Introduction to Machine Learning (67577)

Exercise 2
Linear Regression

Second Semester, 2022

Contents

1 Submission Instructions 2

2 Theoretical Part 2

2.1 Solutions of The Normal Equations 2

2.2 Projection Matrices 3

2.3 Least Squares 4

3 Practical Part 6

3.1 Fitting A Linear Regression Model 6

3.2 Polynomial Fitting 8

1 Submission Instructions

Please make sure to follow the general submission instructions available on the course website. In addition, for the following assignment, submit a single `ex2_ID.tar` file containing:

- An `Answers.pdf` file with the answers for all theoretical and practical questions (include plotted graphs *in* the PDF file).
- The following python files (without any directories): `linear_regression.py`, `polynomial_fitting.py`, `loss_functions.py`, `utils.py`, `house_price_prediction.py`, `city_temperature_prediction.py`

The `ex2_ID.tar` file must be submitted in the designated Moodle activity prior to the date specified *in the activity*.

- Late submissions will not be accepted and result in a zero mark.
- Plots included as separate files will be considered as not provided.
- Do not forget to answer the Moodle quiz of this assignment.

2 Theoretical Part

Let $\mathbf{X}$ be the design matrix of a linear regression problem with m rows (samples) and d columns (variables/features). Let $\mathbf{y} \in \mathbb{R}^m$ be the response vector corresponding the samples in $\mathbf{X}$. Recall that for some vector space $V \subseteq \mathbb{R}^d$ the orthogonal complement of V is: $V^\perp := \left\{\mathbf{x} \in \mathbb{R}^d | \left\langle \mathbf{x}, \mathbf{v} \right\rangle = 0 \quad \forall \mathbf{v} \in V\right\}$

2.1 Solutions of The Normal Equations

1. Prove that: $Ker\left(\mathbf{X}\right) = Ker\left(\mathbf{X}^\top \mathbf{X}\right)$

Let $0 \neq \mathbf{u} \in Ker\left(X\right)$ then:

$$\mathbf{X}\mathbf{u} = 0 \quad \Rightarrow \quad \mathbf{X}^\top \mathbf{X}\mathbf{u} = 0$$

and therefore $\mathbf{u} \in Ker\left(\mathbf{X}^\top \mathbf{X}\right)$. In the other direction let $0 \neq \mathbf{u} \in Ker\left(\mathbf{X}^\top \mathbf{X}\right)$ then it holds that:

$$0 = \mathbf{X}^\top \mathbf{X}\mathbf{u} = \mathbf{u}^\top \mathbf{X}^\top \mathbf{X}\mathbf{u} = ||\mathbf{X}\mathbf{u}||^2$$

From the properties of norms it equals zero if and only if the vector is the zero vector. Therefore $\mathbf{X}\mathbf{u} = 0$ and $\mathbf{u} \in Ker\left(\mathbf{X}\right)$.

2. Prove that for a square matrix A: $Im\left(A^\top\right) = Ker\left(A\right)^\perp$

Let $\mathbf{b} \in Im\left(A^\top\right)$. Then there is $\mathbf{x} \in \mathbb{R}^m$ with $A^\top \mathbf{x} = \mathbf{b}$. Let $\mathbf{w} \in Ker\left(A^\top\right)$:

$$A^\top \mathbf{w} = 0 \Rightarrow \; 0 = \left\langle A^\top \mathbf{w}, \mathbf{x} \right\rangle = \left\langle \mathbf{w}, A\mathbf{x} \right\rangle = \left\langle \mathbf{w}, \mathbf{b} \right\rangle$$

Therefore $Im\left(A\right) \subset Ker\left(A^\top\right)^\perp$. To show the converse inclusion $Ker\left(A^\top\right)^\perp \subset Im\left(A\right)$ we assume that $\mathbf{b} \notin Im\left(A\right)$ and show that $\mathbf{b} \notin Ker\left(A^\top\right)^\perp$. It is enough to find a vector $\mathbf{c} \in Ker\left(A^\top\right)$ such that $\left\langle \mathbf{b}, \mathbf{c} \right\rangle \neq 0$. Indeed, since by assumption $\mathbf{b} \notin Im\left(A\right)$, the vector $\mathbf{b}$

