Software Architectural Styles

Sai Tejini Varma Sangaraju, Divya Sangaraju

Abstract: The software architectural styles are employed in all the software projects as they tell us how the entire code of the business is being organized on a high level. Based on the different architectural styles that can be incorporated in a project, this paper presents patterns in the educational qualifications of developers that have used these styles in various projects using different machine learning algorithms. Software Architectural Styles dataset is considered for analysis and Logistic Regression, Decision tree classifier and KNN classifier algorithms are applied on the data to predict the last degree and job experience of the developer and the results were compared. It is observed that all the algorithms did not produce expected results indicating that the educational details of the developer are independent of architectural styles they have used in the projects. Logistic regression algorithm generated better results compared to Decision tree classifier and KNN classifier. The accuracy for predicting the last degree is 0.775 and for predicting the job experience is 0.46.

1. Introduction and Statement of the Problem

The software architectures are the blueprints of how a software system is being built on a high level. First and foremost, before beginning any project in the real world, we need a design on how everything works. So, to give the outline of the project these architectural styles are used, and this entire structure is called as software architecture. When designing a business process first we need to understand the architectural styles that are there, and we need to choose a suitable style that will suite our project. This gives us an overview of how a project works in general and what are the relations between each layer.

While developing a software system first, the software architecture expert in the office will take a note of the components that are required, what are connections between each component, the data flow and all the other details in depth and then he will provide the outline of the project in the company. Now for designing the outline of the system there are several ways to do it and each method is called as architectural style. While designing a business process first we need to understand the architectural styles that are there, and we need to choose a suitable style that will suite our project. Each style consists of information about the components, connectors, and it will help us to understand the overall system properties.

Therefore, considering things such as providing an overall overview of the entire design and development process and assisting new project participants by giving guidelines on how everything works with quality, architectural styles are important in building any software system. Now in our dataset there are several architecture styles like the Layered, client server and Event driven etc. which were used to predict the developer’s education qualifications. Now knowing all these about the architecture styles we took our dataset that is related to architecture styles and did analysis for it and showed the results in the paper.

In this modern era as the technologies are growing rapidly data has become the key component of any business and any company that utilizes its data or converts the data, they have into some useful information will be in the competition. To get some meaningful data, machine learning is widely used across all the industries. The models that are trained through machine learning will learn the pattern as the humans do and give the necessary outputs.

So, machine learning will give us some useful information from the raw data that is given to the model, that is when the data is given to the model it analyses the data and then find a pattern in that and will give appropriate output. In this first we need to train the model with some input data and check the results and later we must test the trained model with the test data to check if the desired output is obtained.

If any corrections are needed in the future, we can train the model accordingly and the model will learn that, and it will include them in future decisions. There are several machine learning processes in real world and we need to choose those that best fits for our dataset and should train the model. In our project we took Logistic regression, decision tree classifier and KNN classifier machine learning models to train the model and compared the outputs of all the three models.

In most of the previous works related to software architectural styles, work was carried out on the areas of predicting different styles that can be used in the project. But in this paper the patterns in the educational qualifications like the last degree and the job experience of the developers is predicted using the architectural styles the developers used previously for various projects. These patterns are identified by training the model with various machine learning algorithms.

2. Limitations of Study

In this paper, patterns in the educational qualifications of developers are predicted. Last degree and the job experience of the developer are predicted based on different architectural styles they have used for various projects. The dataset that was considered for analysis has attributes like Timestamp, Your Good Name?, Organization?, Last Degree?, Job Experience ?, How Many Repository Architectural Styles have you used for a particular project?. There are 18 such columns in this dataset that discusses various architectural styles and indicate how many times the developer employed each one while working on the projects.

This dataset has high level overview on the architectural styles. It only provided the information regarding how many times a particular architectural style was used by developers in projects. This data was not sufficient to perform detailed analysis to find out the correlation between the educational background of the developer and the architectural styles they have used. If more information was present in the dataset regarding why or on what basis a developer has chosen a particular architectural style, then carrying out detailed analysis to find pattern in the educational qualifications of the developers would have been possible.

Another interesting pattern that could be worked on is predicting the possible architectural styles that were be employed by the developers based on their educational background. For example, we can predict the if developers with computer science background used complex architectural styles while working on projects. But as the dataset that is chosen for performing the analysis does not contain more relevant information it has become difficult to carry out analysis in finding out such patterns. So, in future, work can be carried out in such areas by considering more suitable dataset.

3. Methodology

Patterns in the last degree and job experience of the developers are predicted using various architectural styles that were employed by them in different projects. We have considered Software Architectural Styles dataset from Kaggle website to carry out our analysis. Initially we have performed exploratory data analysis to know about the different attributes that are present in the dataset and then performed data cleaning to remove all the null and duplicate values. We have then used three machine learning algorithms to predict the educational details of the developers. Logistic regression, Decision tree classifier and KNN classifier are used for training the model and then the results are compared to know which model produced better results in finding out these patterns.

The attributes on our dataset are "Timestamp", "Your Good Name?" which tells us the name of the developer, "Organization?" attribute provides the developer's educational institution information, "Last Degree?" attribute presents information about the developer's degree, "Job Experience?" tells us in which industry the developer is working and "How Many Repository Architectural Styles have you used for a particular project?" gives the number of times a developer has used repository architectural style in various projects. There are many such similar columns regarding different software architectural styles in this dataset. These columns indicate how many times the developer employed each architectural style while working on projects. In this dataset high level data on various architectural styles is provided i.e., it just provides the information regarding the number of times an architectural style is being used. Details regarding the below mentioned different software architectural styles are presented in the dataset that we have considered for analysis.

