DATA MINING AND MACHINE LEARNING
TRIMESTER-1 2021
GROUP PROJECT-GROUP 22
MACHINE LEARNING CHALLENGE

DATASET USED: MELBOURNE HOUSING DATASET
SUBMITTED TO: SOMAIYEH MAHMOUDZADEH

Name	***Student ID***	***Contribution***
Navdeep Singh Randhawa	219249449	5: Coding and report writing.
Gurpreet Singh	218701668	5: Report writing and presentation.
Abhishek Kapoor	218590431	5: Report writing and presentation.
Michael Rai	218680443	5: Coding and report writing.
Iftekhar Qureshi	218676618	5: Coding and report writing.
Rawan Awadh K Alharbi	219113465	0

Brief Summary ML Problem Formulation

Conclusions from the last assignment:

We chose the Melbourne housing dataset as it contains a considerable amount of data on which we could work on. Inaccuracies like data duplication, null values for data and invalid data were found in the dataset. Multiple methods were used for cleaning the data, such as dropping out the attributes, linear regression and some cleaning with computational commands.

Categorical Features of Dataset: Type(3:'house','t','u'),

Continuous features of dataset: Rooms (Discrete Values), Suburb/ Postcode, Price, Car, Bathroom, Bedroom, Land size.

Different hypothesis were analysed to get a clear insight of the dataset. Our group arrived to a conclusion in the last assignment that the price is strongly depending on the number of the rooms in the house whereas the price is not strongly dependent on the distance of the house from the city which we analysed and presented in last assignment. Classification and linear regression models both can be used as we have both types of variables present in our data set.

For this assignment, we are building multiple machine learning models to make some predictions. These prediction model will calculate the price of a particular house by using multiple features such as number of rooms etc.

Machine Learning Problem Formulation

- **CLASSIFICATION PROBLEMS:**
    1. Predict the type of the house based on the multiple features, using multiple machine learning algorithms?
- **REGRESSION PROBLEMS:**
    1. Predict the house price based on multiple features of the house?
    2. Prediction of the land size of the house based on multiple features of the given house.

What model to run on the dataset and why?

There are 3 main aspects in machine learning model I.e., Collection and pre-processing of the dataset, EDA, Applying Machine Learning model on the dataset.

Collection and Pre-processing – After the collection of the dataset, pre-processing is one of the main aspects in any Machine Learning model. In pre-processing the dataset is checked whether it is fit and sufficient to use in a Machine learning model. Pre- processing of dataset consists of cleaning the dataset, dealing with the missing and the outlier value of the dataset and much more.

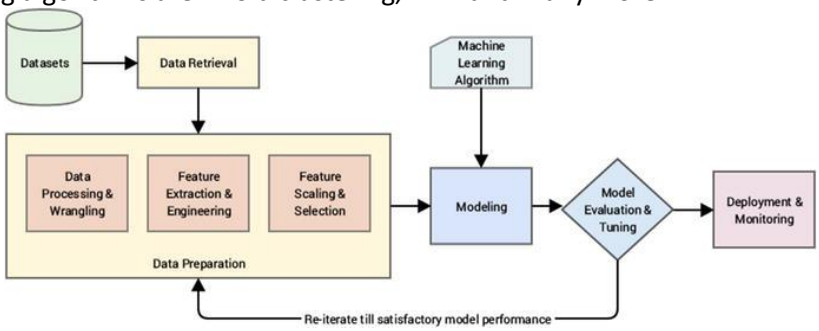
EDA - Exploratory data analysis (EDA) is a part of a machine learning project in which we visualize the dataset from different aspects to notice similarity or fashion which will further help us to implement machine learning models. Machine learning models are implemented based on the results of the EDA.

Applying Machine Learning model - Multiple machine learning models were implemented on the dataset based on the results of the EDA. Linear regression was applied to find some linear relationship between 2 of the different features of the house. Logistic Regression was also used to check the relationship between some of the features. Multivariate regression to search for relationships between more than 2 features of the house such as to find the price of the house based on house land size, distance from the city and more. Multivariate regression is one of the different algorithms where we used more than 1 independent variable to find 1 dependent variable.

