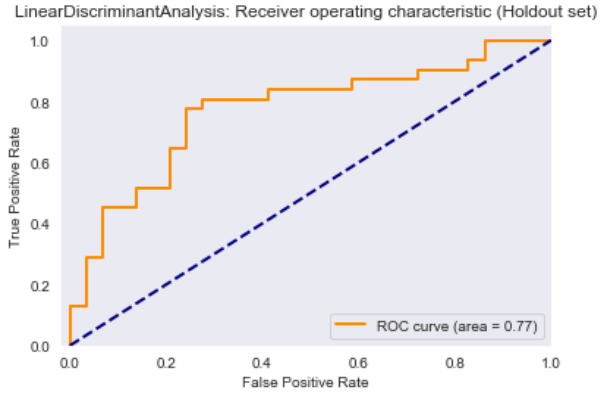
Hold-out set accuracy score (for the best estimator): 0.6833

Class/Metrics	Precision	Recall	F1-Score	suppor
benign	0.64	0.79	0.71	29
malignant	0.75	0.58	0.65	31
accuracy			0.68	60
macro avg	0.69	0.69	0.68	60
weighted avg	0.7	0.68	0.68	60

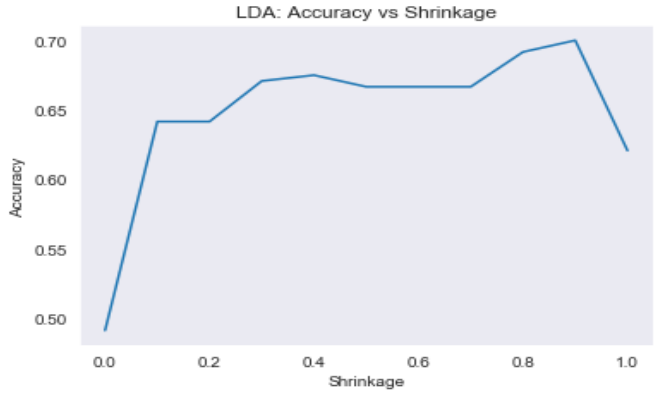
Confusion matrix:



ROC Curve for LinearDiscriminantAnalysis (best estimator):



LDA (Hyperparameter tuning Shrinkage vs Accuracy, *easier to include in the report as only one parameter is tuned, similar plots can be generated from other model's grid search object also, but we haven't included them here due to space constraint*)**:**



STAT 542 / CS 598

Project-2

Literature Review

Skin cancer is one of the most prevalent type of cancer worldwide. In the U.S, skin cancer is considered the most common type of cancer. The number of skin cancer cases has been growing over the past few decades, attributed to the higher depletion rate of Ozone layer and increase in the exposure to UV rays. Numerous studies and research have been conducted in the field of early detection of skin cancer. With the recent advancement in Machine Learning and AI, lot of research work has been conducted to use image processing and AI to detect cancer using images of skin lesions and moles.

We reviewed some of these publications to gain an understanding of the clinically relevant characteristics of the skin lesion that can be learned from the images and understanding the tools and techniques that can be used to extract these features from the images. Based on the understanding from reading few of the relevant publication the main characteristic that are useful for detecting malignant moles can be summarized based on the ABCD rule of dermatoscopy. The rule specifies the visual features associated
with malignant lesions symptoms. The ABCD acronym stands for **Asymmetry**, **Border structure**, **Color variation** and **Diameter** of lesion. These features define the basis for diagnosis of the disease and is commonly used by dermatologists.

- **Asymmetry (A)** - About half the time, a melanoma develops in an existing mole; in other cases, it arises as a new lesion that can resemble an ordinary mole. A noncancerous mole, however, is generally symmetric and circular in shape, while melanoma usually grows in an irregular, asymmetric manner.

- **Border Irregularity (B)** - Benign lesions generally have clearly defined borders. A malignant lesion, in contrast, often shows notched or indistinct borders that may signal ongoing growth and spreading of the cancer.

- **Color Variation (C)** - One of the earliest signs of cancer may be the appearance of various colors within the lesion. Because melanomas arise within pigment-forming cells, they are often varicolored lesions of tan, dark brown, or black, reflecting the production of melanin pigment at different depths within the skin.

- **Diameter (D)** - Malignant moles and lesions tend to grow larger than common moles and show typically at least a diameter of about 6mm.

