School of Engineering and Applied Science (SEAS), Ahmedabad University

B.Tech (ICT) Semester VI: Machine Learning (CSE 523)

- Group No : S_ECC7
- Name (Roll No) : Rajvi Patel (AU1741078)
- Project Title : Estimating the effect of climate change on sea level using Machine Learning
- Project Area : Environment and Climate Change

1 Introduction

1.1 Background

Climate change has certainly become one of the greatest threats to humanity. It is characterized as the change in global or regional patterns in climate. The average surface temperature is increasing rapidly due to the release of greenhouse gases in the air. Sea levels have also arisen as a result of increased global temperatures. This causes to increment in summer melting of glaciers than average summer melting as well as diminished snowfall due to later winters and earlier springs. That creates an imbalance between surface runoff and ocean evaporation, causing sea levels to rise [1]. As almost half of the world's population lives in coastal regions, the climate change due to increase in sea level is most concerning problem.Having knowledge about increase in sea level can help governments and other interested entities in disaster prevention, real estate evaluation and public safety. For example beach erosion, inundation of low lying areas, salt water intrusion into aquifers and increased flooding [2].

Most researches explored the global sea-level rise predictions using yearly data of sea level [3]. A semi-empirical relation is an approach that connects global sea-level rise to global mean surface temperature for prediction [4] which concludes that the rate of sea-level rise is roughly proportional to the magnitude of the temperatures. Greenhouse gases have also a major impact on sea-level rise [5]. As a core requirement of marine meteorology and operational oceanography, sea-level fluctuations are predicted using extreme learning machine and relevance vector machine [6].In the article [7], a four-parameter linear response equation is used to relate global temperature of the past 2,000 years and sea level and using Monte Carlo inversion they estimated likelihood distributions of equation parameters, which then allows visualization of past and future sea-level scenarios. A combination of a convolutional neural network (CNN) and a recurrent neural network (RNN) is also used to analyze both the spatial and the temporal evolution of sea level and to suggest an independent, accurate method to predict sea level anomalies in article [8], [9] and [10].

In the article [11], they relate annual global sea-level rise to temperature and predicted sea level for the year 2020, having features from 2013 [12]. In order to achieve the best prediction they have worked on three machine learning algorithms: random forest, support vector regression and neural network. The performance of machine learning algorithms under various conditions was measured using cross-validation on the latest 15% of the data set. They have chosen this selection because restricting the training set further, to 70% for example, was yielding relatively bad testing errors. Their figure of merit, measuring the performance of the algorithms was the average percent testing error. On these results they have concluded that for the local(San Francisco) sea level, the best performing algorithm was support vector regression and neural network for the global sea level [13].

1.2 Motivation

As mentioned above many researchers have worked on sea level rise prediction but only considering one or two parameter. They haven't considered all parameters (like temperature, snow-melting,green house gases etc.) which could affect sea level rise. Besides improving the model by virtue of having more features, I am interested in looking at which features have the biggest impact on the prediction. The feature selection is performed manually by removing individual features to see, which ones have the most influence on the prediction. Features, that do not change or even exacerbated the prediction is excluded.

1.3 Problem Statement

Sea-level rise affects the lives of people in coastal areas and it is causing the sea to inundate villages during high tide season which has changed people's lives because houses are destroyed and areas, where families gather, are being washed away. By predicting sea-level we can save their lives. Initially, for the local prediction, I have considered the following features which are affecting the most to rise of sea level: Local precipitation, Local temperature, Local population, Local CO_2 concentration and for the global prediction, I have applied a different set of features: Global precipitation, Global temperature index, Global population, Global CO_2 concentration. If we increased features than prediction will get even worse. So we have to choose features that affect the most to the prediction. The performance of linear regression algorithms under various conditions is derived using cross-validation on the latest 20% of the dataset, which was left out from the dataset.

