Ray Sadowski
FIN695

Final Project Writeup

Introduction

The data science field has become increasingly important in recent years as companies seek to extract insights from their data to inform business decisions. With the increasing demand for data scientists, it has become vital for both job seekers and companies to understand the factors that determine the average salary for data science roles. To this end, the Jobs dataset from Glassdoor on Kaggle provides a valuable resource for analyzing job listings from various companies in different industries for data scientist roles.

This dataset contains a wealth of information, including job title, salary estimate, job description, rating, company name, location, headquarters, size, founded, type of ownership, industry, sector revenue, and competitors of the person looking at the job. Additionally, the dataset contains information about key skills that data scientists should possess, including Python, R, Spark, AWS, and Excel. The ability to leverage these skills is critical to a data scientist's success and can have a significant impact on their earning potential.

The goal of this predictive analysis is to determine the factors that determine the average salary for data science roles based on the variables in the Jobs dataset. This will involve using a combination of feature selection, data cleaning, and machine learning algorithms to identify the most relevant variables and develop an accurate predictive analysis. In this write-up, I will outline the steps taken to achieve this goal, including data collection, cleaning, feature selection, and model development. I will also present the results of my analysis and discuss the implications for companies and job seekers in the data science field.

Ray Sadowski
FIN695

Variable to Predict

The variable I have chosen to predict in my models is the average salary. Salary is one of the most important factors to consider when looking for a job. As a data scientist, understanding the salary range for the job title can help job seekers evaluate the value of the job opportunity and make informed decisions about their career paths. In this dataset, the average salary is presented as a range and is a function of various factors such as min salary, max salary, company size, industry, and individual skillset.

It is important to note that predicting the average salary accurately is a complex task that requires analyzing various factors that contribute to the salary range. I will use the average salary as my left-hand variable and consider the other variables in the dataset as my predictors. By building a model that accurately predicts the average salary based on the given variables, I aim to provide insights into the factors that influence the average salary of a data scientist.

I will explore the relationship between the predictors and the average salary to identify which variables have the strongest correlation with the target variable. My predictive analysis will help job seekers evaluate the value of potential job opportunities and help employers determine the competitive salary range for data scientist roles. Through this analysis, I aim to provide valuable insights that could benefit both job seekers and employers in the data science field.

Data Collection:

For this project, I decided (thank you for the inspiration professor) to collect the Salary Prediction dataset from Glassdoor, a popular job search website that provides insights into companies and their job offerings. The dataset was obtained from Kaggle, a popular platform for

hosting and sharing data science projects. The dataset contains 742 observations and 28 variables, providing a comprehensive view of job listings for data scientists across various industries.

To begin the data collection process, I downloaded the dataset from Kaggle in a CSV format. I then imported the dataset into Python using the pandas library, which allowed me to easily manipulate and analyze the data. The pandas library provides powerful tools for data wrangling and analysis, allowing us to explore the dataset with ease and prepare it for modeling.

The Salary Prediction dataset from Glassdoor is an excellent resource for studying the factors that contribute to the average salary of a data scientist. The dataset contains various features such as company size, industry, and individual skillset, making it a rich source of information for my predictive analysis. By leveraging this dataset, I can develop a predictive analysis that accurately predicts the average salary of a data scientist based on various factors.

Institutional Details

In order to ensure that my analysis is unbiased and ethical, I need to be aware of any potential biases, legal or ethical considerations related to the data. The Jobs dataset from Glassdoor on Kaggle is a public dataset that is freely available to anyone. However, I should be aware that the dataset may have some selection bias, as it only includes job listings from companies that have posted their listings on Glassdoor. Additionally, I should be aware of any potential legal or ethical considerations related to using the data, such as data privacy or intellectual property rights.

Ray Sadowski
FIN695

Data Cleaning

The data cleaning process is crucial in preparing the dataset for analysis. Initially, I checked for any missing values in the dataset and removed any observations with missing data. As the goal of the analysis is to predict the average salary, I removed any observations that had an hourly pay rate instead of a salary. This was done to ensure that I only considered salaried positions.

After filtering the data, I examined the variables and removed any that were not relevant to my analysis, such as job description and industry. The salary estimate variable was transformed into a numerical variable by calculating the average of the salary range. This transformation provided a more accurate representation of the salary.

Additionally, I created dummy variables for categorical variables, such as Job Title, Type of ownership, Industry, and Job_State. This step was taken to convert categorical variables into a form that can be easily understood by machine learning algorithms. The dummy variables were created using the pandas get_dummies() method technique. This process created new columns for each category, and a dummy value was assigned to indicate if the observation belonged to that category. The final dataset was then ready for use in building my predictive analysis.