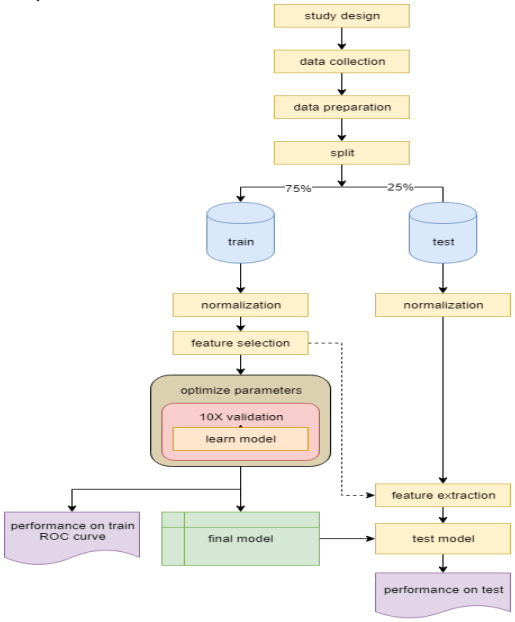
Step by step pictorial depiction (Machine learning flowchart)

Machine learning is a project in which we can make a model to predict a value based on another values. There are mainly 2 different types of machine learning models – Unsupervised, Supervised.

Unsupervised Learning – In Unsupervised Machine Learning model, the model is given unlabelled data (the machine is not told whether the predicted value is correct or incorrect), and the machine is expected to build a model on its own by identifying common patterns in the data. The dataset is split into 2 datasets I.e., train and testing dataset. The machine is trained with the training dataset and is tested with the test dataset to find the accuracy of the model. Some of the common unsupervised machine learning algorithms are K-fold clustering, KNN and many more.



Supervised Learning - In Supervised Machine Learning algorithm, the model is provided with labelled data, I.e., after the machine's prediction, it is told whether the prediction was correct or incorrect. Here for training and testing purposes of the model, the data is split into 2 sets I.e., training data and testing data. Commonly, the size of training data is at least 3 times larger than the testing data, so that the machine is trained properly. Though much larger training data as compared to testing data can also lead to overfitting. After splitting of the dataset, the machine is first trained with the training dataset and then the trained model is tested with the test dataset. By testing the model, we can find the accuracy of the model, which is usually higher than the accuracy of unsupervised machine learning model. Some of the most common Machine Learning algorithms are Linear Regression, Decision tree, Random Forest and much more.



Results and Discussion

TECHNIQUES TO PROCESS DATA TO USE ALGORITHMS:

PCA (Principal Component Analysis)

PCA is a useful tool which is used in multiple machine learning models. The main aim of PCA reduce high dimensional data into low dimension, without compromising much of essential information in the data. This data is generally the number of input variables in training data. There can be thousands of input data that is used in training. We need larger space to store this high dimensional data and spend valuable computational resource to analyse and visualise this data. It also causes poor performance for machine learning algorithms when overfitting occurs.

There are 3 main steps in PCA – Standardization (scaling and centring the data)-> Covariance Matrix Computation -> Compute the Eigen Vectors and Eigen Values of the covariance matrix to identify the Principal Components -> Feature Vector -> Recast the data along the Principal Component Axes.

Hyperparameter Tuning

A Hyperparameter is referred to a parameter whose value is used for the control of the learning process. Hyperparameter Tuning in Machine learning is the process of choosing a set of optimal hyperparameters for a learning algorithm. Hyperparameter tuning works by running single training job in which multiple trials are running. This tuning has a requirement of explicit communication between the AI Platform Training service and the training application.

Label Encoding

Label Encoding in Machine Learning is the process of converting the labels into a numeric form for the purpose of converting it into a form so that the machine can read it. This step is necessary and important for the structured dataset in supervised learning. One of the limitations of this approach is that it assigns a unique number to each class of data which leads to making priority issues in training of data sets.