> must have a component in $Im(A)^{\perp}$. Let $\mathbf{c} \in Im(A)^{\perp}$ such that $\langle \mathbf{b}, \mathbf{c} \rangle \neq 0$.
> Now, since $\mathbf{c} \in Im(A)^{\perp}$ we know that $\mathbf{c}$ is orthogonal to any vector in $Im(A)$. In particular $\left\langle \mathbf{c}, AA^{\top}\mathbf{c} \right\rangle = 0$. Therefore
>
> $$\left|\left|A^{\top}\mathbf{c}\right|\right|^2 = \left\langle A^{\top}\mathbf{c}, A^{\top}\mathbf{c} \right\rangle = \left\langle \mathbf{c}, AA^{\top}\mathbf{c} \right\rangle = 0$$
>
> Hence: $A^{\top}\mathbf{c} = 0 \;\Rightarrow\; \mathbf{c} \in Ker\left(A^{\top}\right)$, so that $\mathbf{c}$ is as required.

3. Let $\mathbf{y} = \mathbf{X}\mathbf{w}$ be a non-homogeneous system of linear equations. Assume that $\mathbf{X}$ is square and not invertible. Show that the system has ∞ solutions $\Leftrightarrow y \perp Ker\left(\mathbf{X}^{\top}\right)$.

> Since $\mathbf{X}$ is not invertible, the system has either 0 solutions or ∞ solutions. Furthermore we know that the system has at least one solution if and only if $\mathbf{y} \in Im(\mathbf{X})$. As such, it follows that:
>
> $$\begin{aligned} \text{The system has } \infty \text{ solutions} \quad &\Longleftrightarrow \quad \mathbf{y} \in Im(\mathbf{X}) \\ &\overset{question\ 2}{\Longleftrightarrow} \quad \mathbf{y} \in Ker(\mathbf{X})^{\perp} \\ &\Longleftrightarrow \quad \mathbf{y} \perp Ker(\mathbf{X}) \end{aligned}$$

4. Consider the (normal) linear system $\mathbf{X}^{\top}\mathbf{X}\mathbf{w} = \mathbf{X}^{\top}\mathbf{y}$. Using what you have proved above prove that the normal equations can only have a unique solution (if $\mathbf{X}^{\top}\mathbf{X}$ is invertible) or infinitely many solutions (otherwise).

> We assume that $\mathbf{X}^{\top}\mathbf{X}$ is not invertible. By the previous question, the system has ∞ solutions if and only if $\mathbf{X}^{\top}\mathbf{y} \perp Ker\left(\mathbf{X}^{\top}\mathbf{X}\right)$. However, recall that $Ker(\mathbf{X}) = Ker\left(\mathbf{X}^{\top}\mathbf{X}\right)$. Therefore, all we need is to show that $\mathbf{X}^{\top}\mathbf{y} \perp Ker(\mathbf{X})$. Indeed, if $\mathbf{u} \in Ker(\mathbf{X})$ then $\left\langle \mathbf{u}, \mathbf{X}^{\top}\mathbf{y} \right\rangle = \langle \mathbf{X}\mathbf{u}, \mathbf{y} \rangle = 0$. So we have proved that the normal equations can only have a unique solution (if $\mathbf{X}^{\top}\mathbf{X}$ is invertible) or ∞ solutions (otherwise).

2.2 Projection Matrices

5. In this question you will prove some properties of orthogonal projection matrices seen in recitation 1. Let $V \subseteq \mathbb{R}^d$, $dim(V) = k$ and let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be an orthonormal basis of V. Define the orthogonal projection matrix $P = \sum_{i=1}^{k} \mathbf{v}_i \mathbf{v}_i^{\top}$ (notice this is an outer product).

   Prove the following properties in any order you wish:

   (a) Show that P is symmetric.

   > P is sum of symmetric metrices (outer product is symmetric).

   (b) Prove that the eigenvalues of P are 0 or 1 and that $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are the eigenvectors corresponding the eigenvalue 1.

   > $P\mathbf{v}_j = \sum_{i=1}^{k} \mathbf{v}_i \mathbf{v}_i^{\top} \mathbf{v}_j = \sum_{i=1}^{k} \mathbf{v}_i \delta_{ij} = \mathbf{v}_j$

   (c) Show that $\forall \mathbf{v} \in V \;\; P\mathbf{v} = \mathbf{v}$.

