|  |  |  |  |
| --- | --- | --- | --- |
| Data Science with Python Assignment 2 Report  |  | | --- | | I certify that this is all my own original work. If I took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in my submission. I will show we I agree to this honor code by typing “Yes” |   **Title: Class Feature prediction of Mice Protein Expression Data Set**  **Author Information:** Shubhankar Jahagirdar – S3793593 Masters of Data Science RMIT University.  **Contact Details:** Phone: 0484260598 || Email: [jahagirdar.shubhankar@gmail.com](mailto:jahagirdar.shubhankar@gmail.com) | | |
| **Table of Contents**   1. Abstract 2. Introduction 3. Methodology    1. Data Preparations    2. Data Exploration    3. Data Modeling Strategy.    4. Model building and hyper-parameter tuning.    5. Model comparison 4. Results 5. Discussions 6. Conclusion 7. References |

**Abstract**

In this report we consider a data set from UCI Machine Learning Repository on Mice Protein Expression and address the target feature the class feature. The data science process provides a life cycle to structure the development of data science projects. The first three steps of the Data Science process namely data preparation, data exploration and data modeling are performed on the Mice Protein Expression data set and predict the class i.e. the target features. Classification models K-nearest neighbor and Decision Trees are implemented in this data set and the accuracy score of predicting the unseen data is calculated. The results show that KNN algorithm is 19% more accurate than Decision Tree for identifying the subsets of protein that are discriminant between the classes.

**Introduction**

Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. (https://en.wikipedia.org/wiki/Data\_science) A real world data set is extracted from the UCI repositories and used to demonstrate the Data science lifecycle steps. The data set used for this report is a Mice Protein expressions data set containing 1080 observations and 83 features. The features are expression levels of 77 proteins measured in the cerebral cortex of 8 classes of control and Down syndrome mice exposed to context fear conditioning, a task used to assess associative learning. (Clara Higuera Department of Software Engineering and Artificial Intelligence, Faculty of Informatics and the Department of Biochemistry and Molecular Biology, Faculty of Chemistry, University Complutense, Madrid, Spain., 2015) This report focuses on the data extraction, exploration and modeling techniques of the data science lifecycle. Two types of modeling techniques can be used on the data namely classification and clustering. In this report we have used classification models K-nearest neighbor and Decision trees.

The acquired data can be divided into two parts one containing the protein information and other having the categorical data leading towards the class feature. The target feature is divided into 8 classes which is described based on the features such as Genotype, behavior and treatment (Clara Higuera Department of Software Engineering and Artificial Intelligence, Faculty of Informatics and the Department of Biochemistry and Molecular Biology, Faculty of Chemistry, University Complutense, Madrid, Spain., 2015) The aim of this data is to identify the proteins that are discriminant between the classes. The data set contains dependent features whose direct inference can be used to predict the class feature, however for the modeling purposes all the protein information has been considered to predict the class features.

**Methodology**

We aim to introduce a classification model to predict the class feature. In order to classify the provided data, we need to prepare, explore and extract the most important features from the data so as to prepare and tune the models. To achieve the same we have used python programming language which helps us by providing interactive libraries and graphical representation methodologies to succeed in our ambition.

Data Preparation, Data exploration, Data modeling Strategy, Model building and hyperparameter tuning and Model comparison has been implemented and discussed below.

**Data Preparation**

Data preparation is the act of manipulating raw data into a form that can readily and accurately be analyzed (https://en.wikipedia.org/wiki/Data\_science) Data preparation plays a vital role in predicting the target as the raw data can contain irregular cardinalities, outliers or null values which can affect the model evaluation strategy and hamper model performance.

In this data set we observe that the class target feature is a multivariate having 8 different target levels. It can be observed that there are 77 numeric features, 4 categorical features and 1 Mouse\_ID feature present in the data set.

