Introduction

In this project, we designed and developed a prototype of a transparent data science framework. The prototype supports six models: multinomial logistic regression, two different decision trees, random forest, svc, and neural network. Transparency achieved through open source tools such as SHAP, ELI5, and Lime. These libraries provide a visual explanation of trained models, based on which data scientists can check if models are biased or discriminative. Our team has tested the framework on two datasets: “Black Friday Dataset” and “Bank Loan Status Dataset”.

Datasets

Black Friday Dataset

A dataset we decided to work with was taken from Kaggle. We chose a “Black Friday” dataset[1]. It contains transactions that were made in a retail store. As is said in the description: “The store wants to know better the customer purchase behavior against different products”. The initial problem there is regression, actually, it’s a prediction of the amount of purchase. But some classification problems can also be settled there. The dataset contains 550 000 transactions. It contains different kinds of features either numerical or categorical. There is a list of all features:

User_ID	User
Product_ID	Product
Gender	Boolean
Age	Age customer
Occupation_ID	Occupation of each customer
City_Category	
Stay_In_Current_City_Years	
Marital_Status	
Product_Category_1	
Product_Category_2	
Product_Category_3	
Purchase	Purchase amount in dollars

Obviously, before working with this dataset we applied some preprocessing. The finalized version of this data can be found in ‘/datasets/preprocessed/pre_black_friday.csv’. This data we used to train our models.

Bank Loan Status Dataset

This dataset[2] was also taken from Kaggle. There is no much information about this dataset. It contains 100514 transactions. This dataset has 18 features:

Loan ID	String
Customer ID	String
Current Loan Amount	Numeric
Term	String
Credit Score	Numeric
Annual Income	Numeric
Years in current job	String
Home Ownership	String
Purpose	String
Monthly Debt	Numeric
Years of Credit History	Numeric
Months since last delinquent	Numeric
Number of Open Accounts	Numeric
Number of Credit Problems	Numeric
Current Credit Balance	Numeric
Maximum Open Credit	Numeric
Bankruptcies	Numeric
Tax Liens	Numeric

Same as with the previous dataset, this one also was preprocessed.

Preprocessing

Black Friday:

1. Grouped by **User_ID** and summarised **Purchase** variable.
2. Dropped **Product_ID, Product_Category_1, Product_Category_2, Product_Category_3** and **Purchase_y** variables.
3. Removed duplicated rows.
4. Added dummy variables for **Gender** and **Purchase_x** variables.

Loans:

1. Dropped **Loan ID** and **Customer ID**
2. Reduced percentage if the NaN values
3. Converted **Loan Status** to the boolean variable
4. Removed spaces from variable names and used underscores instead

Machine Learning models and Explanation tools

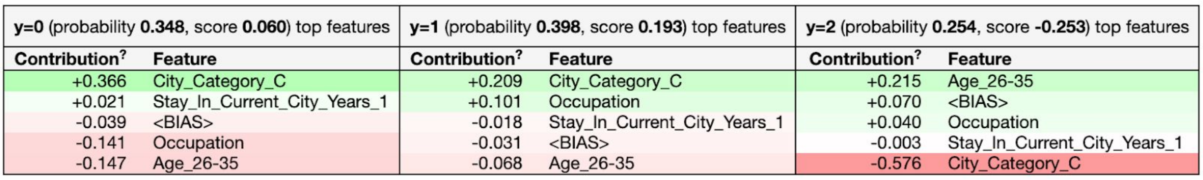
In our project, we used four different models and trained them on our datasets: Logistic Regression, Decision Tree, Random Forest Classifier, XGboost.
In framework implementation following tools were used:

- **ELI5** - a Python library which allows to visualize and debug various Machine Learning models using unified API. It has built-in support for several ML frameworks and provides a way to explain black-box models.
- **SHAP** (SHapley Additive exPlanations) - a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations, uniting several previous methods and representing a consistent and locally accurate additive feature attribution method based on expectations.
- **Lime** (local interpretable model-agnostic explanations) - explains what machine learning classifiers (or models) are doing. Supports explaining individual predictions for text classifiers or classifiers that act on tables (numpy arrays of numerical or categorical data) or images.