$\mathbf{x} \in V \Rightarrow \mathbf{x} = \sum_{i=1}^{k} \alpha_i \mathbf{v}_i \Rightarrow P\mathbf{x} = P\sum_{i=1}^{k} \alpha_i \mathbf{v}_i = \sum_{i=1}^{k} \alpha_i P\mathbf{v}_i = \sum_{i=1}^{k} \alpha_i \mathbf{v}_i$

(d) Prove that $P^2 = P$.

$P^2 = UDU^T UDU^T = UDDU^T = UDU^T = P$

(e) Prove that $(I-P)P = 0$.

$$(I-P)P = P - P^2 = P - P = 0$$

2.3 Least Squares

Given a sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, the ERM rule for linear regression w.r.t. the squared loss is

$$\hat{\mathbf{w}} \in \underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}} \ ||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$$

where $\mathbf{X}$ is the design matrix of the linear regression with rows as samples and y the vector of responses. Let $\mathbf{X} = U\Sigma V^\top$ be the SVD of $\mathbf{X}$, where U is a $m \times m$ orthonormal matrix, Σ is a $m \times d$ diagonal matrix, and V is an $d \times d$ orthonormal matrix. Let $\sigma_i = \Sigma_{i,i}$ and note that only the non-zero σ_i-s are singular values of $\mathbf{X}$. Recall that the pseudoinverse of $\mathbf{X}$ is defined by $\mathbf{X}^\dagger = V\Sigma^\dagger U^\top$ where $\Sigma^\dagger$ is an $d \times m$ diagonal matrix, such that

$$\Sigma^\dagger_{i,i} = \begin{cases} \sigma_i^{-1} & \sigma_i \neq 0 \\ 0 & \sigma_i = 0 \end{cases}$$

6. Show that if $\mathbf{X}^\top\mathbf{X}$ is invertible, the general solution we derived in recitation equals to the solution you have seen in class. For this part, assume that $\mathbf{X}^\top\mathbf{X}$ is invertible.

This follows by substituting $\mathbf{X}$ with the SVD of $\mathbf{X}$. Let $\mathbf{X} = U\Sigma V^\top$ be the SVD of $\mathbf{X}$ so:

$$\begin{aligned} \left[\mathbf{X}^\top\mathbf{X}\right]^{-1}\mathbf{X}^\top &= \left[\left(U\Sigma V^\top\right)^\top \left(U\Sigma V^\top\right)\right]^{-1} \left(U\Sigma V^\top\right)^\top \\ &= \left[V\Sigma^\top\Sigma V^\top\right]^{-1} V\Sigma^\top U^\top \\ &= V\left(\Sigma^\top\Sigma\right)^{-1} \cancel{V^\top V}\Sigma^\top U^\top \\ &= VD^{-1}\Sigma^\top U^\top \end{aligned}$$

where $D := \Sigma^\top\Sigma$ is a $d+1$-by-$d+1$ diagonal matrix with $D_i i = \sigma_i^2$. Notice that as the columns of $\mathbf{X}$ are linearly independent then $\sigma_1 \geq \ldots \geq \sigma_{d+1} > 0$ and therefore D^{-1} is well defined. Next, notice that:

$$\left[D^{-1}\Sigma^\top\right]_{ii} = \frac{1}{\sigma_i^2}\sigma_i = \frac{1}{\sigma_i} = \Sigma^\dagger_{i,i}$$

And therefore we conclude that in the singular case:

$$\left[\mathbf{X}^\top\mathbf{X}\right]^{-1}\mathbf{X}^\top\mathbf{y} = V\Sigma^\dagger U^\top\mathbf{y} = \mathbf{X}^{\dagger\top}\mathbf{y}$$

7. Show that $\mathbf{X}^\top\mathbf{X}$ is invertible if and only if $\operatorname{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} = \mathbb{R}^d$.

The rank of $\mathbf{X}$ is equal to the dimension of the subspace span$\{\mathbf{x}_1,\dots,\mathbf{x}_m\}$. The SVD theorem implies that the rank of $\mathbf{X}$ is equal to the rank of $\mathbf{X}^\top\mathbf{X}$. Hence, $\mathbf{x}_1,\dots,\mathbf{x}_m$ span $\mathbb{R}^d$ iff $rank\left(\mathbf{X}^\top\mathbf{X}\right) = d$. Since $\mathbf{X}^\top\mathbf{X}$ is a $d\times d$ matrix, it is invertible iff its rank is d.

