Project Group Name: Verispor

Erkam Çetkin
Management Information Systems
Kadir Has University
İstanbul, Turkey
erkamcetkin@stu.khas.edu.tr

Helin Kuş
Management Information Systems
Kadir Has University
İstanbul, Turkey
helinkus@stu.khas.edu.tr

Fatih Mehmet Alagöz
Management Information Systems
Kadir Has University
İstanbul, Turkey
fatihmehmet.alagoz@stu.khas.edu.tr

Meltem Karabastık
Management Information Systems
Kadir Has University
İstanbul, Turkey
meltemkarabastik@stu.khas.edu.tr

Uğur Kılıçdoğan
Management Information Systems
Kadir Has University
İstanbul, Turkey
ugur.kilicdogan@stu.khas.edu.tr

***Abstract*— In this project, we worked on the data in the train.csv file. We first examined the data and used various methods to make the data analyzable. We used data preprocessing steps. We have done the necessary work to analyze, organize and clean up the missing data. After cleaning the data and making it analyzable, we observed the relationship of the data with each other with some correlation methods. After that, we tried to analyze and understand our observed correlation results visually. Finally, we built a model by using machine learning techniques and evaluated our model.**

Keywords—data, data cleaning, data analyze, correlations. machine learning techniques

I. Introduction (*Heading 1*)

This template, modified in MS Word 2007 and saved as a "Word 97-2003 Document" for the PC, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

II. Steps

A. Data cleaning:

Firstly, we struggled with editing the missing data in the dataset. there were too many missing data from many columns in the data set, we filled these missing data according to the type of each column. For example, for numeric values, we filled in the missing values by averaging, and for categorical variables, we filled in the missing data by taking the mode. In addition, by using the pipeline method to fill in the missing data, we both saved time and got a cleaner code terminal.

B. Explore Data:

After making the dataset analyzable, we observed the correlation between the columns. We filtered the columns greater than 0.8 and less than -0.8 and visualized them to see the relationship between them. We deleted some columns with strong positive and negative correlations with each other because highly correlated features can lead to overfitting, which will adversely affect the performance of the model.

C. Feature Engineering:

We rearranged Continuous, Binary and Categorical variables in numerical form to increase the power of the model.

D. Split the Data:

We split the data into training and testing. Our target variable is the last column and our independent variables are the remaining columns.

E. Model Selection:

We use Random Forest Classifier and Logistic Regression to make predictions.

F. Model Evaluation:

Finally, we evaluated the performance of the model according to the model's accuracy parameter.

III. Challenges Encountered

The data columns were not named, and we had a hard time analyzing them because we didn't fully understand the purpose of the dataset. For example, we could not understand which columns to analyze with each other for the correlation heatmap. When we look at the correlation heatmap of the whole data set, there is no meaningful result because there are too many columns, and the resulting graph cannot be read. Still, we showed correlation values greater than 0.8 on the heatmap with the filtering method, and the output was cleaner.

It was also difficult to analyze and organize the missing data correctly, as the number of missing data was relatively high. More research has been done to address this issue to identify the causes of missing data and how this data can be changed. In addition, it was difficult for us not to know the logic and purpose of the dataset.

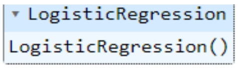
IV. Conclusion

To conclude, we analyzed the data set, applied the data preprocessing steps. We made the model suitable for creating using the Feature Engineering step. Finally, after model selection and model evaluation, we reached 3 different accuracy values.

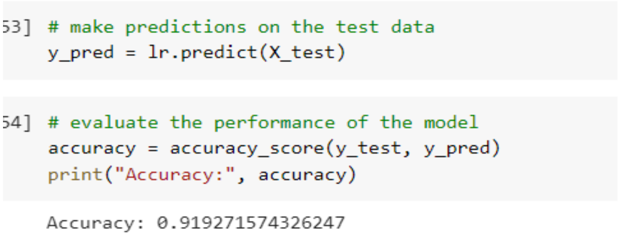
XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

A. *Figures and Tables*

a) Accuracy Rate with Linear Regression



```
▾ LogisticRegression
LogisticRegression()
```

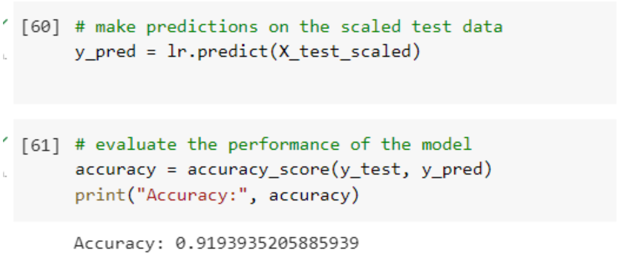


```
53] # make predictions on the test data
    y_pred = lr.predict(X_test)

54] # evaluate the performance of the model
    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy:", accuracy)

    Accuracy: 0.919271574326247
```

b) Standardization of Data:

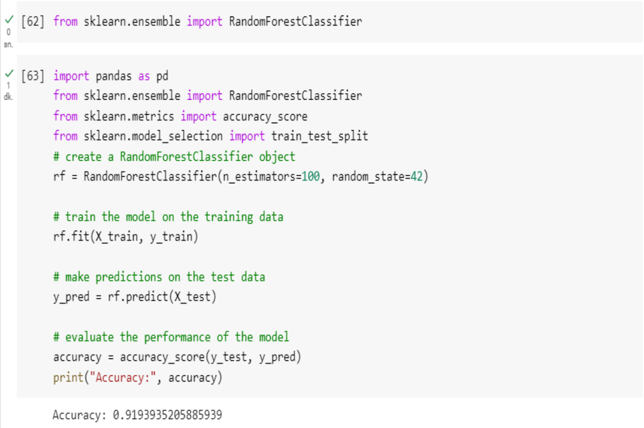


```
[60] # make predictions on the scaled test data
     y_pred = lr.predict(X_test_scaled)

[61] # evaluate the performance of the model
     accuracy = accuracy_score(y_test, y_pred)
     print("Accuracy:", accuracy)

     Accuracy: 0.9193935205885939
```

c) Accuracy with Random Forest Classifier Algorithm:



```
[62] from sklearn.ensemble import RandomForestClassifier

[63] import pandas as pd
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score
     from sklearn.model_selection import train_test_split
     # create a RandomForestClassifier object
     rf = RandomForestClassifier(n_estimators=100, random_state=42)

     # train the model on the training data
     rf.fit(X_train, y_train)

     # make predictions on the test data
     y_pred = rf.predict(X_test)

     # evaluate the performance of the model
     accuracy = accuracy_score(y_test, y_pred)
     print("Accuracy:", accuracy)

     Accuracy: 0.9193935205885939
```

ACKNOWLEDGMENT

E. Çetkin, All codes written, report created.
H. Kuş, Report edited; first sentence added to data exploration field.
U. Kılıçdoğan, Conclusion sentence was written, report edited.
M. Karabastık, Screenshots from code added to report, report edited.
F. M. Alagöz, Report edited and add graphs and tables to the report.

REFERENCES

[1] DataCamp. (n.d.). Introduction to Data Visualization with Matplotlib. Retrieved from https://app.datacamp.com/learn/courses/introduction- to-data-visualization-with-matplotlib

[2] Marloz. (2020). Handling Missing Values with Scikit- Learn Pipelines. Retrieved from https://marloz.github.io/projects/sklearn/pipeline/missing/preprocessing/2020/03/20/sklearn-pipelines- missing-values.h