- Repository Architectural Styles
- Client Server Styles
- Abstract Machine Styles
- Object Oriented Styles
- Function Oriented Styles
- Event Driven Styles
- Layered Styles
- Pipes & Filters Architectural Styles
- Data centric Architectural Styles
- Blackboard Architectural Styles
- Rule Based Architectural Styles
- Publish Subscribe Architectural Styles
- Asynchronous Messaging Architectural Styles
- Plug-ins Architectural Styles
- Micro-kernel Architectural Styles
- Peer-to-Peer Architectural Styles
- Domain Driven Architectural Styles

The machine learning algorithms used for predicting the patterns are stated below.

3.1 Logistic Regression:

Logistic regression is a classification model used to predict the relation between dependent and independent features. In binary logistic regression the target or dependent variable should be a single feature whereas the independent variables could be multiples features. This model is mostly employed when the target variable is in binary format i.e., if the data to be predicted is either yes or no and true or false. When the dependent or target attribute has more than two possible unordered options then multinomial logistic regression can be used. If the target variable has more than two possible ordered options, then ordinal logistic regression can be used for predicting the relation between the target and independent features.

In this paper we have used multinomial logistic regression as our target variable has more than two possible unordered options. The target variable "Last degree" has three possible values which are "BSC (CS or SE)", "MS (CS or SE)" and "PhD (CS or SE)". The target variable "Job Experience" has seven possible values which are "Education", "Software Industry", "Other", "Software Industry; Education", "Education; Other", "Software Industry; Education; Other" and "Software Industry; Other".

3.2 Decision Tree Classifier:

Decision tree classifier can be used for performing classification tasks. This algorithm consists of a tree like structure and has root node, intermediate nodes, branches, and leaf nodes. The target or the dependent variable forms the leaf nodes and the test to be performed on independent variables forms the root and intermediate nodes. The results of these tests are represented by the branches in the tree. Decision tree classifier can be used when we are dealing with multidimensional data i.e., when we have a greater number of features. Based on the results of the tests performed on independent features the algorithm will generate subsets and this process is continued until the lead nodes are reached. This classifier was used to train the model to predict the patterns in the educational qualifications of the developers. We have observed overfit condition while using this classifier.

3.3 KNN Classifier:

K-Nearest Neighbor algorithm makes prediction based on the proximity between the data points. This classifier can be used for both classification and regression tasks. However, it is mostly employed for performing classification tasks. This classifier assigns a label to the new data point based on label that is more commonly reported near that new data point.

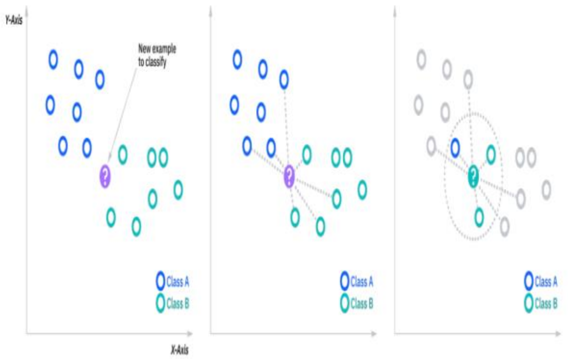


Fig 3.3.1: KNN Classifier

4. Literature Review

Anubha Sharma, Manoj Kumar, Sonali Agarwal [1] explained how important it is to select the most appropriate architecture style while developing any software system and explained how it is important to know that this overview should contain only the requirement of the system and not the implementation. There are a various architecture styles and, in this paper, they presented a detailed overview on the various architectural styles that are used while developing a software system.

Martin Bichler and Christine Kiss [2] in their article compared the three machine learning models and explained the details of each. In this paper they also mentioned some of the drawbacks of each machine learning techniques. From this paper we got the overview of the 3 machine learning techniques.

Sirojiddin Komolov, Gcinizwe Dlamini, Swati Megha and Manuel Mazzara [3] in their paper they took the source code as their input and predicted the architecture style from the input. They trained the model for 9 different machine learning methods and checked the outputs for them respectively. They also talked about the correlation between the elements in the paper. In the results they have trained the model with the various machine learning techniques and have found the relation.

Qadeem Khan, Dr. Usman Qamar, Dr. Wasi Haider Butt, Dr. Saad Rehman [4] in their article have worked on building a dataset as they felt that there were not so many datasets for the architectural styles. They provided the dataset based on various data mining techniques. In this paper there is a lot of information about the architecture styles and their importance in designing a software system.

Batta Mahesh [5] in his article explained how everything around us uses machine learning in one way or the other and how machine learning is important in real world. He explained what model should be chosen when we have less amount of data, labeled data etc. In this paper he also explained about various machine learning algorithms namely decision tree, k-means clustering and support vector machine in greater detail.

Cullen Schaffer [6] in his paper explained the reasons for the overfitting cases in the decision tree machine learning model. Sometimes the model will give a very good result or the output for the training data, but it will fail to give best results for the new data that is given to it and this situation is called as an overfitting case. It is explained that these situations may happen because of various reasons and one such reason is noise that is present in the dataset.