2 Data Acquisition

All the feature and target data are mainly collected from National Oceanic and Atmospheric Administration (NOAA). The main data resources besides NOAA were the National Aeronautics and Space Administration (NASA), University of Hawaii Sea Level Center (UHSLC), Earth Policy Institute (EPI), and

Western Regional Climate Center (WRCC). The preprocessing that was performed on the data consisted of importing from CSV, text and GRIB formats to MATLAB,is bridging over missing values. Finally, the data was normalized by subtracting the mean and dividing by the variance.

3 Machine Learning Concept Used

- **Regression**

  Regression algorithms are used to predict output based on independent variables as a input. There are two type of regression.

  (a) Linear Regression

  When data is linear or we can say that input and output have linear relationship than linear regression can be used for prediction.

  (b) Polynomial Regression

  When features were correlated but the relationship between them doesn't look like linear, in such cases we can use polynomial regression to fit a polynomial line on dataset in order to achieve minimum error.

- **Dimentionality reduction**

  Now,In terms of performance, having data of high dimensionality cause following problems:

  (a) high computational cost to perform learning and inference

  (b) leads to overfitting when learning a model, which means that the model will perform well on the training data but poorly on test data.

  Both of these problems mentioned above could be addressed using dimensionality reduction , while preserving most of the relevant information in the data needed to learn accurate, predictive models. To implement this concept Principal Component Analysis is used.There are two cases,when the dimensionality of dataset is smaller than the number of samples and the dimensionality of dataset is larger than the number of samples.In my problem, I have considered five features which are Local precipitation, Local temperature, Local population, Local CO_2 concentration and the Land Ocean Temperature Index. Firstly I have used four features only. To make higher dimensions data I have tried to add more features but I could only able to add one feature which is the Land Ocean Temperature Index and I couldn't increase these features more to satisfied the condition that dimensionality of dataset is larger than the number of samples. Because no any other feature can predict sea level rise accurately.So I have gone through the first case.

- **Gaussian Mixture Model** The GMM is a statistical model which assumes that every probability distribution can be approximated with a set of gaussians. GMM models the distribution of each class

as a set of weighted gaussians. It has two main steps. First is the E-step, which stands for expectation. Second comes the M-step, which stands for maximization. In this step, we maximize the lower bound of the log-likelihood function by generating a new set of parameters with respect to the expectations. Here GMM is used for cluster observation.

4 Pseudo Code/ Algorithm

- **Regression**

  Polynomial regression is used for prediction because of non-linearity of the dataset.

  Step 1 : The dataset is formed into feature matrix which is Φ and added one column of 1's to meet the dimensions.

  Step 2 : Using following equation θ_{ML} is calculated for a hypothesis function to predict the sea level,

$$\theta_{ML} = (\Phi^T \Phi)^{-1} \Phi^T y$$

  Step 3 : After deriving θ prediction can be done using hypothesis function,

$$y = \theta_0 + \theta_1(x_1 + x_2 + .. + x_m) + \theta_2(x_1^2 + x_2^2 + .. + x_m^2) + ... + \theta_k(x_1^k + x_2^k + .. + x_m^k)$$

$$\therefore Y = \theta_0 + \theta_1 X + \theta_2 X^2 + ... + \theta_k X^k$$

  Step 4 : Observing the value of y for training and testing data,

$$R_{emp}(test) > R_{emp}(train)$$

  which leads to overfitting.

  Step 5 : To minimize the risk regularization parameter is added, which reduces the variance of estimated regression parameters. however, it does this at the expense of adding bias to the estimate. In regularized least squares, we consider the loss function as,

$$||y - \Phi\theta||^2 + \lambda||\theta||^2$$

  Step 6 : On minimizing the loss, we obtain a solution that closely resembles the MAP estimate

$$\theta_{RLS} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y$$

  where, $\lambda = \dfrac{\sigma^2}{b^2}$, σ^2 is noise variance and b^2 is the variance of the (isotropic) Gaussian prior p(θ)=N(0,$b^2 I$).