Normalization

Normalization in machine learning is the process of rescaling numeric attributes with a real value into a range from 0 to 1. It is used to make the training model less sensitive to the scale of features leading in an accurate and better model, making the features more consistent and resulting in more accurate predictions. There are certain ways to normalize the data, first one is by using the 'normalize()' method. The second method is to use 'MinMaxScaler'. One of the advantages of using Normalization is that better execution is guaranteed, and the size of information base is diminished in size.

```
#scale the data using MinMax Scaling
from sklearn.model_selection import train_test_split
Xtrain, Xtest, ytrain, ytest = train_test_split(X_new, y, test_size=0.3, random_state=42)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(Xtrain)
Xtrain = scaler.transform(Xtrain)
Xtest = scaler.transform(Xtest)
```

Chi-Square

Chi-square plays an essential role of selection of features for different machine learning algorithms. For 2 different variables we can get the expected and the observed value. By applying the below formula, we can get the chi-square value. When we try to get the best feature, the feature tends to be highly dependent on another feature.

The Formula for Chi Square Is

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where:

c = degrees of freedom

O = observed value(s)

E = expected value(s)

Mathematically speaking, the chi square indicates the value of dependency of one feature on another feature and vice- versa. So, simply if the chi square value is higher between 2 variables it implies that one of the variables is highly dependent on the another and one of those can be used in the machine learning model to predict the other feature.

```
# the data is tranformed such that only desired features are picked
from sklearn.feature_selection import chi2
from sklearn.feature_selection import SelectKBest
X_new = pd.DataFrame(SelectKBest(chi2, k=6).fit_transform(X, y))
X_new.head()
```

	0	1	2	3	4	5
0	12310.0	0.0	1480000.0	2.5	3067.0	202.0
1	5724.0	0.0	1035000.0	2.5	3067.0	156.0
2	9446.0	0.0	1465000.0	2.5	3067.0	134.0
3	8682.0	0.0	850000.0	2.5	3067.0	94.0
4	10193.0	0.0	1600000.0	2.5	3067.0	120.0

ALGORITHMS USED IN THE PROJECT:

Logistic Regression

Logistic regression is a simple supervised machine learning algorithm which is used as an alternative to simple linear regression. The difference between linear and logistic regression is that the linear regression tries to represent the relationship as a straight line whereas logistic regression tries to represent the relationship as a natural logarithmic function. Logarithmic regression tries to calculate the probability by using the concept of odd ratios. Simply, put it uses the odd of not happening an event against odd of happening the event. Logistic regression than uses the above-mentioned method of odds to predict a logarithmic equation to fit the data.

TRAINING AND TESTING THE MODEL

```
from sklearn.linear_model import LogisticRegression
clf_LR = LogisticRegression()
clf_LR.fit(Xtrain,ytrain)
```

```
LogisticRegression()
```

EVALUATION OF MODEL BUILT:

```
from sklearn.metrics import accuracy_score
print("Training accuracy: {}".format(accuracy_score(ytrain, clf_LR.predict(Xtrain))))
print("Testing accuracy : {}".format(accuracy_score(ytest, clf_LR.predict(Xtest))))
```

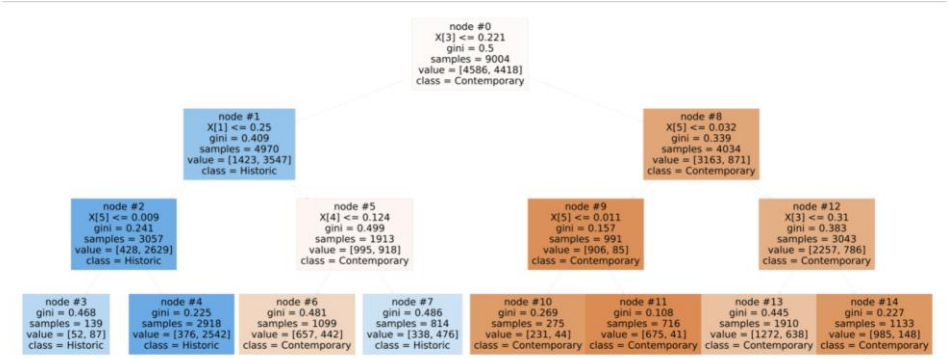
```
Training accuracy: 0.7313416259440249
Testing accuracy : 0.7240932642487047
```

Decision Tree

Decision True algorithm comes under the category of supervised learning algorithm. This algorithm can be used to solve both regression and classification problems. The main goal using this algorithm is the creation of a model which predicts the value of a target variable using the training data provided to it. It uses simple learning decision rules on the training data. The model uses a tree representation model for it's working. For predicting a class label, the start is from the root of a tree. Then for every attribute of the record, comparison is done with the root of the tree and based on the result of comparison, the branch corresponding to that value is followed.