Issam El Naqa and Martin J. Murphy [7] in their paper explained about what machine learning is all about. It tells us the basic and simple meaning of what machine learning is and in what situations machine learning works. He also gave examples of real world were in day-to-day life the machine learning is used by people unknowingly like while watching a movie or while using any social network app, the recommendations that we get is also based on the machine learning algorithms that is used while designing the application. So, like these there are several advantages of machine learning which are explained in the article.

Cullen Schaffer [6] in his paper explained the reasons for the overfitting cases in the decision tree machine learning model. Sometimes the model will give a very good result or the output for the training data, but it will fail to give best results for the new data that is given to it and this situation is called as an overfitting case. It is explained that these situations may happen because of various reasons and one such reason is noise that is present in the dataset.

6. Body

We have analyzed the dataset by following various steps. We have initially preprocessed and cleaned the dataset by checking for the null and duplicate values. It is observed that there are no null values in the dataset that we have considered for analysis and there are two duplicate values which are dropped. There are few unnecessary columns like “Timestamp” and “Your Good Name” in the dataset which do not contribute to predict the educational qualifications of the developers. So, we have dropped these two columns before performing exploratory data analysis on the attributes present in the dataset. We have acquired the counts for the columns "Organization?", "Last Degree?" and "Job Experience?" and have plotted interactive pie charts.

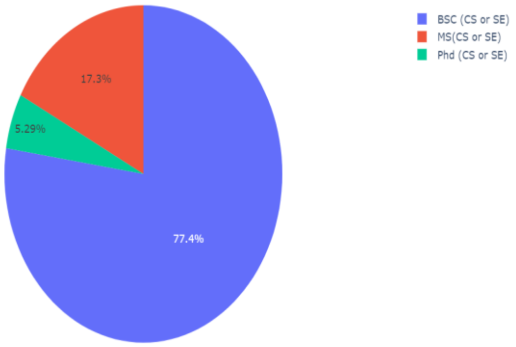


Fig 6.1: Interactive pie chart for value counts of Last Degree

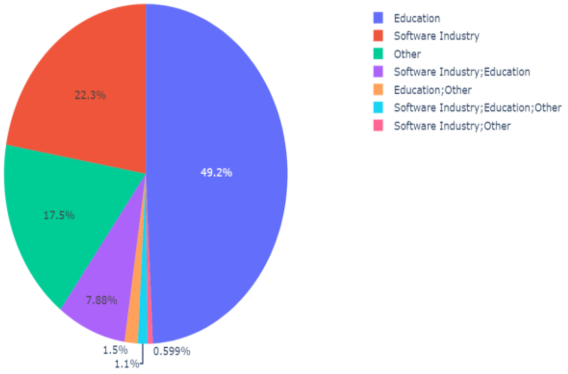


Fig 6.2: Interactive pie chart for value counts of Last Degree and Job Experience

Before applying standardization technique, we analyzed the features that are to be predicted using various machine learning algorithms. We have then plotted interactive histograms that tell us the number of developers with various last degrees who were using a particular architectural style.