8. Recall that if $\mathbf{X}^\top\mathbf{X}$ is not invertible then there are many solutions. Show that $\hat{\mathbf{w}} = \mathbf{X}^\dagger\mathbf{y}$ is the solution whose L_2 norm is minimal. That is, show that for any other solution $\overline{\mathbf{w}}$, $||\hat{\mathbf{w}}|| \le ||\overline{\mathbf{w}}||$.

Hints:

- Recall that the rank of $\mathbf{X}$ and the rank of $\mathbf{X}^\top\mathbf{X}$ are determined by the number of singular values of $\mathbf{X}$. If you are not sure why this is true, go over recitation 1.
- Which coordinates must satisfy $\widehat{w}_i = \overline{w}_i$? What is the value of $\widehat{w}_i$ for the other coordinates? If you are not sure, go back to the derivation of $\widehat{\mathbf{w}}$ (see recitation 4).

Let $\mathbf{X} = U\Sigma V^\top$ be the SVD decomposition of $\mathbf{X}$. Let r be the rank of $\mathbf{X}$, and rewrite

$$V = [V_1\ V_2], \qquad U = [U_1\ U_2], \qquad \Sigma = \begin{bmatrix}\Sigma_1 & 0\\ 0 & 0\end{bmatrix}$$

where V_1, U_1 are the first r columns and V_2, U_2 the rest columns of U, V respectively, and Σ_1 is the diagonal matrix with non-zero elements.

Given some w, define $b = U^\top w$ and $b_1 = U_1^\top w, b_2 = U_2^\top w$ and therefore $b = \begin{bmatrix}b_1\\ b_2\end{bmatrix}$. U is orthonormal matrix, and therefore an isometry (that is $\left|\left|U^\top u\right|\right| = ||u||$ for any vector u), thus we get $||w|| = ||b||$, so we can rewrite our problem as showing that b has the minimal norm.

V is also an isometery, Therefore:

$$\begin{aligned}||y - X^\top w||^2 &= ||y - V\Sigma U^\top w||^2 = ||V(V^\top y - \Sigma b)||^2 = ||V^\top y - \Sigma b||^2\\ &= \left|\left|[V_1\ V_2]^\top y - \begin{bmatrix}\Sigma_1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}b_1\\ b_2\end{bmatrix}\right|\right|^2 = ||V_1^\top y - \Sigma_1 b_1||^2 + ||V_2^\top y||^2.\end{aligned}$$

Therefore, to get a minimum solution for $||\mathbf{y} - \mathbf{X}\mathbf{w}||$, the best we can do is have

$$\Sigma_1 b_1 - V_1^\top y = 0 \iff \Sigma_1 b_1 = V_1^\top y \iff b_1 = \Sigma_1^{-1} V_1^\top y.$$

But more importantly, b_2 does not take any part in the optimization problem, therefore to minimize b we want that both

$$b_1 = \Sigma_1^{-1} V_1^\top y \qquad \text{and} \qquad b_2 = 0.$$

Note that for $\hat{\mathbf{w}} = \mathbf{X}^{\top\dagger}\mathbf{y}$ we have

$$U_1^\top \hat{\mathbf{w}} = U_1^\top X^{\top\dagger}\mathbf{y} = U_1^\top U_1 \Sigma_1^{-1} V_1^\top y = \Sigma_1^{-1} V_1^\top y$$

and

$$U_2^\top \hat{\mathbf{w}} = U_2^\top X^{\top\dagger}\mathbf{y} = U_1^\top U_1 \Sigma_1^{-1} V_1^\top y = 0.$$

For any other solution $\overline{\mathbf{w}}$, we get that the first condition must satisfied, but the second condition may not be satisfied, therefore $||\widehat{\mathbf{w}}|| \leq ||\overline{\mathbf{w}}||$.

3 Practical Part

3.1 Fitting A Linear Regression Model

In this question you will have to deal with a real-world dataset and fit a linear regression model to it. As data is noisy, messy and difficult, take the time to "play" and get familiar with it.

Implement the `mean_square_error` function in the `metrics.loss_functions.py` file and the `LinearRegression` class in the `learners.regressors.linear_regression.py` file. Follow class and function documentation. Then implement code of following questions in the `exercise2/house_price_prediction.py` file.

