IBM Datascience Capstone Project

Predicting the severity of accidents in Seattle City

Content

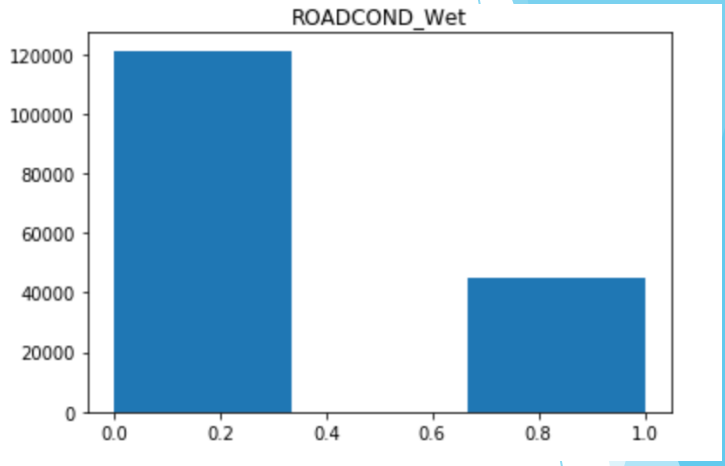
- Introduction
- Methods
  - Data Analysis and Wrangling
  - ML Algorithms
- Results
- Discussion and Conclusion

Introduction
- Predicting the severity of car accidents -

- Interested Stakeholders:
  - First Aid Organizations
  - Public Authorities
  - Infrastructural Planning
  - Navigation System Developers
  - Self Driving Car development
- The Dataset
  - Describes the severity in 4 steps (property damage to fatality)
  - Location: Seattle City

Methods
- Data Preperation -

- Statistical and Graphical analysis



- Drop unnecessary columns
- Drop rows with NaN values
- One-Hot-Encoding in order to obtain categorical features
- Balance Data by dropping ½ of category ‘1’ rows
- Note: only categorical features => no normalization

Out[35]:

	SEVERITYCODE	ADDRTYPE_Block	ADDRTYPE_Intersection	WEATHER_Blowing Sand/Dirt	WEATHER_Clear	WEATHER_Fog/Smog/Smoke	WEATHER_Overcast	WI Cl
1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	0	1	0
3	1	1	0	0	1	0	0	0
5	1	0	1	0	1	0	0	0
6	1	0	1	0	0	0	0	0

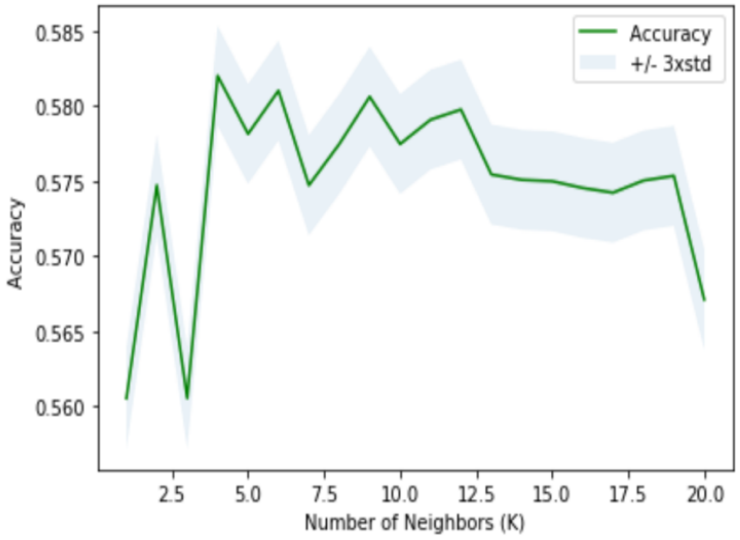
5 rows × 26 columns

- Further, the data has been split into Training and Test set

Methods
- ML algorithms -

- K-nearest Neighbour Classifier
  - Change k number of nearest neighbours 1-20, 50,100, 300
- Support Vector Machine (SVM)
  - Change Kernel ‘linear’, ‘polynomial’, ‘rbf’
- Decision Tree Classifier
  - Change criterion for split ‘gini’, ‘entropy’
- Logistic Regression Classifier
  - Change regularization parameter C