References

1. *American Academy of Dermatology Association (https://www.aad.org/media/stats-skin-cancer)*
2. *Automatic Classification of Specific Melanocytic Lesions Using Artificial Intelligence (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4739011/#B28)*
3. *Feature Extraction for Skin Cancer Lesion Detection (http://ijsetr.org/wp-content/uploads/2015/05/IJSETR-VOL-4-ISSUE-5-1645-1650.pdf)*
4. *A PRELIMARY APPROACH FOR THE AUTOMATED RECOGNITION OF MALIGNANT MELANOMA (HTTPS://IAS-ISS.ORG/OJS/IAS/ARTICLE/VIEW/759/662)*
5. *Skin Cancer Diagnostics with an All-Inclusive Smartphone Application (https://www.mdpi.com/2073-8994/11/6/790/htm)*

STAT 542 / CS 598

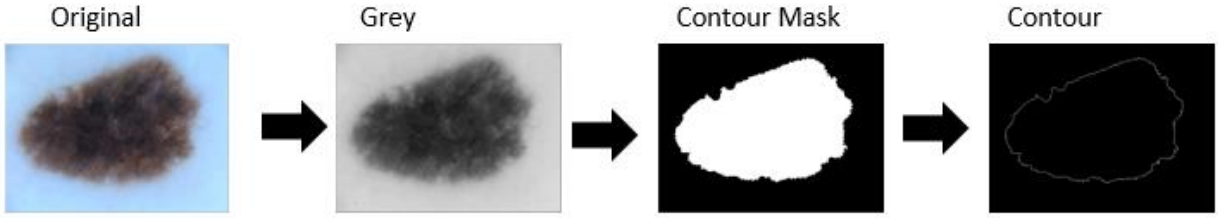
Project-2

Feature Engineering (*extract_features.py for code*)

Based on the understanding from the readings of existing literature, we decided to extract and use some key geometric features of the lesion as features based on the ABCD rule discussed in the previous section. The feature extraction was performed in 2 steps which involved image segmentation followed by feature extraction.

Image segmentation

The images were segmented using OTUS's method. The image was first converted to grey scale and OTSU's method uses grey level thresholding to extract the lesions from the background skin. Then lesion counter was generated using the segmented image mask. The contour was used extract the below mentioned geometric features of the image. Shown below is a sample of the intermediate images in the extraction process –



Feature Extraction

Asymmetry

1. Horizontal asymmetry and Vertical asymmetry: Calculated by overlapping the binary form of the warped segmented image with the mirror images in horizontal and vertical directions. The sum of all the non-zero pixels in the image is computed along the principal horizontal and vertical axis and divided by the area of the contour to get the asymmetry value. ***AS=Non zero pixels/Area***

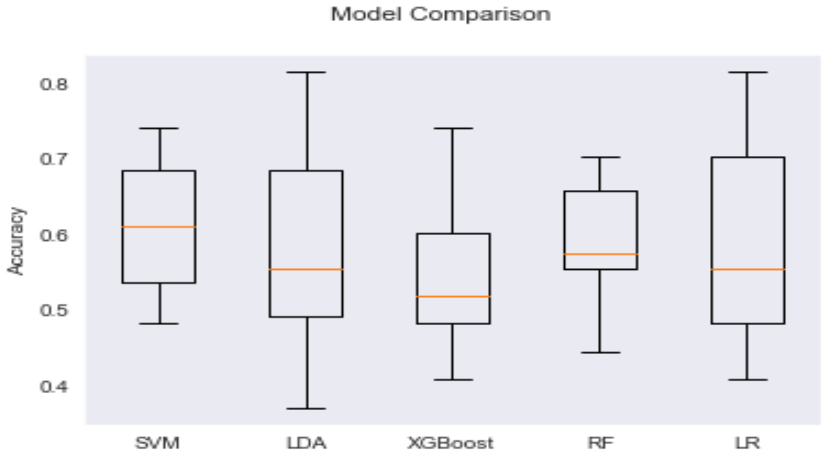
Border Irregularity

2. Border Irregularity Index: The 'Border irregularity' feature is generally defined as the level of deviation from a perfect circle and measured by the irregularity index using the below formula – **BI_index = *Perimeter of contour*^2 /(4*π**Area of contour)***

Diameter based features

3. Horizontal diameter: This is the horizontal diameter along the horizonal principal axis of the image contour
4. Vertical diameter: This is the vertical diameter along the vertical principal axis of the image contour
5. Area of lesion: This is the area of the lesion calculated by counting the non-zero pixels in the contour mask shown above
6. Perimeter of lesion: This is arc length of the image contour