Results

Explanations with ELI5

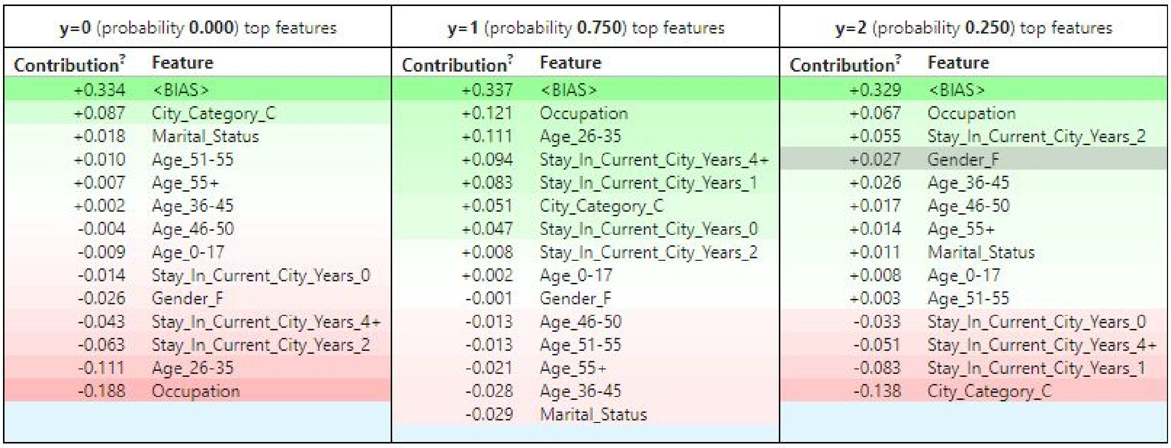
ELI5 explains weights and predictions. This gives us breadcrumbs to understand, why the algorithm made a certain decision.
Using ELI5 over logistic regression running on Black Friday data we've received the results below.
This is how it looks like for Logistic Regression:



y=0 (probability 0.348, score 0.060) top features		y=1 (probability 0.398, score 0.193) top features		y=2 (probability 0.254, score -0.253) top features	
Contribution?	Feature	Contribution?	Feature	Contribution?	Feature
+0.366	City_Category_C	+0.209	City_Category_C	+0.215	Age_26-35
+0.021	Stay_In_Current_City_Years_1	+0.101	Occupation	+0.070	<BIAS>
-0.039	<BIAS>	-0.018	Stay_In_Current_City_Years_1	+0.040	Occupation
-0.141	Occupation	-0.031	<BIAS>	-0.003	Stay_In_Current_City_Years_1
-0.147	Age_26-35	-0.068	Age_26-35	-0.576	City_Category_C

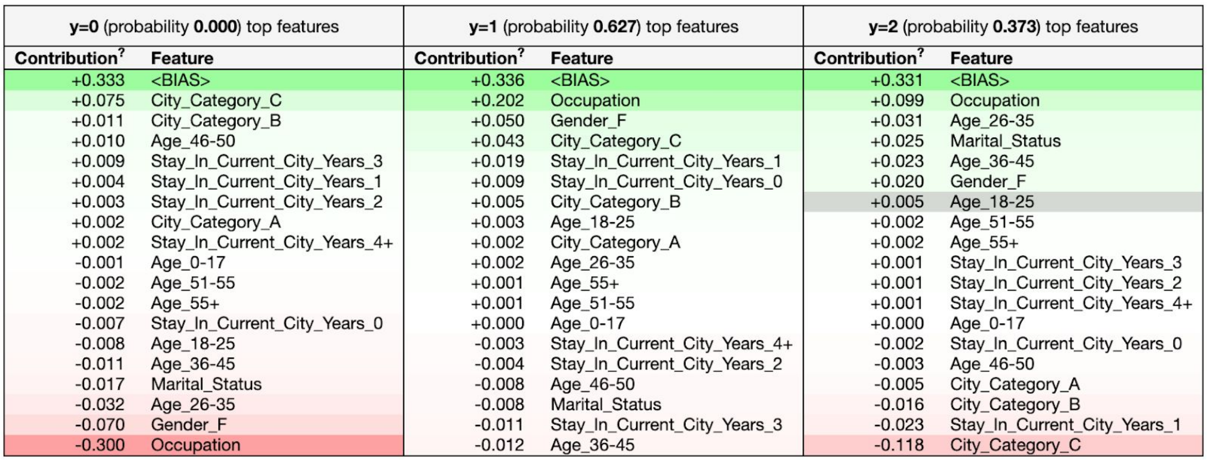
It explains the contribution of each feature to the final probability. For the Black Friday dataset, we've also seen a significant contribution on City_Category_C feature on the other datasets.