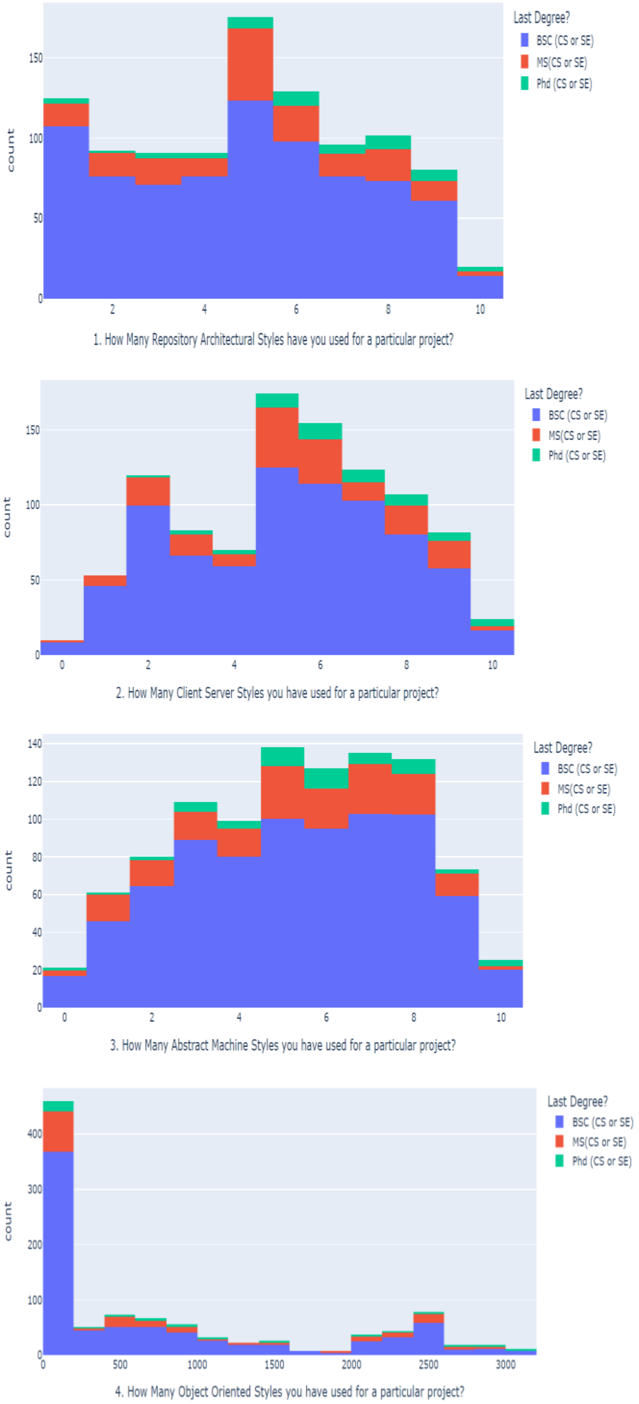


Fig 6.3: Value counts of developers with various Last Degrees using a particular architectural style.

In the same way we plotted the histograms for all the remaining architectural styles in our analysis. Then we done repeated the same process for the feature Job Experience which is one of the target variables in our analysis.

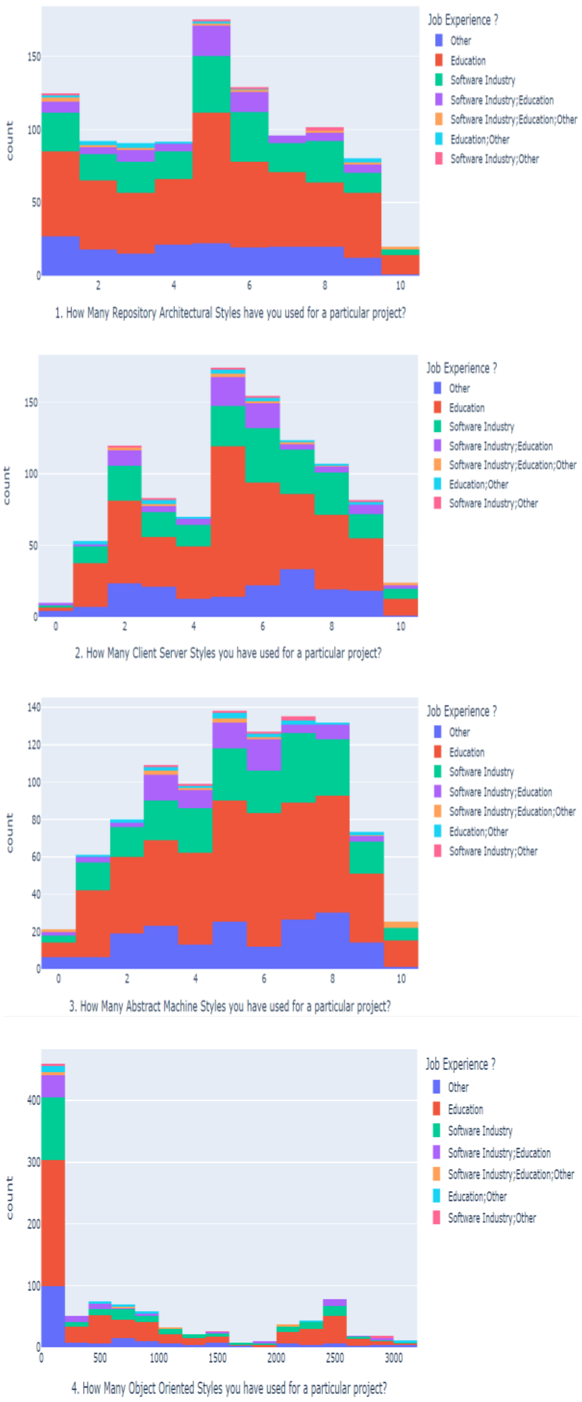


Fig 6.4: Value counts of developers with various Job Experience using a particular architectural style.

In the same way we have plotted histograms for all the remaining architectural styles.

6.1 Label Encoding:

We have performed label encoding to the dataset to convert the input data into machine readable format. If the data that is being fed to the model is not in numeric format, then the machine cannot understand the input. So, it is important to convert the data used for feeding the model into numeric format. In the dataset that we considered both the dependent features “Last Degree?” and “Job Experience?” and another feature “Organization?” are in string format. So, label encoding is used for transforming this data into machine readable format.

6.2 Correlation:

Finding the correlation between all the attributes gives us information regarding how well all the features in the dataset are related. We have plotted the heatmaps to know the correlation between the all the attributes in the dataset. Attributes “4. How Many Object-Oriented Styles you have used for a particular project?” and “5. How Many Function Oriented Styles you have used for a particular project?” were highly correlated with value of 0.77 amongst all the attributes. From this we can say that attributes in the dataset are not highly correlated.

The correlation between the target variables and the different architectural styles is very low indicating that the educational qualifications of the developers are not dependent on the software architectural styles they have used in various projects. We have plotted an interactive heatmap to know correlation between various attributes and another heatmap shows the correlation values for various attributes.