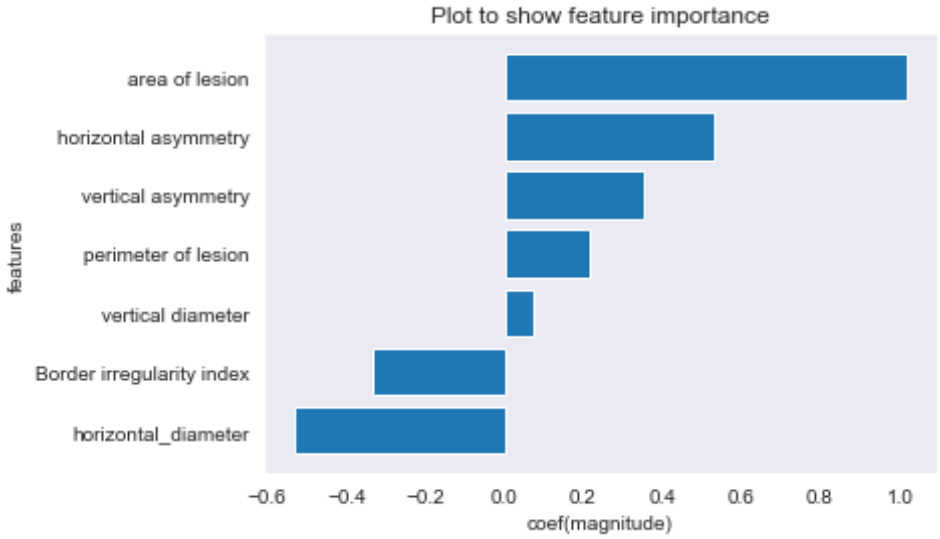
Classification model based on new Features (*Project2_question2 notebook for code***)**
We have generated the feature vectors for all 150 benign and 150 malignant images as per the feature extraction explained earlier. Following is the 10-fold cross-validation (similar to question1) to get a baseline idea of different model performances.



We have picked SVM (as it is best performing one) and XGBoost (decent performance, but provide ways to evaluate the feature importance). We also have LDA (in the notebook, which was also performing well).

Classification Model 1: SVM

Feature importance:
In order to analyze the feature importance we have trained 'linear' kernel SVM (as we can analyze the coefficient because they will be in the same feature space) and tuned hyperparameters using GridSearchCV.

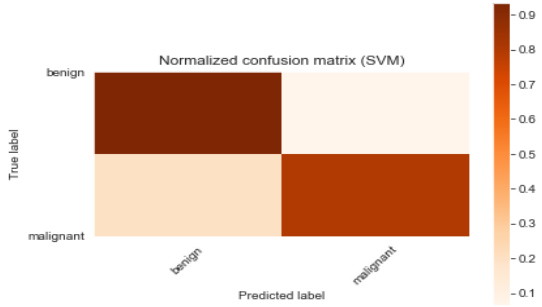


We also did the GridSearch with 'rbf' kernel and found that it is performing better on the dataset

SVM	Accuracy (Hold out set)
Linear Kernel	0.8
RBF	0.867

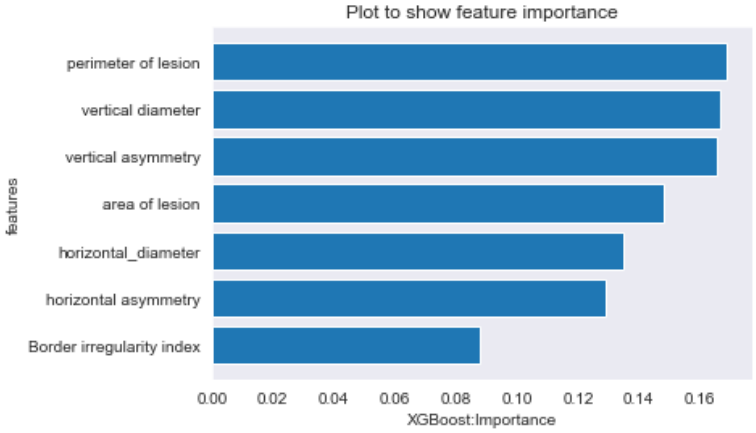
Best parameters: `{'C': 1000, 'gamma': 0.01, 'kernel': 'rbf'}`

Confusion matrix:



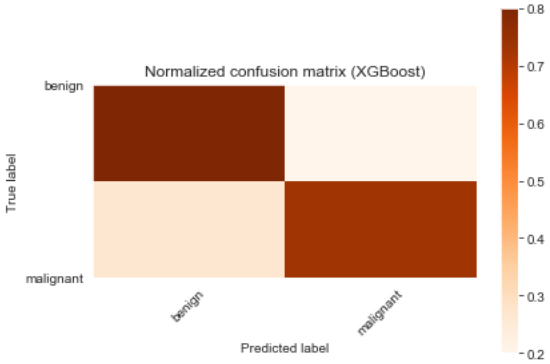
Classification Model 2: XGBoost

XGboost library provides methods to get the feature importance. As can be seen from the XGBoost feature importance data, it seems the least important feature is 'Border irregularity index', which can also be seen having low coef in linear SVM above, as well as low coefficient in LDA model also (check the code notebook)



Best parameters: `{'gamma': 1, 'learning_rate': 0.1, 'max_depth': 5, 'min_child_weight': 5}`
Accuracy on hold-out set: 0.767

Confusion Matrix:



So, to conclude we are getting better performance on this reduced feature set as these features capture the contextual information about the images, which is leading to better generalization.