1. Implement the `load_data` function. The function receives the path to the `house_prices` dataset, loads it as a `pandas.DataFrame` object and returns the data after preprocessing. You may decide to return it as a single data frame of both observations and response or as a tuple of the design matrix and response vector. Explore the data (some information can be found on Kaggle) and perform any necessary preprocessing (such as but not limited to):
   - What sort of values are valid for different types of features? Can house prices be negative? Can a living room size be too small?
   - Some of the features are categorical with no apparent logical order to their values (for example zip-code). Correctly address these features such that it will make sense to fit a linear regression model using them. For assistance you may refer to the following StackOverflow question.
   - Are there any additional features that might be beneficial for predicting the house price and that can be derived from existing features?

   Describe in details the analysis process that lead you to the decisions of:
   - Which features to keep and which not?
   - Which features are categorical how how did you treat them?
   - What other features did you design and what is the logic behind creating them?
   - How did you treat invalid/missing values?
   - Explain any additional processing performed on the data.

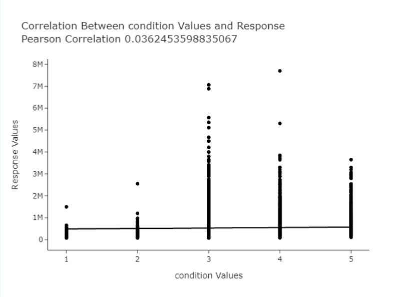
   The answers to these question should be added to your `Answers.pdf` file.

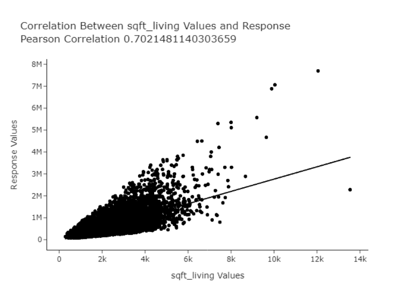
2. Basics of feature selection - implement the `feature_evaluation` as specified in the documentation. This function will compute the Pearson Correlation between each of the features and the response:

$$\text{Pearson Correlation: } \rho := \frac{COV(X,Y)}{\sigma_X \sigma_Y}$$

for X,Y being one of the features and the response. You are allowed to use functions that calculate the standard deviation and covariance, but not functions that calculate the Pearson correlation itself.

Choose two features, one that seems to be beneficial for the model and one that does not. In your `Answers.pdf` add the graphs of these two chosen features and explain how do you conclude if they are beneficial or not.


Correlation Between condition Values and Response
Pearson Correlation 0.0362453598835067


Correlation Between sqft_living Values and Response
Pearson Correlation 0.7021481140303659

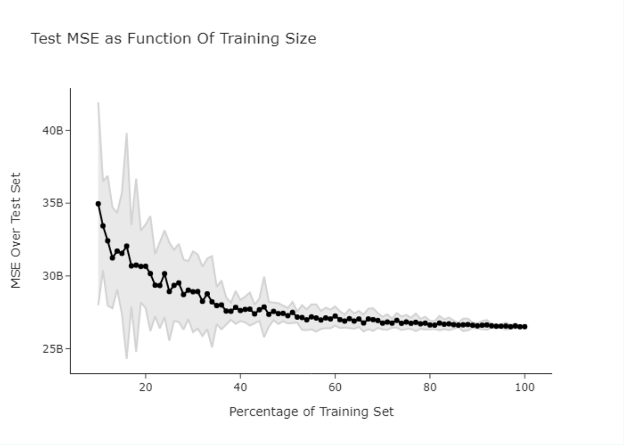
3. Implement the `split_train_test` function in the `utils.utils.py` file as described in the function documentation. Be sure to not lose which sample is connected to which response. Then, in the `house_price_prediction.py` file, split the data frame (after performing the preprocessing) to a training set (75%) and test set (25%).

4. Fit a linear regression model over increasing percentages of the *training set* and measure the loss over the *test set*:
   - Iterate for every percentage $p = 10\%, 11\%, ..., 100\%$ of the training set.
   - Sample $p\%$ of the train set. You can use the `pandas.DataFrame.sample` function.
   - Repeat sampling, fitting and evaluating 10 times for each value of p.

   Plot the mean loss as a function of $p\%$, as well as a confidence interval of $mean(loss) \pm 2 * std(loss)$. If implementing using the `Plotly` library, see how to create the confidence interval in Chapter 2 - Linear Regression code examples.