The classes of the Target feature are based on different features like Genotype, behavior and treatment. According to Genotype the mice can be control or trisomic. According to behavior, some mice have been stimulated to learn(context-shock) where as some are not while according to treatment, some mice are treated with drug like memantine while others are just to saline.

The mouse ID feature in the data set contains both the ID of the mouse and the measurement number. Since two different features are represented in the same column these features are separated and Mouse ID feature is dropped or deleted from the data set as ID like columns are used for indexing and representation purposes and are of no use in exploration or modeling the data.

Taking the data preparation process to the next step we observe that there are some missing values that are found in the data set, interestingly we observe that the missing values are from the numerical data in the data set and the categorical data do not show any missing values. Hence we use python function to impute the missing values by the mean values of the particular numerical data.

Since we have the prepared data without any irregular anomalies or missing values, we can proceed towards exploring the interesting facts and trends of the data.

**Data Exploration**

In this section we have explored some of the features of the data. Interactive python features are used to study the data and plot graphs to explore data trends. Univariate and bivarate data explorations are performed on the data to analyze the trends. Univariate data exploration is the method in which a single feature is explored where as bivarate data exploration allows us to explore pairs of data and establish relationship between them.

Various data trends such as the skewness of a particular feature, the median and the range of values, the outliers and quantiles can be plotted and explored. Ten univariate features were explored and the findings are discussed below:

**Protein BDNF\_N:** The values of the protein were observed to be normalized with values ranging from 0.10 to 0.50.

**Protein NR1\_N:** The values of this protein were normalized with values having some observations at 0.5 while the maximum values range from 1.2 to 3.7 having maximum frequency around 360.

**Protein NR2A\_N:** The values of this protein were normalized with values ranging from 1 to 8 having maximum frequency around 350.

**Protein DYRK1A\_N**: The values of this protein were right skewed with values ranging from less than 0.5 to around 1.0 having maximum frequency around 575.

**Protein** **ITSN1\_N:** The values of this protein were right skewed with values ranging from less than 0.5 to around 2.0 having maximum frequency around 550.

**Protein** **BAD\_N:** The values are observed to be normally distributed having range around 0.10 to around 0.25.

**Protein** **BCL2\_N:** The values of this protein are observed to be highly right skewed having range of values from below 0.10 to 0.40

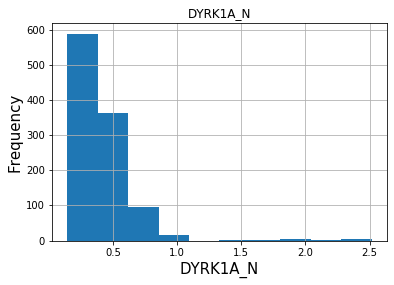
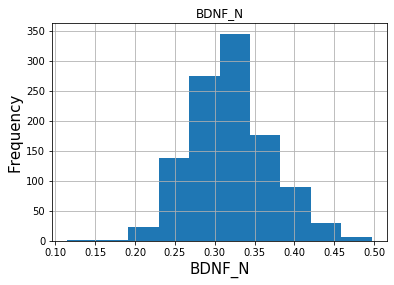
**Protein** **H3MeK4**\_**N**: The values of this protein are observed to be highly right skewed having range of values from below 0.10 to 0.40 and having median at around 0.22.

**Protein** **H3AcK18**\_**N**: The values range from below 0.10 to more than 0.35, however, some outliers are found in this feature.

**Protein** **EGR1**\_**N**: This protein defines a range of values from 0.10 to 0.35, however, some outliers are noted above 0.40 and median values is noted to be around 0.20

The following figures represent the univariate data plots protein DYRK1A\_N and BDNF\_N

*Figure 1:* ***DYRK1A\_N***  *Figure 2:* ***BDNF\_N***

Bivariate exploration of data allows us to build relationship between the features of the data set. After exploring 10 pairs of features we found some features showing linear dependency on each other whereas some features were completely independent and showed no possible hypothesis. The following hypotheses were explored on each pair of columns.