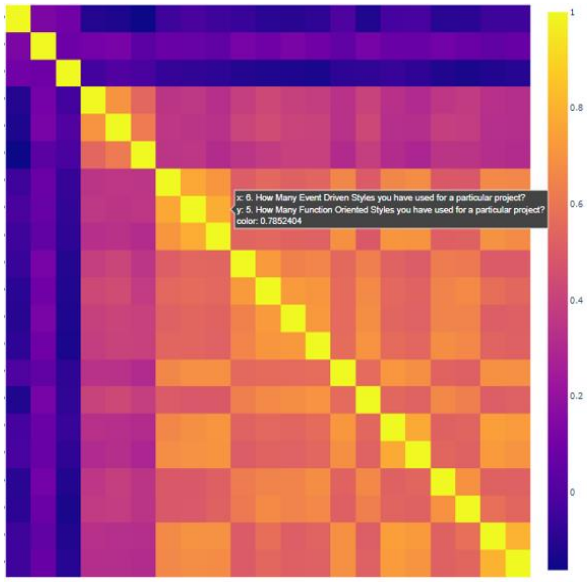

x: 6. How Many Event Driven Styles you have used for a particular project?
y: 5. How Many Function Oriented Styles you have used for a particular project?
color: 0.7852404
1
0.8
0.6
0.4
0.2
0

Fig 6.2.1: Interactive Heatmap

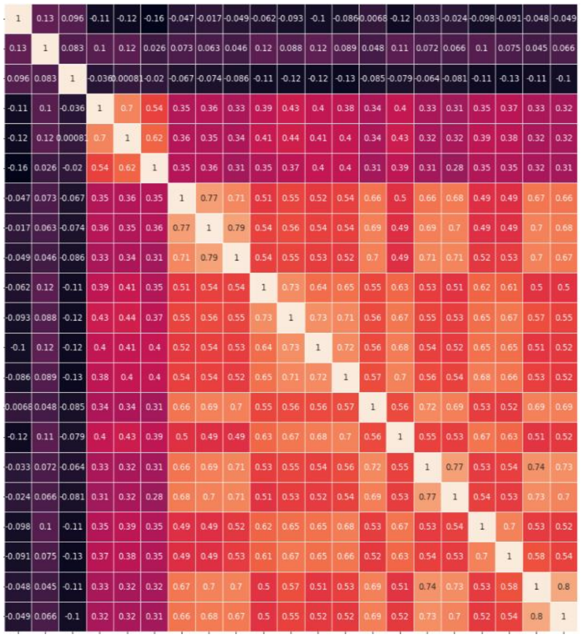


Fig 6.2.2: Heatmap

6.3 Standardization:

Generally, before training the model with machine learning algorithms we must make sure that all the independent features that are used for predicting the dependent feature should be in the same scale to ensure that the model is trained accurately. If all the features are not in the same scale, then the model may be biased towards the feature that has wide range of values. For example, if we have two independent features, one feature has values in the range of 1 to 10,000 and other feature has values in the range of 1 to 1000. When a model is trained with these features the model maybe biased towards the feature in the scale of 1 to 10,000 as the scale of these features is widely varied. It believes that the feature with high values is important and will start predicting the target variable based only the attribute with wide range of values.

To avoid such conditions, we are applying standardization technique to the dataset to so that all the features are scaled to a common scale. In this way we can ensure that all the attributes are contributing equally to the prediction of the target variable, and we can eliminate the biasing condition. We can also use normalization technique to the same. Normalization is also used for rescaling all the features to a common scale to ensure that all the features contribute equally to predict the output. In this paper we have employed standardization technique to rescale the features in the dataset.

After rescaling all the features to a common scale, we have trained the model with various machine learning algorithms.

6.4 Machine Learning Algorithms:

We have employed three different machine learning algorithms to predict our dependent variables which are “Last Degree” and “Job Experience”. They are:

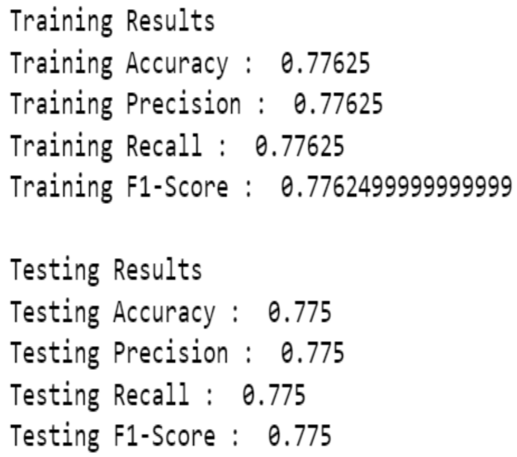
- Logistic Regression
- Decision Tree Classifier
- KNN Classifier

Results of all the models were compared over various performance metrics. The performance metrics that we chose for validating our model are Accuracy, Precision, Recall and F1-score.

6.5 Target variable: Last Degree?

6.5.1 Results of Logistic Regression:

For the training and testing sets in logistic regression model the results of various performance metrics are shown in the below figure. The results produced by this model were not as expected as the accuracy was very low for both testing and training data.