  Step 7 : To calculate noise variance,

$$\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^{N} (y_n - \phi^T(x_n)\theta)^2$$

Step 8 : θ_{MAP} can be written as,

$$\theta_{MAP} = (\Phi^T\Phi + \frac{\sigma^2}{b^2})^{-1}\Phi^T y$$

Step 9 : After deriving θ_{MAP} prediction can be done using following hypothesis function,

$$y = X^T\theta_{MAP}$$

θ 's are actually the weights assigned to a particular feature. Not all parameters affect the hypothesis function equally. Different features contribute to rise differently. The whole purpose of writing the ML algorithm is to come-up with values of θ that will make our hypothesis function fit the training set well.

- **PCA**

  In PCA, we are supposed to find projections of data points $\tilde{x_n}$ that are as similar to the original data points as possible, but which have a significantly lower intrinsic dimensionality.

  Step 1 : Standardize the dataset.

  Consider an i.i.d dataset $X = \{x_1, x_2, x_3, x_4, x_5\}$ where features (x_n) are Local precipitation, Local temperature, Local population, Local CO_2 concentration and the Land Ocean Temperature Index.

$$\bar{X} = \frac{X - \mu}{\sigma}$$

  Step 2 : Calculate the covariance matrix for the features in the dataset.

$$S = \bar{X}^T * \bar{X}$$

  Step 3 : Compute eigen values and eigen vector of covariance matrix S

$$Ax = \lambda x$$

  Step 4 : Sort eigenvalues and their corresponding eigenvectors.

  Step 5 : Pick M eigenvalues and form a matrix of eigenvectors.

$$B = [b_1, b_2, b_3, ..., b_M] \in R^{D*M}$$

  Here the columns of B are orthonormal(eigen vectors).

  Step 6 : Compute Projection matrix P,

$$P = B * B^T$$

  The shape of the projection matrix is (5*5) which is(D*D)

  Step 7 : Transform the original matrix.

$$X_{reconstruct} = P * \bar{X}^T$$

Step 8 : Calculate Mean square error and plot the graphs.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2$$

where y_i=predicted value and $\bar{y}_i$=actual value

- **Gaussian Mixture Model**

  Gaussian mixture model implements expectation-maximization(EM) algorithm which is implemented as following:

  Step 1 : Intialization of the parameters number of clusters, mean μ_k, covariance σ_k, and fraction per class π_k.

  Step 2 : Calculate for each datapoint x_n the probability r_{nk} that datapoint x_n belongs to cluster k with:

$$r_{nk} = \frac{\pi_n * \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\Sigma_{j=1}^{K}\pi_j * \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

  where $\mathcal{N}(x|\mu, \Sigma)$ describes the mulitvariate Gaussian with:

$$\mathcal{N}(x_n|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_k|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(x_n - \mu_k)^T\Sigma_k^{-1}(x_n - \mu_k)\right)$$

  Step 3 : For each cluster k, calculate the total weight m_k and update μ_k, σ_k, and π_k using r_{nk} with:

$$N_k = \Sigma_n r_{nk}$$

$$\pi_k^{new} = \frac{N_k}{N}$$

$$\mu_k^{new} = \frac{1}{m_k}\Sigma_n r_{nk} x_n$$

$$\Sigma_k^{new} = \frac{1}{m_k}\Sigma_n r_{nk}(x_n - \mu_k)^T(x_n - \mu_k)$$

  Step 4 : Iteratively repeat step 2 and 3 until the log-likelihood function converges where the log likelihood is computed with:

$$\ln p(\mathcal{X}|\pi, \mu, \Sigma) = \Sigma_{n=1}^{N} ln\left(\Sigma_{k=1}^{K}\pi_k\mathcal{N}(x_n|\mu_k, \Sigma_k)\right)$$

- **Support Vector Machine**

  Step 1 : Preprocessing of data and extract X and y. Split the data into training and testing.

  Step 2 : Fit training set to the SVM classifier.

  Step 3 : Adjust performance of model by changing C(Regularization factor) and gamma.

  Step 4 : Predict output for test set.