**Genotype and Protein (DYRK1A\_N): Plausible Hypothesis :-** It can be concluded from the data that the genotype control lies in the range 0.1 to 0.7 having median value 0.4 whereas the genotype trisomic lies in the range 0.1 to 0.8 having median 0.45 of the protein value DYRK1A\_N, however, it can be said that though control genotype have few outliers above 0.5 if protein DYRK1A\_N no trisomic genotype represents more than 0.1 of the protein DYRK1A\_N

**Treatment and Protein (pP70S6\_N) :- Plausible Hypothesis:-** It can be observed that the mice affected with memantine have values in the range 0.1 to 0.7 where as the Saline mice have values in the range 0.1 to 0.8, but most of the values of the mice being saline are in the range 0.1 to 0.5 having median value around 0.4.

**Behavior and protein(ADARB1\_N) :- Plausible Hypothesis:-** It can be observed from the following Boxen plots that there is significant difference between the range of values of behavior with respect to the protein(ADARB1\_N). Although the median values context shock and non context shock are almost similar the range of context shock lies from 1.0 to 1.50 where as the range of not context shock lies from 1.0 to 1.25.

**protein(pPKCG\_N) and protein(CDK5\_N) :- Plausible Hypothesis:-** It can be observed that all the values of protein(pPKCG\_N) lies below the 0.4 of protien(CDK5\_N), however, one exception can be observed.

**Protein(CREB\_N) and protein(CAMKII\_N):- Plausible Hypothesis:-** It can be that the values of protein(CAMKII\_N) increases linearly with the value of protein(CREB\_N).

**Protein(GluR3\_N) and protein(GluR4\_N):- Plausible Hypothesis:-** It can be concluded that there is a visible linear increase in the value of protien GluR3\_N with the increase in the value of protein GluR4\_N having some 2-3 exceptional cases. However, the bandwidth of increase is narrow and is in the range below 0.1 to not more than 0.2.

**Treatment type and Protein(RAPTOR\_N):- Plausible Hypothesis:-** It can be observed in the following graph that both Memantine and Saline treatment of RAPTOR\_N protein lies in almost similar range values ranging from 0.1 to 0.45

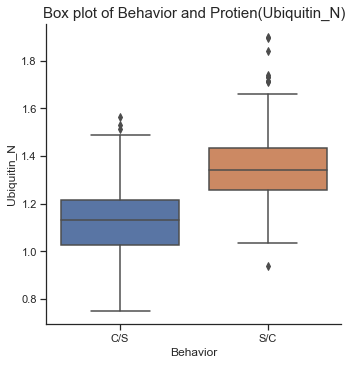
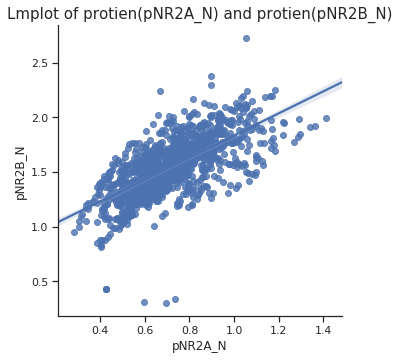
**Genotype and Protein(Tau\_N):- Plausible Hypothesis:-** It can be observed in the below graph that the median values of both the Genotypes Control and trisonic lies slightly below and above 0.2 respectively, however, there are no triscot values above 0.5 Tau\_N.

**Behavior and protein(Ubiquitin\_N):- Plausible Hypothesis:-**  It can be observed in the following box plot that there exists a vast difference in the range of values of context shock and non context shock of behavior with respect to the protein values of(Ubiquitin\_N).

**Protein(pNR2A\_N) and Protein(pNR2B\_N):- Plausible Hypothesis:-**  It can be observed in the below graph that the value of the protein pNR2A\_N linearly increases with the value of pNR2B\_N within range slightly below 1.0 to slightly below 2.5.