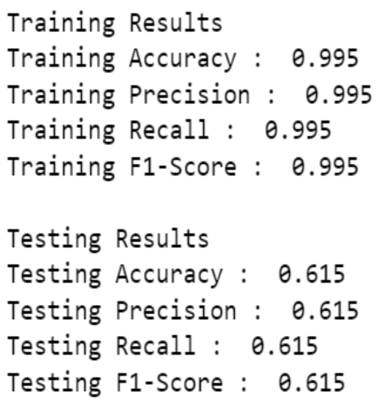


```
Training Results
Training Accuracy :  0.77625
Training Precision :  0.77625
Training Recall :  0.77625
Training F1-Score :  0.7762499999999999

Testing Results
Testing Accuracy :  0.775
Testing Precision :  0.775
Testing Recall :  0.775
Testing F1-Score :  0.775
```

Fig 6.5.1.1: Logistic Regression results

6.5.2 Results of Decision Tree classifier:

For the training and testing sets in decision tree classifier model the results of various performance metrics are shown in the below figure. The results produced by this model were not as expected as the accuracy was very low for both testing and training data.



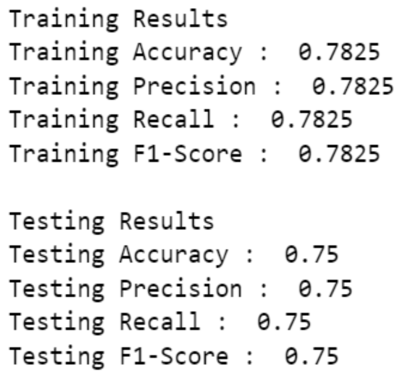
```
Training Results
Training Accuracy :  0.995
Training Precision :  0.995
Training Recall :  0.995
Training F1-Score :  0.995

Testing Results
Testing Accuracy :  0.615
Testing Precision :  0.615
Testing Recall :  0.615
Testing F1-Score :  0.615
```

Fig 6.5.2.1: Decision tree classifier results

Overfitting case is observed when the machine learning model fits closely to the training data but not the testing data. We have observed the overfitting condition when training the model with decision tree classifier. This condition might have raised due to the large amount of irrelevant data present in the dataset.

6.5.3 Results of KNN classifier:

For the training and testing sets in KNN classifier model the results of various performance metrics are shown in the below figure. The results produced by this model were not as expected as the accuracy was very low for both testing and training data.



```
Training Results
Training Accuracy :  0.7825
Training Precision :  0.7825
Training Recall :  0.7825
Training F1-Score :  0.7825

Testing Results
Testing Accuracy :  0.75
Testing Precision :  0.75
Testing Recall :  0.75
Testing F1-Score :  0.75
```

Fig 6.5.3.1: KNN classifier results

6.5.4 Plots of Classification Results:

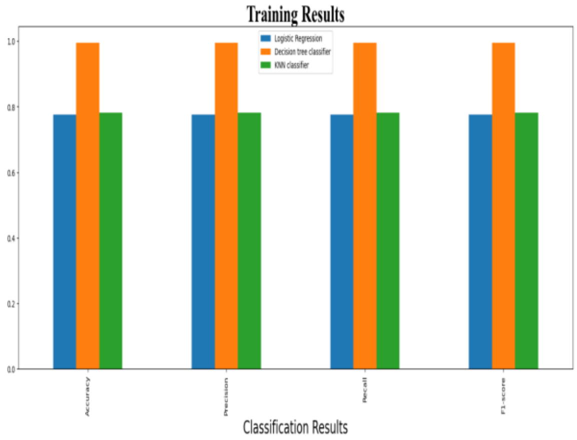


Fig 6.5.4.1: Classification Results

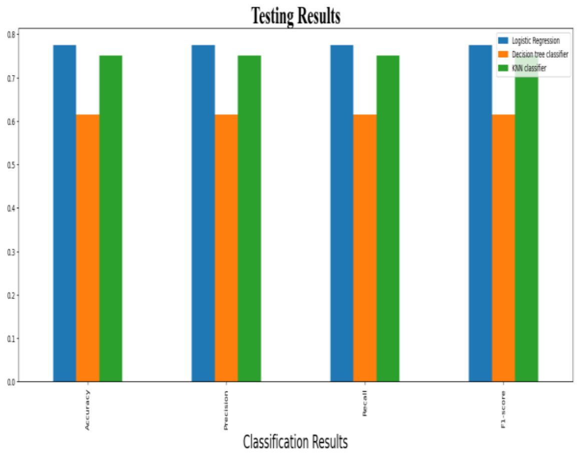
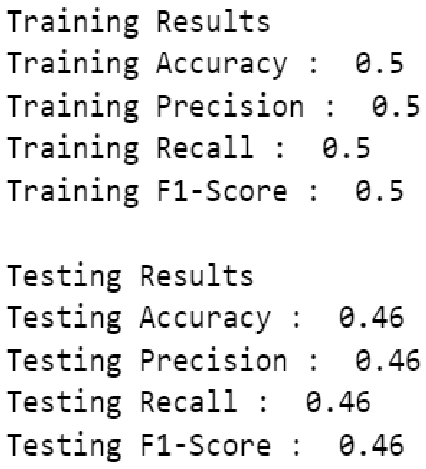


Fig 6.5.4.2: Classification Results

6.6Target variable: Job Experience?

6.6.1 Results of Logistic Regression:

For the training and testing sets in logistic regression model the results of various performance metrics are shown in the below figure. The results produced by this model were not as expected as the accuracy was very low for both testing and training data.