  Step 5 : Compare the result of y_{pred} and y_{test} to calculate accuracy and confusion matrix.

5 Coding and Simulation

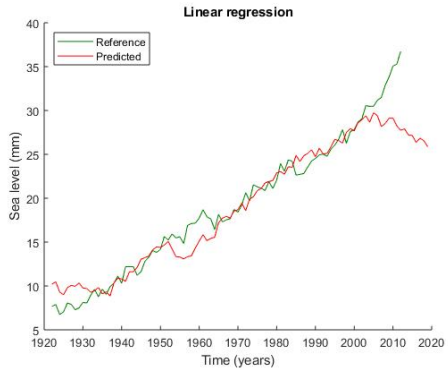
5.1 Simulation Framework

For random forest, pre-compiled function is used which takes the training feature matrix, the training target vector and the parameters n_{trees} and m_{try} as inputs. In this algorithm each tree is created by splitting the feature set randomly into m_{try} subsets.Initially, the random forest was overfitting, which was solved by setting the number of trees to 500. The two layer feed-forward neural network was overfitting as well, but this behavior was solved by specifying a validation fraction. The training algorithm from the MATLAB Neural Network toolbox sets this fraction of the training set aside as validation set from the dataset, when the error stops improving, the training process is stopped and returns the model.

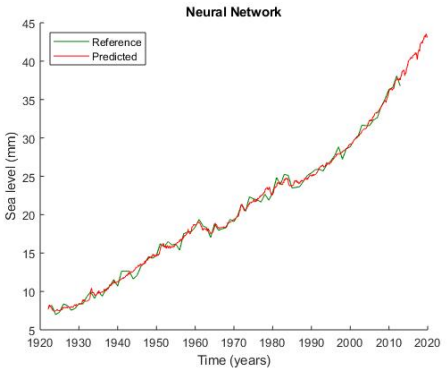
5.2 Results

- Used Tools: MATLAB and Python

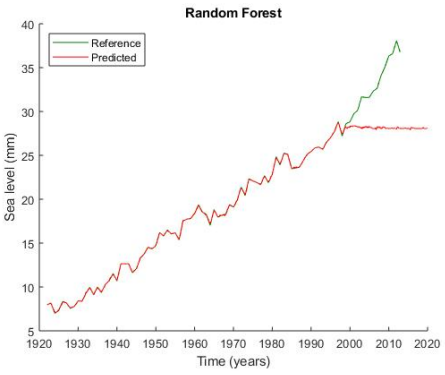