This is how it looks like for Decision Tree:



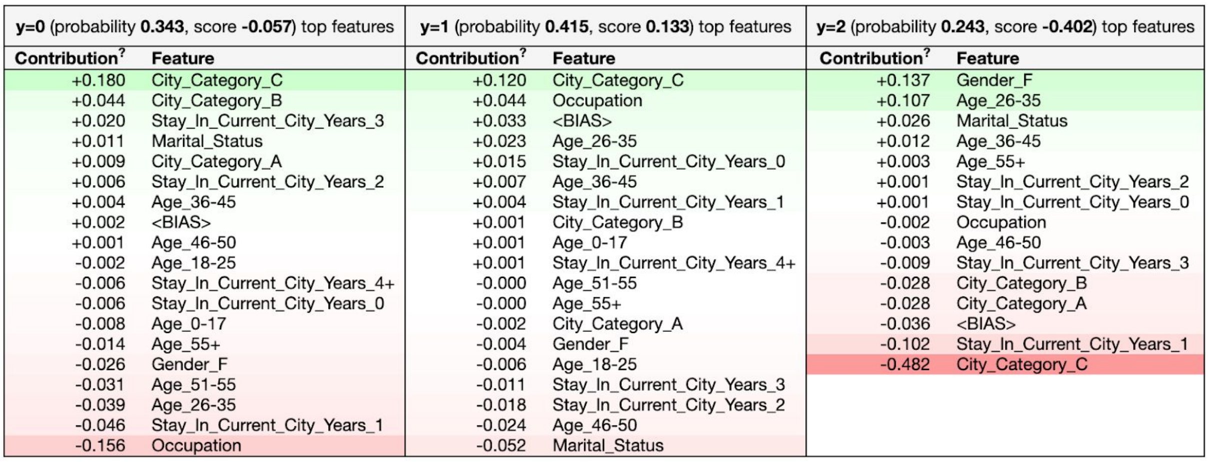
y=0 (probability 0.000) top features		y=1 (probability 0.750) top features		y=2 (probability 0.250) top features	
Contribution?	Feature	Contribution?	Feature	Contribution?	Feature
+0.334	<BIAS>	+0.337	<BIAS>	+0.329	<BIAS>
+0.087	City_Category_C	+0.121	Occupation	+0.067	Occupation
+0.018	Marital_Status	+0.111	Age_26-35	+0.055	Stay_In_Current_City_Years_2
+0.010	Age_51-55	+0.094	Stay_In_Current_City_Years_4+	+0.027	Gender_F
+0.007	Age_55+	+0.083	Stay_In_Current_City_Years_1	+0.026	Age_36-45
+0.002	Age_36-45	+0.051	City_Category_C	+0.017	Age_46-50
-0.004	Age_46-50	+0.047	Stay_In_Current_City_Years_0	+0.014	Age_55+
-0.009	Age_0-17	+0.008	Stay_In_Current_City_Years_2	+0.011	Marital_Status
-0.014	Stay_In_Current_City_Years_0	+0.002	Age_0-17	+0.008	Age_0-17
-0.026	Gender_F	-0.001	Gender_F	+0.003	Age_51-55
-0.043	Stay_In_Current_City_Years_4+	-0.013	Age_46-50	-0.033	Stay_In_Current_City_Years_0
-0.063	Stay_In_Current_City_Years_2	-0.013	Age_51-55	-0.051	Stay_In_Current_City_Years_4+
-0.111	Age_26-35	-0.021	Age_55+	-0.083	Stay_In_Current_City_Years_1
-0.188	Occupation	-0.028	Age_36-45	-0.138	City_Category_C
		-0.029	Marital_Status		