Results
- K-Nearest Neighbours -



Accuracy-score for 50 nearest neighbors is 0.582808847329617

	precision	recall	f1-score	support
1	0.58	0.64	0.61	11230
2	0.59	0.53	0.56	11014
micro avg	0.58	0.58	0.58	22244
macro avg	0.58	0.58	0.58	22244
weighted avg	0.58	0.58	0.58	22244

Accuracy-score for 300 nearest neighbors is 0.5876191332494156

	precision	recall	f1-score	support
1	0.58	0.63	0.61	11230
2	0.59	0.54	0.56	11014
micro avg	0.59	0.59	0.59	22244
macro avg	0.59	0.59	0.59	22244
weighted avg	0.59	0.59	0.59	22244

Best performance: k=300 with accuracy 0.5876

Results
- SVM -

Accuracy-score for Kernel "linear" is 0.5934184499190793

	precision	recall	f1-score	support
1	0.58	0.70	0.64	11230
2	0.61	0.48	0.54	11014
micro avg	0.59	0.59	0.59	22244
macro avg	0.60	0.59	0.59	22244
weighted avg	0.60	0.59	0.59	22244

Accuracy-score for Kernel "poly" is 0.5935982736917821

	precision	recall	f1-score	support
1	0.58	0.70	0.64	11230
2	0.61	0.48	0.54	11014
micro avg	0.59	0.59	0.59	22244
macro avg	0.60	0.59	0.59	22244
weighted avg	0.60	0.59	0.59	22244

Accuracy-score for Kernel "rbf" is 0.5934184499190793

	precision	recall	f1-score	support
1	0.58	0.70	0.64	11230
2	0.61	0.48	0.54	11014
micro avg	0.59	0.59	0.59	22244
macro avg	0.60	0.59	0.59	22244
weighted avg	0.60	0.59	0.59	22244

The accuracy is almost independent of the used kernel

Results
- Decision Tree Classifier -

Best performing model over all:

Accuracy-score for criterion "gini" is 0.9

	precision	recall	f1-score	support
1	1.00	0.80	0.89	5
2	0.83	1.00	0.91	5
micro avg	0.90	0.90	0.90	10
macro avg	0.92	0.90	0.90	10
weighted avg	0.92	0.90	0.90	10

Accuracy-score for criterion "entropy" is 0.8

	precision	recall	f1-score	support
1	0.80	0.80	0.80	5
2	0.80	0.80	0.80	5
micro avg	0.80	0.80	0.80	10
macro avg	0.80	0.80	0.80	10
weighted avg	0.80	0.80	0.80	10

Best performance: criterion ‘gini’ with accuracy 0.9

Results
- Logistic Regression-

Accuracy-score for Regularization Parameter "0.5" is 0.8

	precision	recall	f1-score	support
1	0.80	0.80	0.80	5
2	0.80	0.80	0.80	5
micro avg	0.80	0.80	0.80	10
macro avg	0.80	0.80	0.80	10
weighted avg	0.80	0.80	0.80	10

Accuracy-score for Regularization Parameter "0.001" is 0.6

	precision	recall	f1-score	support
1	0.57	0.80	0.67	5
2	0.67	0.40	0.50	5
micro avg	0.60	0.60	0.60	10
macro avg	0.62	0.60	0.58	10
weighted avg	0.62	0.60	0.58	10

Probably underfitting for C<0.05

Discussion and Conclusion

- SVM and KNN are probably **overfitting training data**
  - Could be addressed as a next step
- Simple, computatinally **cheap algorithms** might just be **accurate enough**
- **X**, **Y**, and **Datetime** should be included in future step to improve prediction
- Potentially get **larger dataset**, if 'learning curve' plots indicate that such would improve model performance
- **Best performing Model overall: Decision Tree with criterion 'gini', acc. 0.9**