  1. Reconstructed results



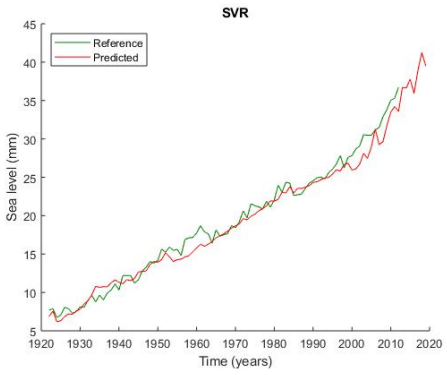
(a) Prediction using Linear regression



(b) Prediction using Neural network



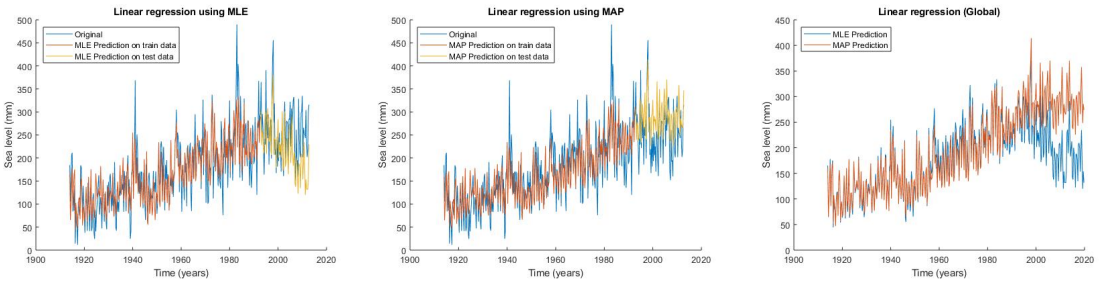
(c) Prediction using Random forest



(d) Prediction using Support vector regression

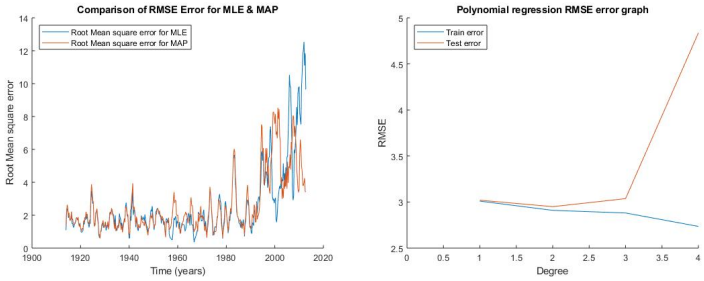
Figure 1: Prediction using different algorithms

2. Regression results



(a) MLE Prediction (b) MAP Prediction (c) Overlapping of MLE and MAP

Figure 2: Linear Regression using MLE and MAP

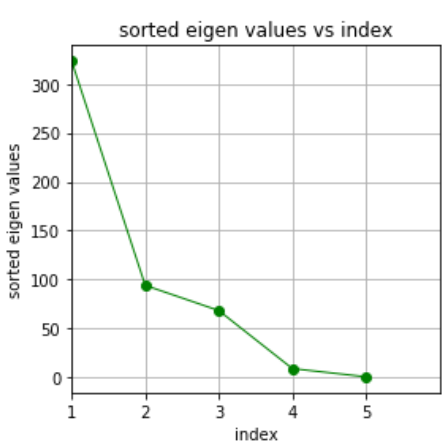


(a) RMSE for MLE and MAP (b) degree vs. RMSE error

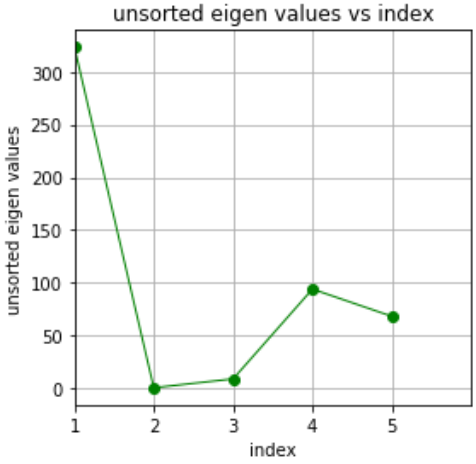
Figure 3: Error graphs in MLE and MAP

Here in figure 2.a sea level rise is predicted using maximum likelihood estimation and in figure 2.b sea level rise is predicted using MAP estimation. Both graphs are plotted along with original data. In figure 2.c MLE and MAP both are overlapped. Figure 3.a represents RMSE(root mean square error) of MLE and MAP estimation & 3.b represents graph for degree of polynomial vs. RMSE error.