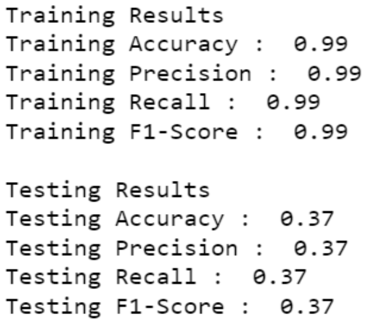
```
Training Results
Training Accuracy :  0.5
Training Precision :  0.5
Training Recall :  0.5
Training F1-Score :  0.5

Testing Results
Testing Accuracy :  0.46
Testing Precision :  0.46
Testing Recall :  0.46
Testing F1-Score :  0.46
```

Fig 6.6.1.1: Logistic Regression results

6.6.2 Results of Decision Tree classifier:

For the training and testing sets in decision tree classifier model the results of various performance metrics are shown in the below figure. The results produced by this model were not as expected as the accuracy was very low for both testing and training data.

The F1-score for training set is 0.99 and for testing set it is 0.37. This condition is called as overfitting.



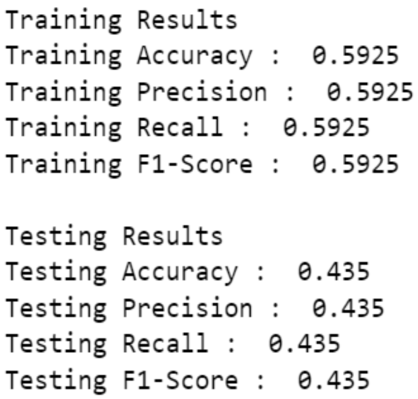
```
Training Results
Training Accuracy :  0.99
Training Precision :  0.99
Training Recall :  0.99
Training F1-Score :  0.99

Testing Results
Testing Accuracy :  0.37
Testing Precision :  0.37
Testing Recall :  0.37
Testing F1-Score :  0.37
```

Fig 6.6.2.1: Classification Results

When predicting the target variable job experience also we have observed overfitting condition.

6.6.3 Results of KNN classifier:

For the training and testing sets in KNN classifier model the results of various performance metrics are shown in the below figure. The results produced by this model were not as expected as the accuracy was very low for both testing and training data.



```
Training Results
Training Accuracy :  0.5925
Training Precision :  0.5925
Training Recall :  0.5925
Training F1-Score :  0.5925

Testing Results
Testing Accuracy :  0.435
Testing Precision :  0.435
Testing Recall :  0.435
Testing F1-Score :  0.435
```

Fig 6.6.3.1: Classification Results

6.6.4 Plots of Classification Results:

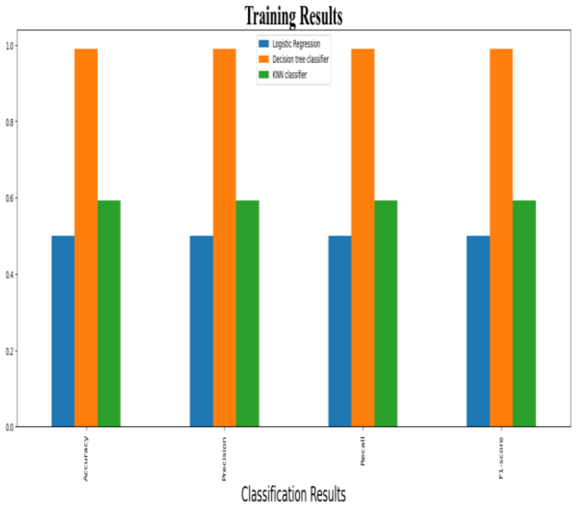


Fig 6.6.4.1: Classification Results

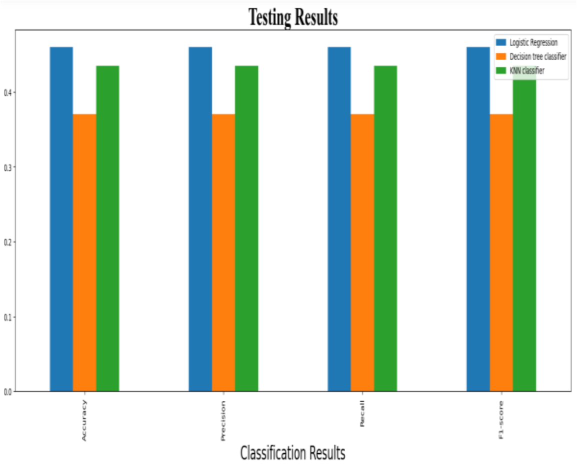


Fig 6.6.4.2: Classification Results

7. Conclusion

Software architectural styles are really very useful as they provide us information regarding how the entire code of the business is being organized on a high level. So, we have worked on a dataset that contains information regarding the educational qualifications of the developers and the different software architectural styles they have used in projects. We built models to predict the educational qualifications of the developer such last degree and job experience based on the software architectural styles they have used.

8. Works Cited

AnubhaSharma, ManojKumar, and SonaliAgarwal, Author links open overlay, et al. “A Complete Survey on Software Architectural Styles and Patterns.” Procedia Computer Science, Elsevier, 21 Nov. 2015, https://www.sciencedirect.com/science/article/pii/S187705091503183X.

Martin Bichler, and Christine Kiss A Comparison of Logistic Regression, K-Nearest Neighbor, and Decision tree induction for campaign management. https://aisel.aisnet.org/cgi/viewcontent.cgi?httpsredir=1&article=1806&context=amcis2004.

