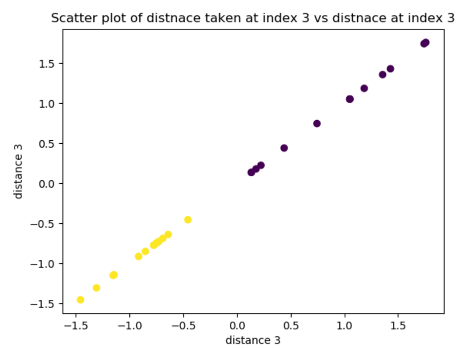
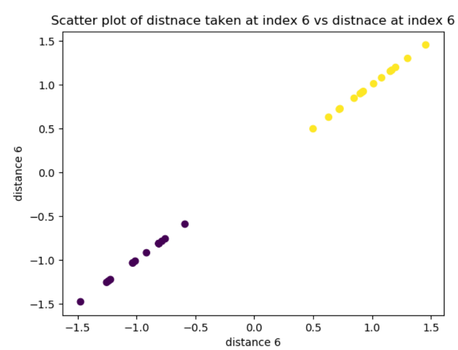
# Vehicle Classification from Boundary Images

**Meriam Elabor,** *University of Witwatersrand, COMS4030A*

March 12, 2018

## Question 1




- The features extracted are the distances from the top of the image to the top of the vehicle at different points
- By observation, these features distinguish the sedan class from the pick-up-truck class, seeing as the most distinguishing difference between them is the shape at the top
- To obtain these values, the image is read in as a matrix. 7 column indices that are evenly distributed are then calculated from the matrix.
- The distance at each index is then obtained by finding the row value of the 1st black pixel in that column. This value is then multiplied by $\frac{50.0}{image.height}$ to get the features on the same scale
- After these features are normalized, scatter (commented out on line 184 of $veh_{c}lass.py$) plots were drawn of the features against each other to visualize the degree to which the features separated the classes. The following two graphs presented the predominant features:

- as a result, only the distances taken at index 3 and 6 were used. The dimensions of the data is thus 2 (3 with the basis)

## Question 2

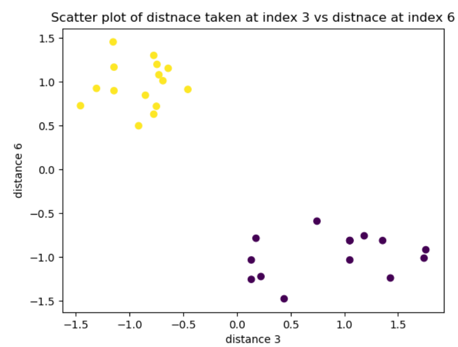
- Using the concept of hold-out cross-validation, the data was split randomly into training set (20 points) and validation set (10 points)
- different values of $K$ were then used to classify the points in the validation set, suing the training set, $K$ starting at 1, and incrementing by 1 each time, thus choosing the smallest value of $K$ needed.
- using the $KNN\_error$, if the error stopped decreasing, the previous value of $K$ was chosen since the error would only increase after this point, implying that the smallest error occurred during the previous iteration
- If the error reached 0.0, the current value of $K$ was chosen
- for the given data, $KNN\_error$ produced an error of 0.0 on the 1st iteration. As a result, $K = 1$ was returned as the most optimal value of $K$

## Question 3

- The value of $\theta$ was approximated using Batch Gradient Descent Algorithm. This is because the data set is fairly small (would not noticeably benefit from another algorithm, such as the Stochastic Gradient Descent Algorithm)
- $\theta$ is initialized to $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
- The loop is terminated when $LR_error$ returned a value less than $10^{-5}$.
- The learning rate ($\alpha$) is kept constant at 0.1, with which an accurate value for the weights is produced after only 1 iteration, with a misclassification error of 0.0
- The final value of $\theta$ is $\begin{bmatrix} -0.10 \\ -0.09 \\ 0.10 \end{bmatrix}$

## Question 4

- The testing and training errors for both $KNN$ and $LR$ were 0.0
- This was achieved through adequate feature selection, using features that managed to separate the classes into 2 distinguishable clusters




- This error does not, however, properly indicate the accuracy of either model.
- To better visualize the accuracy of these models, a larger data set would be required
- if the accuracy decreases on a more diverse dataset, the feature selection process may need to be reconsidered, or perhaps a different machine learning technique could be considered