3. Dimentionality Reduction results

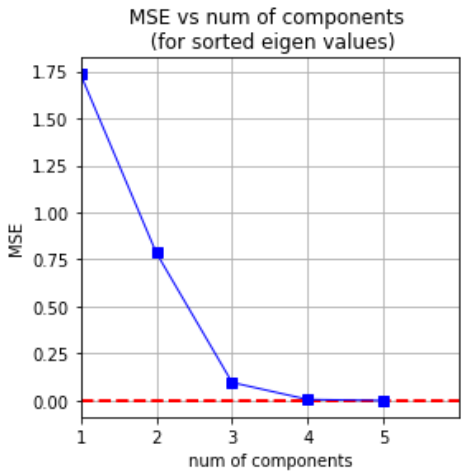


(a) Sorted eigen values vs index

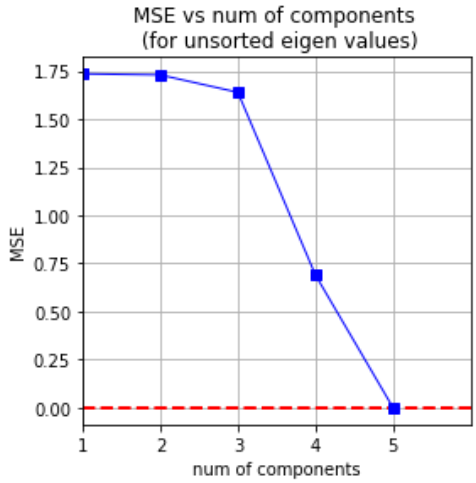


(b) Unsorted eigen values vs index

Figure 4: eigen values vs index



(a) MSE vs index (Sorted eigen vals)



(b) MSE vs index (Unsorted eigen vals)

Figure 5: MSE vs. number of components

Figure 4 represents graph of sorted and unsorted eigen values. Figure 5 represents mean square error for both sorted and unsorted eigen values. In figure 6 variance is plotted with respect to different number of principle components.

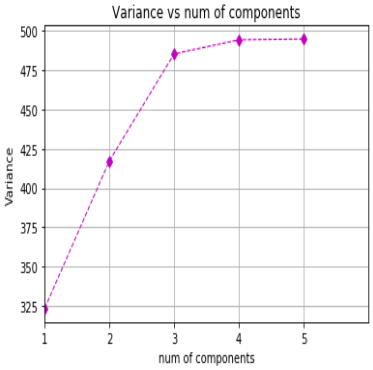
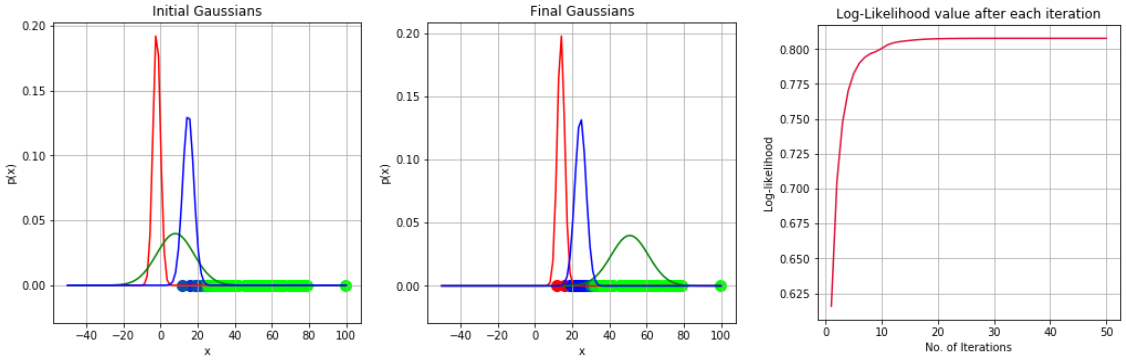