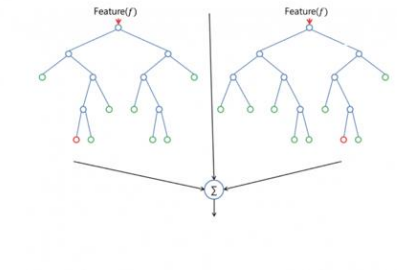
Decision Tree makes use of different algorithms for decision of splitting a node into multiple sub-nodes. The purity of a node increases with respect to the target variable. The decision tree splits the

nodes on all available variables and then on basis of which split results in most homogenous sub-nodes, it selects them.



Random Forest

Random forest is one of the easy-to-use supervised machine learning algorithms, which produces outstanding result, even without hyperparameter tuning. Here forest refers to the group of multiple decision trees. The basic approach of random forest is to implement multiple decision trees and consolidate the outcome to get a more accurate prediction. Random forest is applicable on both classification and regression problems. Below is a simple example of a random forest using 2 decision trees.



RANDOM FOREST REGRESSOR

```
# determine which model to use
from sklearn.ensemble import RandomForestRegressor
RF = RandomForestRegressor(criterion = 'mse', n_jobs = 1, n_estimators = 100,  min_samples_leaf
RF.fit(Xtrain, ytrain)
y_pred = RF.predict(Xtest)

# Display.
print('Random Forest Regressor')
print('\nR-squared training set:')
print(RF.score(Xtrain, ytrain))

print('\nR-squared test set:')
print(RF.score(Xtest, ytest))
importances = pd.DataFrame({'feature':Xtrain.columns,'importance':np.round(RF.feature_importanc
importances = importances.sort_values('importance',ascending=False).set_index('feature')
importances
```

```
Random Forest Regressor

R-squared training set:
0.9711022532948411

R-squared test set:
0.7866183552030603
```

Multiple Regression

Linear Regression is one of the most used and one of the basic supervised machine learning algorithms on which multiple other algorithms are based on. This algorithm is mostly used to predict the continuous feature of the data rather than to predict the categorical feature. Linear regression uses the simple line equation I.e., $y = mx + c$, where y is the value to be predicted (dependent variable), x is the feature which is used to predict (independent variable), m is the slope of the line(gradient), and c is any other constant which might affect the value of the dependent variable but does not merely affect the fashion of the line. The same algorithms are attributed as multiple when the dependent variable depends on multiple factors.

There are certain advantages of using Multiple Regression. Using this regression, we can determine the relative influence of one or more predictor variables to the criterion value.

The Disadvantages of using this Regression is that these techniques are complex and requires high level calculation.

Training and Testing ML model

```
from sklearn.linear_model import LinearRegression
my_model = LinearRegression()
#fit the model using our data
my_model.fit(X_train_norm, ytrain)
```

```
LinearRegression()
```

```
ypredicts = my_model.predict(X_test_norm)
print("The predicted sales:")
print(ypredicts)
```

```
The predicted sales:
[1298620.51078909 1443673.75765347  715804.02728344 ... 1214502.37421448
  532520.82054878  854341.436663  ]
```

EVALUATION OF THE MODEL

```
from sklearn import metrics
print('MAE:',metrics.mean_absolute_error(ytest,ypredicts))
print('MSE:',metrics.mean_squared_error(ytest,ypredicts))
print('RMSE:',np.sqrt(metrics.mean_squared_error(ytest,ypredicts)))
print('R^2 =',metrics.explained_variance_score(ytest,ypredicts))
```

```
MAE: 253588.513659183
MSE: 115356608949.59808
RMSE: 339641.8833854242
R^2 = 0.5669466028126404
```

Gradient Boosting

Gradient Boosting is a technique which is used for regression and classification problems. Gradient Boosting uses gradient descent to minimize the loss. It comes under machine learning techniques. Gradient Boosting is used to produce a predictive model from a collection of weak predictive models. Gradient boosting allows a user to optimise a user specified cost function, instead of using a function which results in less control.