   Add the plot to the `Answers.pdf` file and explain what is seen. Address both trends in loss and in confidence interval as function of training size. What can we learn about the estimator $\hat{y}_i$ in terms of estimator properties?

   > When fitting the linear model over increasing that both the test-error decreases as well as the variance in the prediction



3.2 Polynomial Fitting

Implement the `PolynomialFitting` class in the `learners.regressors.polynomial_fitting.py` file as specified in class documentation. Avoid repeating code from the `LinearRegression` class and instead use inheritance or composition patterns. You are allowed to use the `np.vander` function.

In the following questions you will use the Daily Temperature of Major Cities dataset. Notice, that the supplied file is a modified subset of the dataset found on Kaggle containing only 4 countries. You will fit and analyse performance of a polynomial model over the dataset.

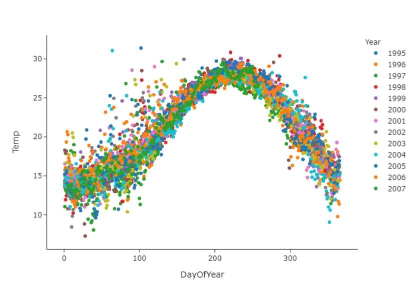
1. Implement the `load_data` function in the `city_tempretaure_prediction.py` file.
   - When loading the dataset remember to deal with invalid data.
   - Use the `parse_dates` argument of the `pandas.read_csv` to set the type of the ‘`Date`‘ column.
   - Add a ‘`DayOfYear`‘ column based on the ‘`Date`‘ column. This column will be the feature to be used for the polynomial fitting.

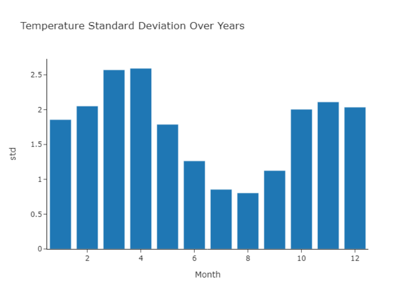
2. Subset the dataset to caintain samples only from the country of Israel. Investigate how the average daily temperature (‘`Temp`‘ column) change as a function of the ‘`DayOfYear`‘.
   - Plot a scatter plot showing this relation, and color code the dots by the different years (make sure color scale is discrete and not continuous). What polynomial degree might be suitable for this data?
   - Group the samples by ‘`Month`‘ (have a look at the pandas ‘`groupby`‘ and ‘`agg`‘ functions) and plot a bar plot showing for each month the standard deviation of the daily temperatures. Suppose you fit a polynomial model (with the correct degree) over data sampled uniformly at random from this dataset, and then use it to predict temperatures from random days across the year. Based on this graph, do you expect a model to succeed equally over all months or are there times of the year where it will perform better than on others? Explain your answer.

   Add both plots and answers to the `Answers.pdf` file.

Based on the first plot, we see that the overall behaviour of the data is similar between different years taking a wave-like shape with higher temperature at days ±200. As we can observe from the data (and know from the real world) the temperatures do not continue to drop on both sides of the peak but stabilize. As such we probably need a polynomial of degree higher than 2. It seems suitable to fit a polynomial of degree 3 or 4.

It is reasonable to assume that the model will not equally succeed in prediction across the entire year. In month of low variability (6-9) the model will probably manage to fit closer to data points. As we assume test set was generated from the same distribution as train set, prediction will probably be more accurate. On months with high variability (highest in 3-4), due to similar reasons, it is likely to assume that prediction performance will be less good.





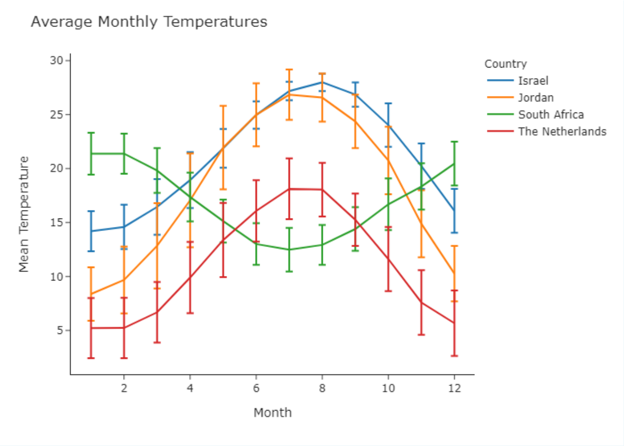
3. Returning to the full dataset, group the samples according to ‘`Country`‘ and ‘`Month`‘ and calculate the average and standard deviation of the temperature. Plot a line plot of the average monthly temperature, with error bars (using the standard deviation) color coded by the country. If using `plotly.express.line` have a look at the `error_y` argument.