The following graphs can be observed for reference purposes

*Figure 3: Behavior and Protein*  *Figure 4: protein(pNR2A\_N) and protein(pNR2B\_N)*

**Data Modeling Strategy**

Data modeling in a Data Science process allows us to train a Machine learning algorithm and predict the labels of the class variable and validate it on the holdout data. (Koehsren, 2018) Prior to modeling feature selection and splitting the data into training and testing is performed. Feature selection algorithms are used to extract the most important features and they are ranked according to their preferences. Once we have the important features, we split the data into training and testing data so that the model can be trained and validated on sample data and tested on unseen data so that we can observe the performance of the model and make comparisons and decisions.

Since the target feature is a derivative feature of the three categorical features namely Genotype, Treatment and Behvior, but in this report we aim to predict the class feature based on the protein features in the mice, we drop the categorical features for modelling.

**Feature Scaling and Standartization**

As we have noted after exploring the data that some features are skewed as well are some are normalized. To standardize the features we need to represent the data in a particular range. To achieve this we scale the data using min-max scaling. Since we have numerical data, mix-max scaler works best on this type of data as the scaler maps all the observations in the range 0 and 1. Python library sklearn.preprocessing is used in scaling the data. The scaled data is stored separtely to use for further operations.

**Feature Engineering**

Feature engineering defines the use of the data knowledge to extract the important features from the raw data. These extracted features can be used to improve performance of machine learning algorithms. For this data we use Random Forest Importance(RFI) method to feed the extracted most important features to the machine learning algorithms. The RFI feature extraction method is chosen for this data as it is the most popular method due to the relatively good accuracy, robustness and ease of use. Random Forest Importance uses the ensembling technique in which the number of estimators are defined as 100. The number of estimators is a parameter used in RFI to control the number of trees used in the process.

We find the top 10 features in the decreasing order of their importance which are as follows : ‘SOD1\_N', 'pPKCG\_N', 'pERK\_N', 'APP\_N', 'pCAMKII\_N', 'Ubiquitin\_N', 'CaNA\_N', 'DYRK1A\_N', 'ITSN1\_N', ARC\_N'

Once we have the important features we can use these features in training our machine learning models. However to tune the models, we vary the number of features and use RFI as a step in the pipelining process we use to fine tune our machine learning models.

**Test Train Split**

On the given data set to predict the accuracy of the models, we train the models and then test them on the unseen data. In order to do so, we divide our data set in 70/30 portion having training and testing data respectively. The classification algorithms are then used on the training data, their hyper-parameter tuning is performed and then we use the testing data to predict the accuracy.

The test train split for this report is performed by model selection(sklearn.model\_selection) library predefined in the python language. We obtain 756 observations for training and 324 for testing purposes.

**K-Fold Cross Validation**

Cross validation technique is used to reduce over fitting in the data. K-Fold cross validation is a technique in which K represents to the number of groups the given data sample is split into. For this data value of K is set to 5. It is popular method at it generally results in less biased or less optimistic estimate of the model skill. This strategy is passed as a method in the exhaustive search methodology which later will be implemented on the Machine learning models.

**Classification Model 1: K-Nearest-Neighbour Algorithm**

KNN is a supervised classification algorithm that predicts the output labels based on the k number of closest observations. The protein features from the data set are provided as an input the KNN algorithm and the accuracy of the model on the training data is observed. Two types of distance metrics Manhattan and Euclidean are used by KNN algorithm to calculate the difference between the values of the observations. The Euclidean distance is the shortest distance between two points, whereas, Manhattan distance is the right angled difference, these distance metrics along with the number of nearest neighbours form the parameters on which KNN algorithm can be tuned.

Pipeline function along with GridSearch from the sklearn library has been implemented in this project to maximise the performance of the models. Pipeline provides us with a function that arranges the sequence in which models can be used as we need to extract the important features using the RFI ensembling technique and use those feature parameters on the KNN model.