This is how it looks like for Random Forest Classifier:



y=0 (probability 0.000) top features		y=1 (probability 0.627) top features		y=2 (probability 0.373) top features	
Contribution?	Feature	Contribution?	Feature	Contribution?	Feature
+0.333	<BIAS>	+0.336	<BIAS>	+0.331	<BIAS>
+0.075	City_Category_C	+0.202	Occupation	+0.099	Occupation
+0.011	City_Category_B	+0.050	Gender_F	+0.031	Age_26-35
+0.010	Age_46-50	+0.043	City_Category_C	+0.025	Marital_Status
+0.009	Stay_In_Current_City_Years_3	+0.019	Stay_In_Current_City_Years_1	+0.023	Age_36-45
+0.004	Stay_In_Current_City_Years_1	+0.009	Stay_In_Current_City_Years_0	+0.020	Gender_F
+0.003	Stay_In_Current_City_Years_2	+0.005	City_Category_B	+0.005	Age_18-25
+0.002	City_Category_A	+0.003	Age_18-25	+0.002	Age_51-55
+0.002	Stay_In_Current_City_Years_4+	+0.002	City_Category_A	+0.002	Age_55+
-0.001	Age_0-17	+0.002	Age_26-35	+0.001	Stay_In_Current_City_Years_3
-0.002	Age_51-55	+0.001	Age_55+	+0.001	Stay_In_Current_City_Years_2
-0.002	Age_55+	+0.001	Age_51-55	+0.001	Stay_In_Current_City_Years_4+
-0.007	Stay_In_Current_City_Years_0	+0.000	Age_0-17	+0.000	Age_0-17
-0.008	Age_18-25	-0.003	Stay_In_Current_City_Years_4+	-0.002	Stay_In_Current_City_Years_0
-0.011	Age_36-45	-0.004	Stay_In_Current_City_Years_2	-0.003	Age_46-50
-0.017	Marital_Status	-0.008	Age_46-50	-0.005	City_Category_A
-0.032	Age_26-35	-0.008	Marital_Status	-0.016	City_Category_B
-0.070	Gender_F	-0.011	Stay_In_Current_City_Years_3	-0.023	Stay_In_Current_City_Years_1
-0.300	Occupation	-0.012	Age_36-45	-0.118	City_Category_C

And this is for XGboost:



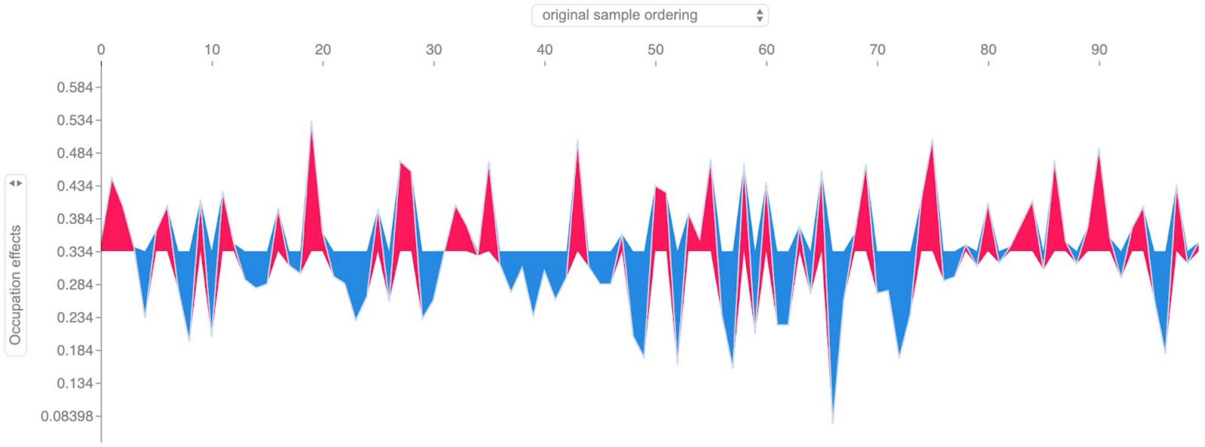
y=0 (probability **0.343**, score **-0.057**) top features		**y=1** (probability **0.415**, score **0.133**) top features		**y=2** (probability **0.243**, score **-0.402**) top features	
Contribution?	**Feature**	**Contribution?**	**Feature**	**Contribution?**	**Feature**
+0.180	City_Category_C	+0.120	City_Category_C	+0.137	Gender_F
+0.044	City_Category_B	+0.044	Occupation	+0.107	Age_26-35
+0.020	Stay_In_Current_City_Years_3	+0.033	<BIAS>	+0.026	Marital_Status
+0.011	Marital_Status	+0.023	Age_26-35	+0.012	Age_36-45
+0.009	City_Category_A	+0.015	Stay_In_Current_City_Years_0	+0.003	Age_55+
+0.006	Stay_In_Current_City_Years_2	+0.007	Age_36-45	+0.001	Stay_In_Current_City_Years_2
+0.004	Age_36-45	+0.004	Stay_In_Current_City_Years_1	+0.001	Stay_In_Current_City_Years_0
+0.002	<BIAS>	+0.001	City_Category_B	-0.002	Occupation
+0.001	Age_46-50	+0.001	Age_0-17	-0.003	Age_46-50
-0.002	Age_18-25	+0.001	Stay_In_Current_City_Years_4+	-0.009	Stay_In_Current_City_Years_3
-0.006	Stay_In_Current_City_Years_4+	-0.000	Age_51-55	-0.028	City_Category_B
-0.006	Stay_In_Current_City_Years_0	-0.000	Age_55+	-0.028	City_Category_A
-0.008	Age_0-17	-0.002	City_Category_A	-0.036	<BIAS>
-0.014	Age_55+	-0.004	Gender_F	-0.102	Stay_In_Current_City_Years_1
-0.026	Gender_F	-0.006	Age_18-25	-0.482	City_Category_C
-0.031	Age_51-55	-0.011	Stay_In_Current_City_Years_3		
-0.039	Age_26-35	-0.018	Stay_In_Current_City_Years_2		
-0.046	Stay_In_Current_City_Years_1	-0.024	Age_46-50		
-0.156	Occupation	-0.052	Marital_Status		

Overall across different models, we can see a huge contribution of the City_Category_C. In our tests, results achieved with ELI5 were corresponding to the results of SHAP and Lime algorithms.

Explanations with SHAP

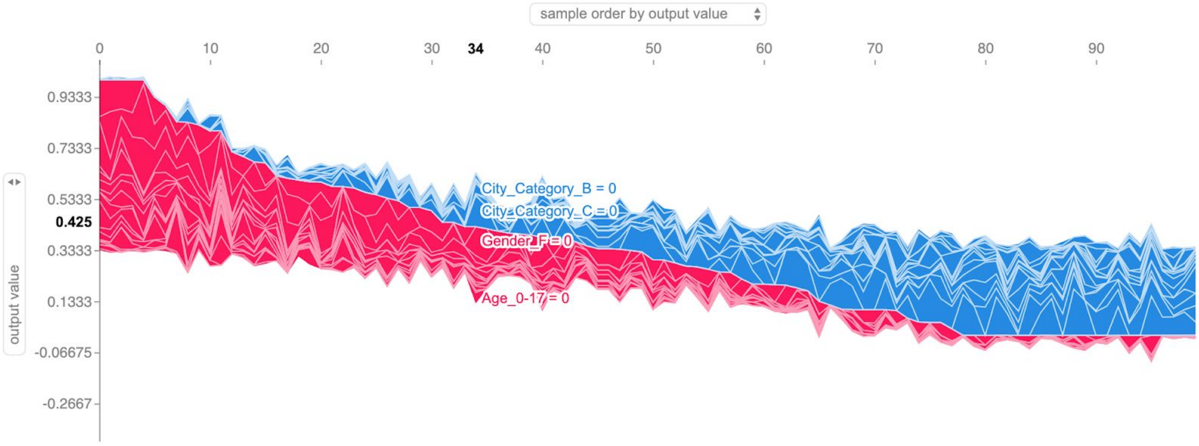
SHAP helps to answer how the model works for an individual prediction. We used it to explain three of our models: Decision Tree, Random Forest, and XGboost. SHAP itself provides powerful interactive visualizations. In order to show them, we need to calculate SHAP values. They interpret the impact of having a certain value for a given feature in comparison to the prediction we'd make if that feature took some baseline value. Usually, on SHAP visualizations feature values causing increased predictions are in pink, and their visual size shows the magnitude of the feature's effect. Feature values decreasing the prediction are in blue.