GRADIENT BOOST REGRESSOR

```
from sklearn import ensemble
params = {'n_estimators': 100,
          'max_depth': 10,
          'learning_rate': 0.01,
          'loss': 'ls'}
reg = ensemble.GradientBoostingRegressor(**params)
y_pred = reg.fit(Xtrain, ytrain)
from sklearn.metrics import r2_score
print("Testing accuracy :" , (reg.score(Xtest,ytest).round(5))*100 ,"%")
print("Training accuracy :", (reg.score(Xtrain,ytrain).round(5))*100, "%")
```

```
Testing accuracy : 66.63799999999999 %
Training accuracy : 77.45 %
```

ANN Regression

The idea of Artificial Neural Network (ANN) is derived from the biological network of the brain where billions of neurons are interconnected. Brain neurons learn and generalise information from other neurons and then transmit that information to other nerve cells in the body.

There are several types of ANN models and the one we focused on is the feedforward neural network (information travels only in one direction). The entire process for prediction using this ANN model is-

- There is a mandatory input layer of neurons and output layer of neurons, and several hidden layers in between them.
- Hidden layers process the data for the output layer to predict the object identity.
- Number of neurons assigned to the layers is depended on the dataset involved.
- A weight is assigned to connections between each neuron. Initially, the weight is a random number which changes as the model keeps on training to adjust and find the right weights.
- An activation function is present which depicts the internal state of a neuron. The function the input signal of a node to an output signal.

Support Vector Machine

Support Vector Machine (SVM) can be used to address classification problems. The aim of SVM is to return a best fit hyperplane that can divide our dataset into distinct classes. Since, SVM is used

mostly in binary classification, it divides the data into two classes- one on the negative side of the hyperplane and another on the positive side. Multiclass classification is possible using SVM which is generally tedious.
The data is simple to classify when the data is linearly separable. In case of non-linearly separable data, we cannot simply draw a straight line and get two distinct classes. Instead, the non-linear data can be converted to linearly separable data in higher dimensions. From higher dimension, we can transfer back to original dimension using mathematical transformation. This transformation is difficult to do, and we have SVM "kernels" to do the job for us.

EVALUATION OF MODEL:

R^2 – Regression Score Function

R^2 score is used to evaluate the performance of a linear regression model.
The best possible score of this function is 1.0 and the value can be negative as well depending on the model if it is arbitrarily worse. Lower the value, lower the correlation.
The mathematical formula of R^2 score is

$$R^2 = 1 - S_{res} / S_{tot}$$

S_{res} is the sum of squares of all the residual errors

S_{tot} is the total sum of the errors

If the R^2 score is 0.70, it means that the 70% of the changeability of the output attribute which is dependent is explainable by the model and the rest 30% is not.

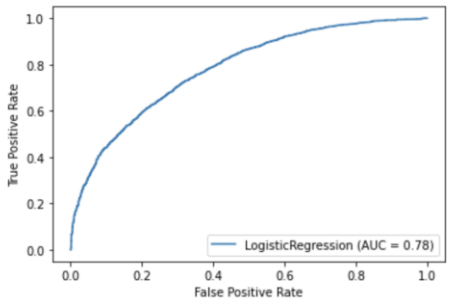
ROC Curve

ROC curve is a plot of the false positive rate versus the true positive rate .
The true positive rate is calculated by dividing the number of true positives by the sum of number of true positives and false negatives.
True Positive Rate = True Positives / (True Positives + False Negatives)
The false positive rate is calculated by dividing the number of false positives by the sum of number of true positives and false negatives.
False Positive Rate = False Positives / (False Positives + True Negatives)
Uses of ROC Curve:

- The area under the curve depicts the summary of the model skill.
- It can be also used for comparison of different curves in regard to different thresholds.

```
from sklearn import metrics
metrics.plot_roc_curve(clf_LR_PCA, Xtest, ytest)
```

```
<sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7fa376425430>
```

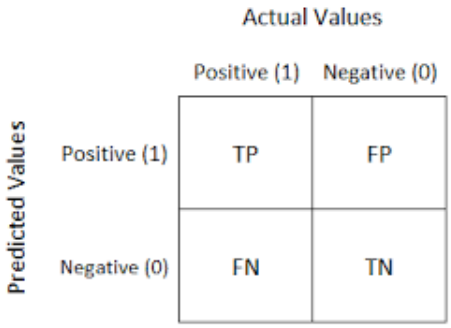


K-Fold Cross-Validation

The K-Fold Cross-Validation method is a standard for the estimation of the performance of a machine learning algorithm. The procedure works by dividing a limited dataset into k number of groups. This procedure is a popular one because of its simplicity and its less biased results and less optimistic results.

