

Theme of Project,

The theme of this project is to explore and prepare data for the application of machine learning algorithms to predict whether a passenger survived or not on the Titanic. The project involves loading and cleaning the data using Pandas and NumPy libraries in Python. The project then explores the relationships between the survival rate and various features such as gender, passenger class, and number of siblings/spouses or parents/children aboard. The project also involves feature engineering by creating a new feature called 'Title' extracted from the 'Name' column, and using it to fill in missing values in the 'Age' column. Age banding is also applied to both train and test datasets to group ages into categorical bins. The project is an example of the data wrangling and preprocessing tasks typically performed in machine learning projects.

Report of Project,

This project is an example of data preprocessing and feature engineering for the Titanic dataset, which is a common introductory dataset for data analysis and machine learning.

The code first loads the training and test data into Pandas dataframes and combines them for future data wrangling. It then performs some exploratory data analysis (EDA) by calculating the mean survival rate for different features such as passenger class, gender, and number of siblings/spouses or parents/children aboard.

The code then drops some columns that are not useful for the analysis, such as 'Ticket' and 'Cabin', and adds a new feature called 'Title' by extracting the title from the 'Name' column. The 'Title' feature is then cleaned up by merging similar titles together and replacing certain titles with 'Rare'. The 'Title' feature is then mapped to numerical values and any missing values are filled with 0.

The 'Name' and 'PassengerId' columns are dropped from the train dataframe and the 'Name' column is dropped from the test dataframe. The 'Sex' feature is mapped to numerical values, with 'female' mapped to 1 and 'male' mapped to 0.

The code then imputes missing values for the 'Age' feature by guessing the median age for each combination of 'Sex' and 'Pclass' and rounding to the nearest 0.5. The imputed age values are then binned into age bands and assigned numerical values. The original 'Age' feature is dropped and the new 'AgeBand' feature is added to the datasets.

Overall, this code is an example of some basic data preprocessing and feature engineering techniques that are commonly used in data analysis and machine learning. These techniques are important for cleaning up and preparing the data for further analysis and modeling.

Making predictions on a dataset in machine learning means using a trained model to guess the outcome of new data points. You first prepare the data, load the trained model, input the new data, generate predictions, and then interpret the results.

My approach was Support Vector Classification,

SVC stands for Support Vector Classification, which is a type of supervised learning algorithm used in machine learning. Specifically, it is used for classification problems where you want to predict which category a new observation belongs to based on a set of input variables.

The basic idea behind SVC is to find the hyperplane in a high-dimensional space that best separates the different classes. This hyperplane is then used to classify new observations based on which side of the hyperplane they fall on.

One of the main advantages of SVC is that it is effective in high-dimensional spaces, meaning that it can handle a large number of input variables. Additionally, SVC is often able to handle non-linear relationships between the input variables and the output variable.

Overall, **if you have a classification problem and are dealing with a large number of input variables or non-linear relationships**, SVC might be a good choice of algorithm to use.

To use SVC, you need to follow these steps:







1. **Collect and preprocess your data:** You need to gather a dataset that contains examples of the categories you want to classify. The data should be preprocessed, which may involve tasks such as cleaning, normalizing, and feature extraction.
2. **Split the data into training and testing sets:** You need to split your data into two sets: a training set and a testing set. The training set is used to train the SVC model, while the testing set is used to evaluate its performance.
3. **Choose appropriate hyperparameters:** SVC has hyperparameters that need to be tuned for optimal performance. These include the kernel type, regularization parameter C , and gamma (for non-linear kernels). You can use techniques such as grid search and cross-validation to find the optimal hyperparameters.
4. **Train the SVC model:** Once you have chosen the hyperparameters, you can train the SVC model on the training set. The model will learn to classify the data based on the chosen hyperparameters.
5. **Evaluate the model performance:** After training, you can evaluate the performance of the model on the testing set. This will give you an estimate of how well the model will perform on new, unseen data.

6. Use the model to classify new data: Once you are satisfied with the performance of the model, you can use it to classify new, unseen data. Simply pass the new data through the trained model, and it will predict the category it belongs to based on what it learned during training.

Overall, SVC is a powerful tool for solving classification problems, but it requires careful selection of hyperparameters and appropriate preprocessing of the data to achieve optimal performance.

Conclusion,

Due to the complexity of the classification problem and the large number of input variables or non-linear relationships involved, we opted to use the Support Vector Classification (SVC) method. Our analysis yielded an accuracy of 77%, which was the best result when compared to other algorithms we tested.

	gender_submission.csv Complete · 41m ago · Last try with svc method (it was best)	0.77033
	gender_submission.csv Complete · 44m ago · knn	0.74401
	gender_submission.csv Complete · 1h ago · gauss	0.72009
	gender_submission.csv Complete · 1h ago · perceptron 6	0.57894
	gender_submission.csv Complete · 1h ago · sgdc	0.72488
	gender_submission.csv Complete · 1h ago · 6 random forest	0.76555

After trying various techniques in my code, I found that the Support Vector Classifier (SVC) was the most effective.

Umit Numan Duman

180254019

References

<https://www.kaggle.com/code/mtnumanduman/titanic-data-science-solutions/edit>

<https://www.codecademy.com/article/seaborn-design-i>

<https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8>