The top 10,20 and all 77 features are provided as an input to the KNN model with an array of 5 neighbour values [1, 5, 10, 15, 20] and the value of P [1,2] i.e Manhattan as well as Euclidean.

GridSearch method is used which performs an exhaustive search using the model parameters and the extracted number of top features using the 5 fold cross validation and records the best parameters and the highest accuracy score obtained. The fit() method allows us to pass the training data and the training target upon which the model will be evaluated.

Post hyper-parameter tuning, the models best accuracy score was noted to be **0.99**. The best parameters we noted as **Number of neighbours = 1**; **Number of features = 77** and the **value of P = 1;**

**Classification Model 2: Decision Tree Algorithm**

A Decision Tree is a supervised predictive model that can learn to predict discrete or continuous outputs by answering a set of simple questions based on the values of the input features it receives. A pipeline function is implemented with RFI feature selection algorithm and the decision tree classifier. The hyper-parameters of decision tree include the maximum depth of the tree, the minimum number of sample splits and the criterion of the split. The Decision tree classifier, the extracted top 10,20 and all 77 features are provided with array of maximum depth values : [6,8,10], criterion :[Gini, Entropy] and minimum number of sample splits:[2,5,10] .

GridSearch method is used which performs an exhaustive search using the model parameters and the extracted number of top features using the 5 fold cross validation and records the best parameters and the highest accuracy score obtained. The fit() method allows us to pass the training data and the training target upon which the model will be evaluated.

Post hyper-parameter tuning, the models best accuracy score was noted to be **0.85**. The best parameters were recorded as **split criterion : Entropy**, **Maximum depth: 1**, **Minimum Sample Split: 2** and on **top 20 features.**

**Testing Models**

The well-tuned models have an optimal accuracy score which permits us to try them on the test or unseen data. We have used 756 observations of the training data on the cross validated framework. The best parameters we noted for each of the classification models. Since we have obtained the tuned models, we move forward to the next step and test these models on the unseen data and evaluate their respective performance accuracy.

To validate the models and reduce over-fitting 5-fold cross validation technique is used using the cross\_val\_score method from the sklearn.model\_selection package. The predict() function takes the testing data and returns the learned of the new unseen data provided. The prediction score for both the models KNN and Decision tree is noted and it is found that KNN model labels the data with an accuracy score of 0.9320 where as Decision tree implementation yields score of 0.7158.

**Model Comparison**

From the mice protein expression data set the numerical features i.e the protein information was considered after splitting it into 70% training and 30% testing data. On a cross validated platform it was observed that KNN outperforms the Decision Tree classifier on the training data having an accuracy score of 0.99 where as Decision tree score was noted to be 0.85.

These models were then used on the 30% unseen data and it was found that KNN predicts the labels with an accuracy score of 0.9320 and Decision tree’s optimal score was 0.715.

Once we obtain the highest cross validated accuracy scores of both the models, we compare the difference between the performance of the models on with combinations of hyper parameter values. The following tables are obtained for KNN as well as DT.