   Based on this graph, do all countries share a similar pattern? For which other countries is the model fitted for Israel likely to work well and for which not? Explain your answers.

Based on the plotted graph we conclude that different countries have a different distribution of average daily temperatures to time of year. If we fit a model over the domain space of samples of Israel, it is plausible to assume better results when testing over Jordan compared to The Netherlands or South Africa.

It is interesting to note that the shape of the distribution of The Netherlands is very similar to that of Israel, with the main difference being that the Netherland's distribution is about 10 degrees lower across all range. We might be able to use the model fitted for Israel by simply adjusting the value of the intercept.

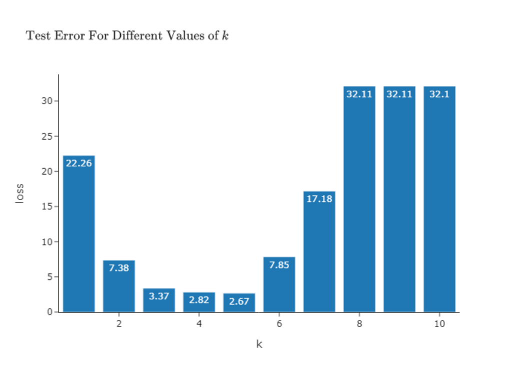
The distribution in the case of South Africa is both opposite and less extreme than that of Israel. As South Africa is on the southern hemisphere seasons (and therefore average daily temperatures) are opposite than in Israel.



4. Over the subset containing observations only from Israel perform the following:
   - Randomly split the dataset into a training set (75%) and test set (25%).
   - For every value $k \in [1,10]$, fit a polynomial model of degree k using the training set.
   - Record the loss of the model over the test set, rounded to 2 decimal places.

   Print the test error recorded for each value of k. In addition plot a bar plot showing the test error recorded for each value of k. Based on these which value of k best fits the data? In the case of multiple values of k achieving the same loss select the simplest model of them. Are there any other values that could be considered?

The model fitted with $k = 5$ achieved the lowest test error of 2.67. Slightly above that the models of $k = 4$ achieved a test error of 2.82 and $k = 3$ achieved a test error of 3.37. Based on these results we should select $k = 5$.

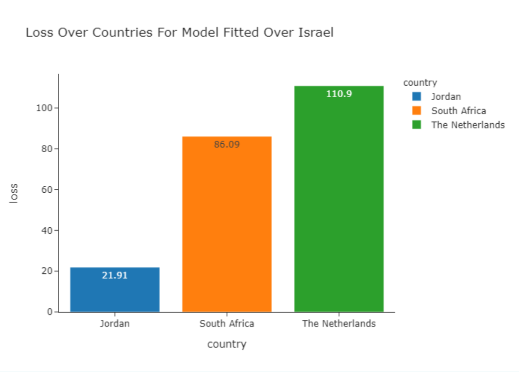


It is important to note that for a different random split of the dataset we might have achieved different values, leading us to choose a different value of k. We will address this issue later on in the course.

5. Fit a model over the entire subset of records from Israel using the k chosen above. Plot a bar plot showing the model's error over each of the other countries. Explain your results based on this plot and the results seen in question 3.

The model fitted over the subset of observations from Israel performed less good over observations from other countries. As we have seen in question 3, the distribution of temperatures in Jordan resembles that of Israel. Therefore, out of the three countries, the model performed best on Jordan.

The distributions of South Africa and The Netherlands were further from those of Israel and therefore the fitted model performed poorly over them. Notice that even though the distribution of observations from The Netherlands has a very similar shape to that of Israel, and that the distribution of observations from South Africa has a very different shape, the model performed better over South Africa. This is probably because on average observations from Israel are closer to those of South Africa. Therefore, even though the model does not capture correctly the distribution of observations from South Africa, the errors are still smaller than in the case of observations from The Netherlands.


Loss Over Countries For Model Fitted Over Israel
loss
Jordan
South Africa
The Netherlands
country
country
Jordan
South Africa
The Netherlands