Figure 6: Variance vs. number of components

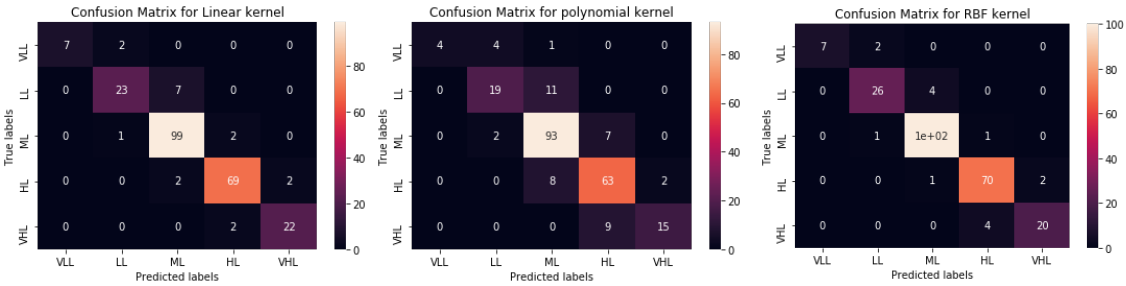
4. GMM results



(a) Initial Gaussian PDF (b) Final Gaussian PDF (c) Log Likelihood value

Figure 7: Gaussian Mixture Model

5. SVM results



(a) Linear Kernel (b) Polynomial Kernel (c) RBF Kernel

Figure 8: SVM Classification

6 Conclusions

(a) Reproduced results

		Global Sea-level Rise		Local Sea-level Rise	
		Random Forest	**Neural Network**	**Random Forest**	**Neural Network**
RMSE	**Seasonal**	5.6	0.25	45	27
(mm)	**Yearly**	5.5	0.48	29	26
Training	**Seasonal**	0.39	1.2	7.9	16
Error(%)	**Yearly**	1.4	3.6	6.9	17
Testing	**Seasonal**	15	0.85	14	8.0
Error(%)	**Yearly**	15	1.1	8.9	8.1

From the above table, Neural network appears to be the most suitable learning algorithm for this case and it does not overfit.It does a good job on the cross-validation and presents a believable prediction. Random forest algorithm overfits the training set. The support vector regression algorithm matches well the sub-year variations, but has a significant negative bias compared to the reference.

(b) Regression

RMSE Error(mm)		
Model	**Training error**	**Testing Error**
MLE	12.8231	19.8554
MAP	13.2227	12.0539

All of the above graph of regression and from the first row of table we can say that MLE model models the training data too well but fails to perform well on the testing data which leads to overfitting.

$$R_{emp}(test) > R_{emp}(train)$$

Performing sufficiently good on testing data is considered as a kind of ultimatum in machine learning. Thus, to avoid overfitting λ is used which is called regularization parameter. λ trades off between minimizing the loss on training set and magnitude of the parameter. If λ is higher then it adds high penalty to error term. Regularization parameter λ reduces almost 40% testing error and forced to fit model on unseen data closely. Figure 3.a represents RMSE(root mean square error) of MLE and MAP estimation. For training data MLE model predicts output perrfectly but for testing data RMSE error increasing exponentially and goes to the peak level. After doing regularization this peak level is forced to goes down. Figure 3.b describe the behaviour of polynomial degree on RMSE error. RMSE error is in range till polynomial degree 3 but after that it increases exponentially which means the model is capturing higher noise in data and leads overfitting. Even though model passes from most of the data but it will fail to generalize on unseen data. Using above observation we can derive following inferences:

	Low training error	**High training Error**
Low testing error	The model is training	Error in code
High testing error	Overfitting	The model is not training

(c) Dimentionality reduction

As shown in figure 5.a, the graph of MSE is getting almost zero for number of components M=3 and M=4. So we can use the number of components needed for proper reconstruction is 3. While increasing the number of components the mean square error(MSE) is decreasing which means we can reconstruct X with minimum data loss.

As shown in figure 5.a and 5.b, for sorted eigen values we can say that MSE will always decreased exponentially while increasing number of components. But for unsorted eigen values decrement in MSE is less than as for sorted eigen values. So for unsorted eigen values MSE is high while increasing number of components which shows more data loss in reconstruction of X. In figure 6.a, Variance is increasing while increasing number of components. Which means we are maximizing the variance to preserve more information and less loss of data. So,maximizing the variance is for preserving as much variability (or information) as possible during the process of reducing the dimension of the data. Thus, we can say that maximizing variance in principal component space is equivalent to minimizing least-squares reconstruction error.

(d) Gaussian Mixture Model

A GMM assumes that the data is made up of a mixture of several Gaussian distributions and given different mixture weights. The final distribution is obtained by multiplying each mixture component by its associated mixture weight before and adding them together. As we can see in figure 7.b all data points are classified in 3 clusters perfectly. Graph of log likelihood function shows that this function converges as the number of iterations becomes large and there will be no improvement in GMM after some point. So we can stop the algorithm to iterate.