Table 1: : KNN top parameters and scores

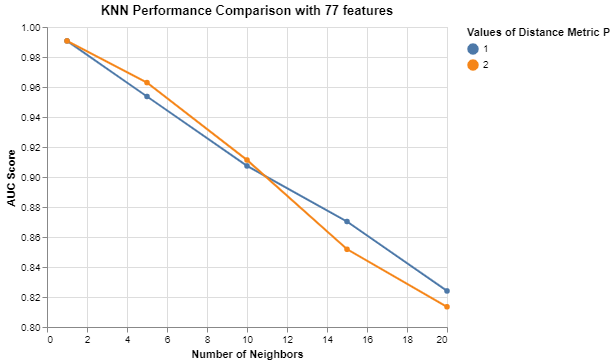
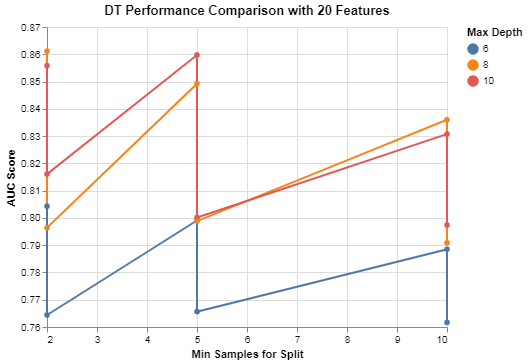
|  |  |  |  |
| --- | --- | --- | --- |
| **Mean Score** | **N\_neighbors** | **KNN\_P** | **RFI\_no\_Features** |
| 0.990728 | 1 | 1 | 77 |
| 0.990728 | 1 | 2 | 77 |
| 0.989404 | 1 | 1 | 20 |
| 0.985448 | 1 | 2 | 20 |
| 0.962940 | 5 | 2 | 77 |

Table 2: DT top parameters and scores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mean Score** | **Criterion** | **Max\_depth** | **Minmum Sample Split** | **RFI\_no\_Features** |
| 0.854505 | Entropy | 1 | 2 | 20 |
| 0.853181 | Entropy | 2 | 5 | 20 |
| 0.850558 | Entropy | 1 | 2 | 20 |
| 0.847891 | Entropy | 2 | 5 | 20 |
| 0.838663 | Entropy | 2 | 2 | 77 |

It can be observed from the above tables that the performance scores of both the models differ by a significant amount of accuracy levels. Once we have the data of both the models, we plot this data using an interactive altair graph to observe the model performances. The following figures represents the performances of the respective models.

Figure 5 : KNN performance with 77 features Figure 6: DT performance with 20 features

Since maximum highest performance parameters are obtained in KNN are from 77 features and DT are of 20 features we plot the graph differences between these two algorithms. For KNN algorithm the distance metric P was used with accuracy score and Number of neighbours on the axes, where as for DT the Maximum depth parameters were used to plot the accuracy score with Min samples split on the axes.

**Results:**

The following observations were made after the evaluating both the classification models on 70/30 portion train test split divided cross validated data.

* On the training model KNN performed with an accuracy score of 0.99 with best hyper-parameters having p = 1, number of neighbors = 1, number of features = 77.
* On the training data Decision Tree algorithm performed with an accuracy score of 0.85 with best hyper-parameters split criterion : Entropy, Maximum depth: 1, Minimum Sample Split: 2 and on top 20 features.
* On the testing data the KNN algorithm performed with an accuracy of 0.932 where as Decision tree performs 0.7158.

**Discussions:**

Since the KNN algorithm works completely on the numerical data, even if we have categorical data we need to integer encode the data before providing as an input to the KNN model. Hence it can be observed that KNN performs significantly well when it comes to numerical data.

Since we dropped all the categorical columns and provided all the numerical protein expressions columns as an input to both the models we observe that KNN algorithm outperforms decision tree on training as well as testing data.

Decision tree algorithm works by selecting a feature and categorizing the values based on the series of questions, decision tree performance on both the training and testing data can be observed significantly low as compared to KNN.

**Conclusion**

Based on the above data and the observed performance scores it can be concluded for the mice protein expressions data considering all the protein information and on the cross validated platform, KNN classification algorithm with p=1, number of features = 77 and number of neighbours =1 can be used to classify the unseen data. It renders the maximum performance and highest accuracy score as compared to the decision tree model.

**References**

Clara Higuera Department of Software Engineering and Artificial Intelligence, Faculty of Informatics and the Department of Biochemistry and Molecular Biology, Faculty of Chemistry, University Complutense, Madrid, Spain. (2015, 08 04). *Mice Protein Expression*. Retrieved from https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression: https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression

https://en.wikipedia.org/wiki/Data\_science. (n.d.). *wiki/Data\_science*. Retrieved from wikipedia.