Models to Predict

In order to predict the average salary, I utilized several machine learning algorithms, including linear regression, polynomial regression, support vector regression (SVR), decision tree regression, and random forest regression. Each of these models has its own strengths and weaknesses, and the choice of which model to use depends on the specific characteristics of the dataset and the problem being addressed.

Linear regression is a straightforward model that seeks to establish a linear relationship between the response variable and the predictors. Polynomial regression, on the other hand, allows for a more complex relationship between the variables by adding polynomial terms to the model. SVR is a powerful model that can handle both linear and non-linear relationships between the variables by mapping the data to a higher-dimensional feature space. Decision tree regression is a non-parametric model that can handle both numerical and categorical variables and is able to capture complex interactions between the variables. Random forest regression is an ensemble model that combines multiple decision trees to improve prediction accuracy and reduce overfitting.

Data splitting

Data splitting is an important step in machine learning to assess how well a model is likely to perform when presented with new, unseen data. In my predictive analysis, I split the dataset into training and test sets using the widely adopted 67/33 ratio. To achieve this, I utilized the train_test_split() function from the sklearn.model_selection library, which randomly divided the dataset into two subsets, one for training and the other for testing. The training dataset (67%) was used to fit the model, while the testing dataset (33%) was used to evaluate its performance on new data.

Splitting the dataset in this way allowed me to examine the model's ability to generalize to new, unseen data. Moreover, it helped prevent overfitting of the model to the training data, which can lead to poor performance on new data. The train_test_split() function provided an easy

and efficient way to randomly partition the data into training and test sets, making this step of the analysis quick and simple.

Model Comparison / Results

In my analysis, I used several machine learning algorithms to predict the average salary, including Linear Regression, Polynomial Regression, Support Vector Regression (SVR), Decision Tree Regression, and Random Forest Regression. I then compared the performance of each model using a train/test split of 67/33.

The Linear Regression model had a train score of 0.963 and a test score of -3.3946e+28. The high discrepancy between train and test score suggests that the model is overfitting to the training data and performs poorly on unseen data.

The Polynomial Regression model had a perfect train score of 1.000 and a test score of 0.524, indicating that the model is performing well on both training and test data. However, the difference between train and test score is relatively large, indicating some overfitting may still be present.

The SVR model had a train score of 0.955 and a test score of 0.497, indicating good performance on both training and test data. The small difference between train and test scores suggests that the model is not overfitting to the training data.

The Decision Tree Regression model had a perfect train score of 1.000 and a test score of 0.383, indicating some overfitting may be present. The model may not generalize well to new data.

The Random Forest Regression model had a train score of 0.945 and a test score of 0.455, suggesting that the model is performing well on both training and test data, but may have some overfitting present.

Overall, the Polynomial Regression and SVR models performed the best on my data, with the lowest difference between train and test scores and the highest test scores. The Linear Regression model showed signs of severe overfitting, and the Decision Tree Regression model may not generalize well to new data. The Random Forest Regression model had good performance but may have some overfitting present.

Final Model Selection

After training and testing multiple machine learning models on my dataset, I can now evaluate their performance and select the best model for my predictive analysis. Out of the five models that were tested - Linear Regression, Polynomial Regression, SVR, Decision Tree Regression, and Random Forest Regression - two models emerged as the top performers: Polynomial Regression and SVR.

Both the Polynomial Regression and SVR models had the lowest difference between train and test scores and the highest test scores. These models were able to effectively capture the underlying trends and patterns in my data, allowing them to make accurate predictions on new data.

On the other hand, the Linear Regression model showed signs of severe overfitting, with a large discrepancy between the train and test scores. This suggests that the model is not able to generalize well to new data and may not be the best choice for my predictive analysis.

Similarly, the Decision Tree Regression model had a perfect score on the training set, but its performance on the test set was comparatively lower. This indicates that the model may not generalize well to new data, which is a critical consideration for any predictive model.

Finally, the Random Forest Regression model performed well on the test set, but there may be some overfitting present in the model. Overfitting occurs when the model fits too closely to the training data and fails to generalize well to new data.

Overall, based on the results of the model testing and evaluation, I can confidently select the Polynomial Regression and SVR models as the top performers for my predictive analysis. These models have the highest potential to make accurate predictions on new data, allowing me to gain valuable insights into the average salary trends and patterns in the Glassdoor data scientist dataset.

Conclusion

In conclusion, building a predictive model for the Jobs dataset from Glassdoor is a challenging but rewarding task. By following the steps outlined in this writeup, I can build an accurate and reliable model that can predict the average salary based on the variables in the dataset.