One type of such visualizations in SHAP is `force_plot`. Here is it's an example for Decision Tree:



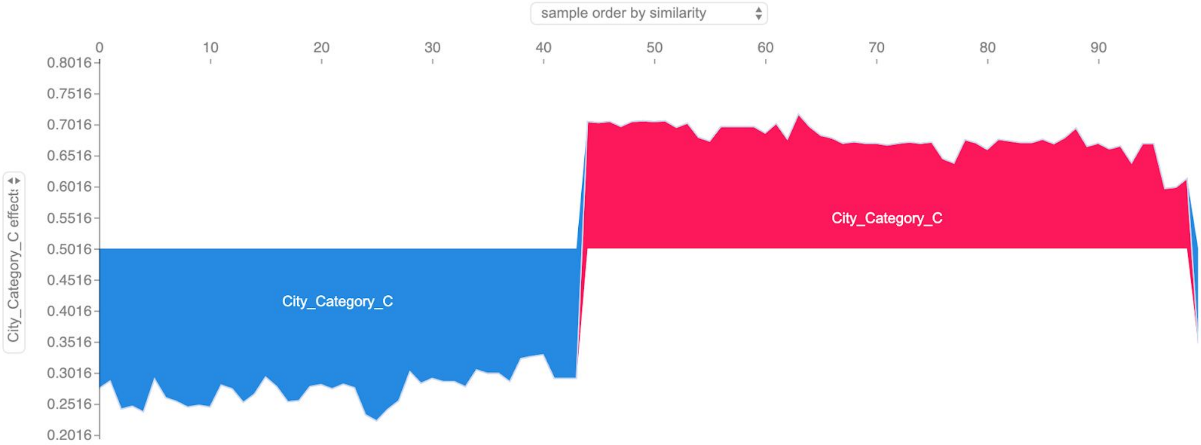
Here data is ordered as it is originally in a sample (first 100 elements). As value here we can see the impact of "Occupation" field. It's numerical property, so it's very good to see how the impact of its value changes for different data points. For both axes, we can change what to display, how to order data. More details and plots variations can be found in the source file ('/modelling/explained_fitted_models/explained_models.ipynb').

Here is a bit another example for Random Forest Classifier:



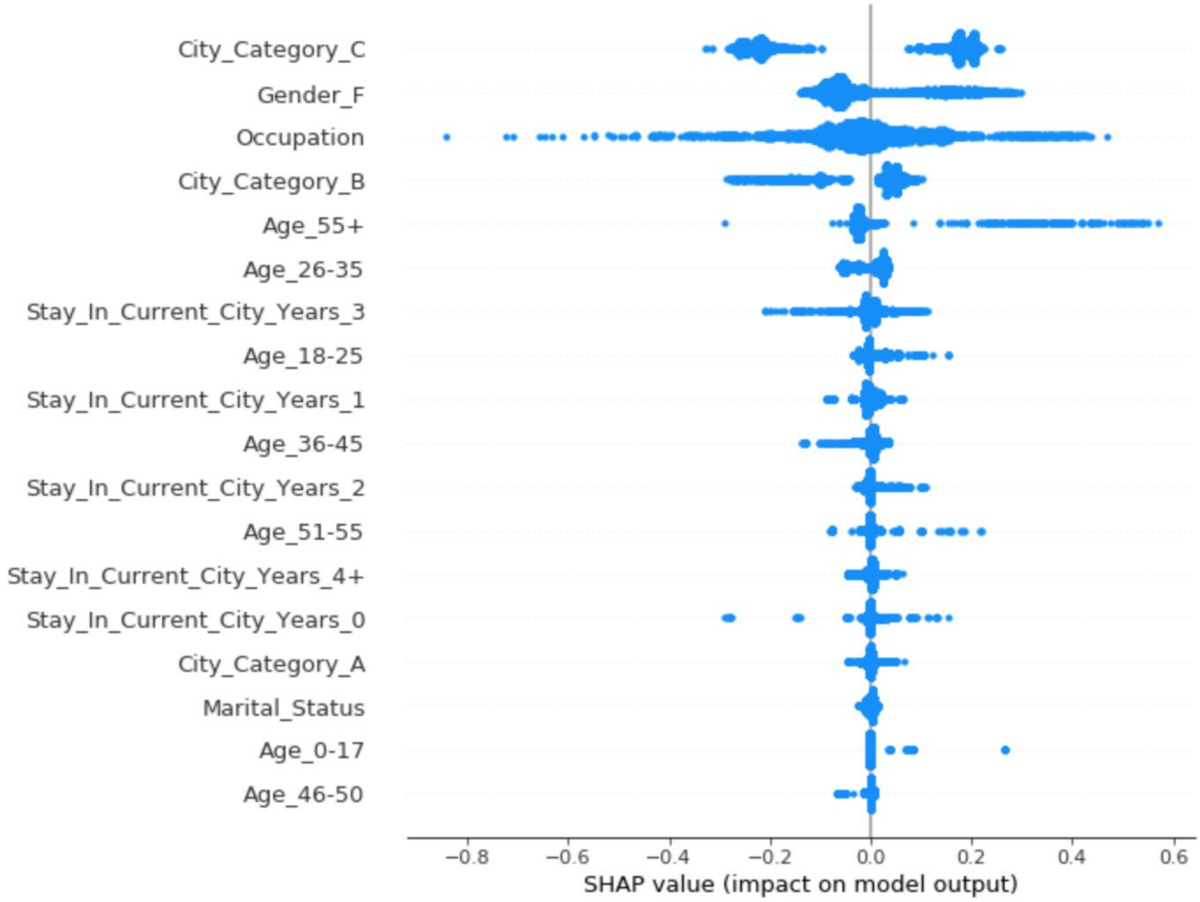
Here data is ordered by the output value. We can see the total impact of all fields onto output value.

And one more example for XGboost:



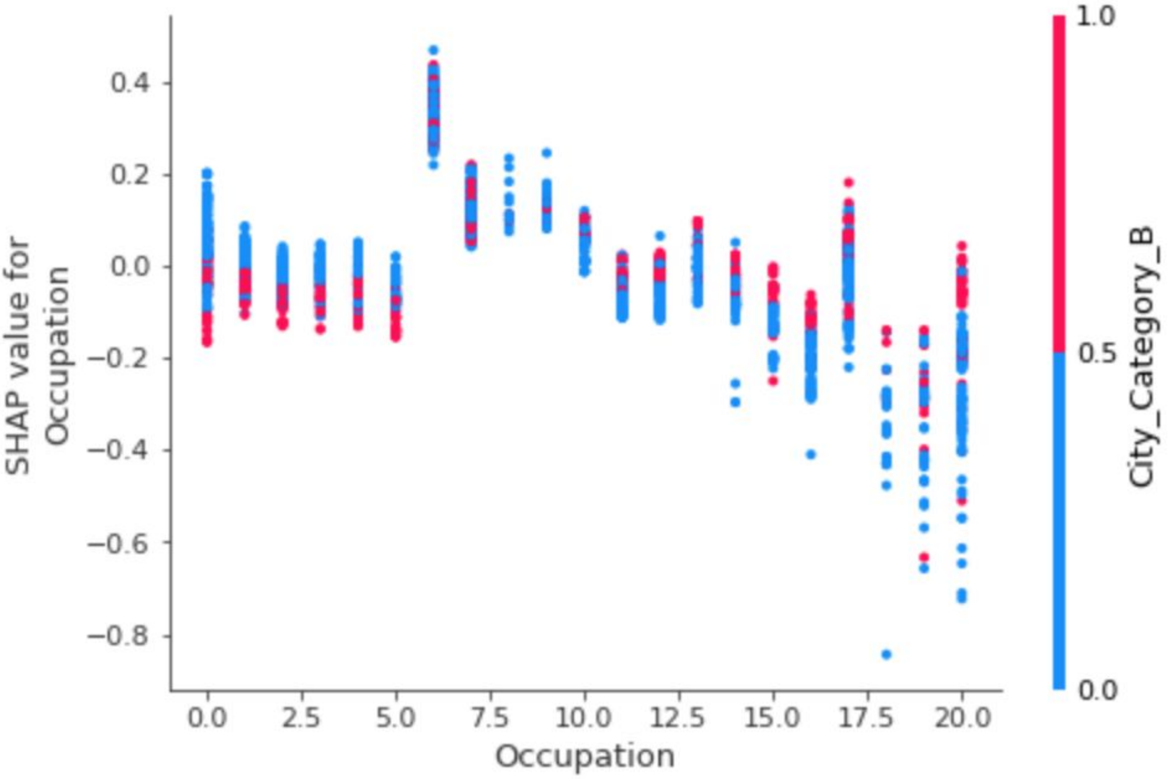
Here data is ordered by similarity. The effect of City_Category_C onto output value is displayed.