Confusion Matrix

Confusion matrix, also known as error matrix, which helps in visualization of algorithm's performance. It is used to describe the performance of a supervised classification machine learning model. Confusion matrix is *n*n* table in which each row of the table represents the occurrences of the actual class(feature) whereas each column represents the occurrences of the predicted class. From Confusion matrix we can describe some of the terms such as True positive rate, false positive rate and much more which helps us further in the analysis of the machine learning model.



Conclusions

Achievement by using ML algorithm: In our dataset, I.e., the Melbourne housing snapshot, after cleaning and pre-processing the dataset, we implemented multiple Supervised Machine Learning algorithms such as Linear Regression, Random Forest, SVM and many more so that we can build a model by which we can predict a feature of a particular house, given the other required features are provided. Our main aim of the assignment was to predict the features such as house price, house land size and house type. Different machine learning models gave us different accuracies when implemented on the dataset. But we were able to achieve a model by which we can achieve accuracy more than 80%, which is good enough to use the model in real life scenarios. We achieved that model by implementing multiple Machine Learning concepts such as PCA, hyperparameter tuning, feature engineering and much more. Using different Machine Learning models also gave us the flexibility to choose a better model over another which in turn gave us better results.

Suggestions: There are numerous ways by which the model can be improved, and we can get better results. The first and foremost method by which we can achieve better accuracy is by the amount of data. More data can lead to more training and testing data by which the machine will have more information to learn from and can achieve higher accuracies. Secondly, better accuracy can be achieved if the data is clean. More data to the dataset is an essential part but it will be of no use if half of the features are missing and some of them has outliers in it. Missing and outlier values affect the Machine learning model in a dramatic way and should be treated properly.

Section 4: References

- Zhao, Z. and Liu, H., 2007, June. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning* (pp. 1151-1157).
- Sutton, R.S. and Barto, A.G., 2018. *Reinforcement learning: An introduction*. MIT press.
- Satorra, A. and Bentler, P.M., 2001. A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika*, *66*(4), pp.507-514.
- Ash, A. and Shwartz, M., 1999. R2: a useful measure of model performance when predicting a dichotomous outcome. *Statistics in medicine*, *18*(4), pp.375-384.
- Hanley, J.A. and McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), pp.29-36.
- Rodriguez, J.D., Perez, A. and Lozano, J.A., 2009. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, *32*(3), pp.569-575.
- Worster, A., Fan, J. and Ismaila, A., 2007. Understanding linear and logistic regression analyses. *Canadian Journal of Emergency Medicine*, *9*(2), pp.111-113.
- Shlens, J., 2014. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.
- Binder, J.J., 1985. On the use of the multivariate regression model in event studies. *Journal of Accounting Research*, pp.370-383.
- Zou, J., Han, Y. and So, S.S., 2008. Overview of artificial neural networks. *Artificial Neural Networks*, pp.14-22.
- Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A. and Brown, S.D., 2004. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *18*(6), pp.275-285.
- Friedman, J.H., 2002. Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), pp.367-378.
- Yogatama, D. and Mann, G., 2014, April. Efficient transfer learning method for automatic hyperparameter tuning. In *Artificial intelligence and statistics* (pp. 1077-1085). PMLR.
- Noble, W.S., 2006. What is a support vector machine?. *Nature biotechnology*, *24*(12), pp.1565-1567.
- Educative: Interactive Courses for Software Developers. n.d. *Data normalization in Python*. [online] Available at: <https://www.educative.io/edpresso/data-normalization-in-python> [Accessed 23 May 2021].
- Yadav, D., 2019. *Categorical encoding using Label-Encoding and One-Hot-Encoder*. [online] Medium. Available at: <https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd> [Accessed 23 May 2021].
- ES, S., 2021. *Hyperparameter Tuning in Python: a Complete Guide 2021 - neptune.ai*. [online] neptune.ai. Available at: <https://neptune.ai/blog/hyperparameter-tuning-in-python-a-complete-guide-2020> [Accessed 23 May 2021].