Komolov, Sirojiddin, et al. “Towards Predicting Architectural Design Patterns: A Machine Learning Approach.” MDPI, Multidisciplinary Digital Publishing Institute, 12 Oct. 2022, https://www.mdpi.com/2073-431X/11/10/151.

Qadeem Khan, Dr. Usman Qamar, Dr. Wasi Haider Butt, Dr. Saad Rehman Dataset Designing of Software Architectures Styles for Analysis through data mining clustering algorithms https://www.researchgate.net/publication/312596176_Dataset_Designing_of_Software_Architectures_Styles_for_Analysis_through_Data_Mining_Clustering_Algorithms.

Batta Mahesh Machine Learning Algorithms - a Review - Researchgate. https://www.researchgate.net/profile/Batta-Mahesh/publication/344717762_Machine_Learning_Algorithms_-A_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-A-Review.pdf?eid=5082902844932096.

Schaffer, Cullen. “When Does Overfitting Decrease Prediction Accuracy in Induced Decision Trees and Rule Sets?” SpringerLink, Springer Berlin Heidelberg, 1 Jan. 1991, https://link.springer.com/chapter/10.1007/BFb0017014.

El Naqa, Issam, and Martin J. Murphy. “What Is Machine Learning?” SpringerLink, Springer International Publishing, 1 Jan. 1970, https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1.

Charbuty, Bahzad, and Adnan Abdulazeez. "Classification based on decision tree algorithm for machine learning." Journal of Applied Science and Technology Trends 2.01 (2021): 20-28.

jamil, Irfan. “Machine Learning and System Engineering by SIO-IONG AO, Burghard Rieger, Mahyar A. Amouzegar, 2010.” Academia.edu, 26 May 2014, https://www.academia.edu/4147942/Machine_Learning_and_System_Engineering_by_Sio_Iong_Ao_Burghard_Rieger_Mahyar_A_Amouzegar_2010.

Yu, Jun, and Dacheng Tao. “Modern Machine Learning Techniques and Their Applications in Cartoon Animation Research.” Wiley.com, 18 Mar. 2013, https://www.wiley.com/en-us/Modern+Machine+Learning+Techniques+and+Their+Applications+in+Cartoon+Animation+Research-p-9781118115145.

Yu, Jun, and Dacheng Tao. “Modern Machine Learning Techniques and Their Applications in Cartoon Animation Research.” Wiley.com, 18 Mar. 2013, https://www.wiley.com/en-us/Modern+Machine+Learning+Techniques+and+Their+Applications+in+Cartoon+Animation+Research-p-9781118115145.

Fangchao Tian School of Computer Science, et al. “Relationships between Software Architecture and Source Code in Practice: : An Exploratory Survey and Interview: Information and Software Technology: Vol 141, NO C.” Information and Software Technology, 1 Jan. 2022, https://dl.acm.org/doi/abs/10.1016/j.infsof.2021.106705.

Sarker, Iqbal H. “Machine Learning: Algorithms, Real-World Applications and Research Directions - SN Computer Science.” SpringerLink, Springer Singapore, 22 Mar. 2021, https://link.springer.com/article/10.1007/s42979-021-00592-x.

Ardabili, Sina F., et al. “Covid-19 Outbreak Prediction with Machine Learning.” MDPI, Multidisciplinary Digital Publishing Institute, 1 Oct. 2020, https://www.mdpi.com/1999-4893/13/10/249.

Fatima, Meherwar, and Maruf Pasha. “Survey of Machine Learning Algorithms for Disease Diagnostic.” Journal of Intelligent Learning Systems and Applications, Scientific Research Publishing, 24 Jan. 2017, https://www.scirp.org/journal/paperinformation.aspx?paperid=73781.

Charbuty, Bahzad, and Adnan Abdulazeez. "Classification based on decision tree algorithm for machine learning." Journal of Applied Science and Technology Trends 2.01 (2021): 20-28.

Zhang, Shichao, et al. "Efficient kNN classification with different numbers of nearest neighbors." IEEE transactions on neural networks and learning systems 29.5 (2017): 1774-1785.

Tan, Songbo. "An effective refinement strategy for KNN text classifier." Expert Systems with Applications 30.2 (2006): 290-298.

Nick, Todd G., and Kathleen M. Campbell. “Logistic Regression.” SpringerLink, Humana Press, 1 Jan. 1970, https://link.springer.com/protocol/10.1007/978-1-59745-530-5_14.

Wright, R. E. (1995). Logistic regression. In L. G. Grimm & P. R. Yarnold (Eds.), *Reading and understanding multivariate statistics* (pp. 217–244). American Psychological Association.

Wiyono, Slamet, and Taufiq Abidin. "Comparative study of machine learning KNN, SVM, and decision tree algorithm to predict student’s performance." International Journal of Research-Granthaalayah 7.1 (2019): 190-196.

Rajaguru, Harikumar, and Sannasi Chakravarthy SR. "Analysis of decision tree and k-nearest neighbor algorithm in the classification of breast cancer." Asian Pacific journal of cancer prevention: APJCP 20.12 (2019): 3777.

Rahman, Hezlin Aryani Abd, et al. "Comparisons of ADABOOST, KNN, SVM and logistic regression in classification of imbalanced dataset." International Conference on Soft Computing in Data Science. Springer, Singapore, 2015.