Another type of plots is `summary_plot`. The difference here is that this visualization summarises data for all features from all samples and displays in one variation. Here is an example for XGboost:



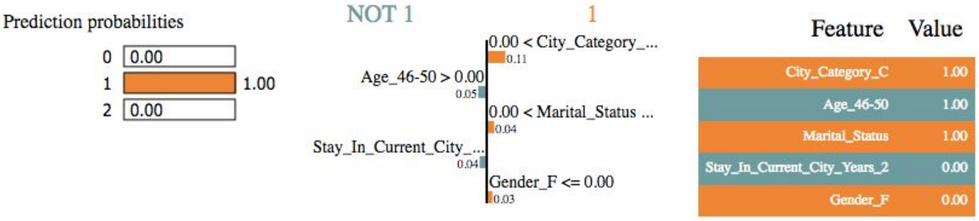
We can see the summary of all the properties of our model and how they impacted output value.

One more type of visualization is `dependence_plot`. It plots the value of the feature on the x-axis and the SHAP value of the same feature on the y-axis. This shows how the model depends on the given feature and is like a richer extension of the classical partial dependence plots. Vertical dispersion of the data points represents interaction effects. Grey ticks along the y-axis are data points where the feature’s value was NaN.
Here is an example for XGboost:



This example is for ‘Occupation’ with ‘City_Category_B’.

Explanations with LIME



Feature	Value
City_Category_C	1.00
Age_46-50	1.00
Marital_Status	1.00
Stay_In_Current_City_Years_2	0.00
Gender_F	0.00

Same as in the example with ELI5, we can see that living in a particular city affects the classifier decision. These results were obtained using the Decision Trees algorithm and, by design of the algorithm, it is easy to understand the features that would affect final prediction out of the box.

Summary

Understanding feature importance and affect over the model’s predictions results is important to explain and trust the results.

Provided framework gives an ability to see the results produced by ELI5, Lime and SHAP tools, cross-validated calculated significances and understand whether the model has any serious issues.

The code is implemented in the style which allows to easily extend the current solution. Adding new datasets, models or different frameworks does not require architectural changes to the project.

Appendices description

With this report we also send two files with detailed code and results of our work:

- Appendix 1 (‘presenter.html’) - models explanations, including interactive plots
- Appendix 2 (‘models_training.html’) - models training logic results

Resources

1. https://www.kaggle.com/mehdidag/black-friday - black Friday dataset

2. https://www.kaggle.com/zaurbegiev/my-dataset - loan data dataset

3. https://github.com/roman-riazantsev/tensoptic-flow - GitHub repository