(e) Support Vector Machine

From figure 8, linear and polynomial kernels are less time consuming and provides less accuracy than the rbf kernel. Both linear and RBF kernel SVM work equally well on the dataset. Error in kernels are as following:

- Linear Kernel: 0.92
- Polynomial Kernel: 0.82
- RBF Kernel: 0.94

Polynomial kernel is less useful for efficiency (computational as well as predictive) performance reasons. The squared exponential kernel is generally more flexible than the linear or polynomial kernels in that you can model a whole lot more functions with its function space.Because,the complexity of the rbf

kernel is potentially infinite. Thus it's complexity can grow with the data. If you give it more and more data, it will be able to represent more and more complex relationships. In contrast a parametric model's size is fixed, so after a certain point your model will be saturated, and giving it more and more data won't help.So, we can say that use linear SVM for linear problems, and nonlinear kernels such as the Radial Basis Function kernel for non-linear problems.

References

[1] V. Masson-Delmotte, M. Schulz, A. Abe-Ouchi, J. Beer, A. Ganopolski, J. F. González Rouco, E. Jansen, K. Lambeck, J. Luterbacher, T. R. Naish, T. Osborn, B. L. Otto-Bliesner, T. M. Quinn, R. Ramesh, M. Rojas, X. Shao, and A. Timmermann, "Information from Paleoclimate Archives," 2013.

[2] R. Nicholls and A. Cazenave, "Sea-level rise and its impact on coastal zones," *Science (New York, N.Y.)*, vol. 328, pp. 1517–20, 06 2010.

[3] A. Slangen, M. Carson, C. Katsman, R. Wal, A. Köhl, B. Vermeersen, and D. Stammer, "Projecting twenty-first century regional sea-level changes," *Climatic Change*, vol. 124, pp. 317–332, 03 2014.

[4] S. Rahmstorf, "A semi-empirical approach to projecting future sea-level rise," *Science (New York, N.Y.)*, vol. 315, pp. 368–70, 02 2007.

[5] J. Gregory and J. Lowe, "Predictions of global and regional sea-level rise using aogcms with and without flux adjustment," *Geophysical Research Letters - GEOPHYS RES LETT*, vol. 27, pp. 3069–3072, 10 2000.

[6] M. Imani, H. C. Kao, W. H. Lan, and C. Kuo, "Daily sea level prediction at chiayi coast, taiwan using extreme learning machine and relevance vector machine," *Global and Planetary Change*, vol. 161, 12 2017.

[7] A. Grinsted, J. Moore, and S. Jevrejeva, "Reconstructing sea level from paleo and projected temperatures 200 to 2100ad," *Climate Dynamics*, vol. 34, pp. 461–472, 03 2009.

[8] A. Braakmann-Folgmann, R. Roscher, S. Wenzel, B. Uebbing, and J. Kusche, "Sea level anomaly prediction using recurrent neural networks," 10 2017.

[9] M. Wenzel and J. Schröter, "Reconstruction of regional mean sea level anomalies from tide gauges using neural networks," *Journal of Geophysical Research: Oceans*, vol. 115, no. C8, 2010.

[10] O. Makarynskyy, D. Makarynska, M. Kuhn, and W. Featherstone, "Predicting sea level variations with artificial neural networks at hillarys boat harbour, western australia," *Estuarine, Coastal and Shelf Science*, vol. 61, no. 2, pp. 351 – 360, 2004.

[11] J. K. Mala Alahmadi, "Estimating the effect of climate change on global and local sea level rise," 2015.

[12] A. Artegiani, A. Giommoni, A. Goldmann, P. Sguazzero, and A. Tomasin, "Sea level prediction models for venice." pp. 213–221, 05 1973.

[13] A. Slangen, C. Katsman, L. Vermeersen, and R. Riva, "Towards regional projections of twenty-first century sea-level change based on ipcc sres scenarios," *Climate Dynamics*, vol. 38, pp. 1191–1209, 2012.